

**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**



Fundamentos de un ERP con administración eficiente y optimización técnica de los recursos.

Tesis para obtener el título de:

Ingeniero en computación

Presenta:

Erick Reynaldo Valdivia Ortega

Luis Enrique Gutiérrez Zamora

Pedro Ruiz Espinoza

Director de tesis:

Ing. Román V. Osorio Comparan

Ciudad Universitaria, México, D.F. Diciembre, 2012





# Índice

Índice.....	I
Agradecimientos .....	IV
Objetivo.....	V
Definición del problema .....	V
Método .....	VI
Resumen .....	VII
<b>1. Fundamentos de un sistema ERP .....</b>	<b>1</b>
1.1. Definición de ERP y los sistemas ERP.....	1
1.2. La necesidad de un ERP en las organizaciones actuales.....	1
1.3. Evolución de los sistemas ERP.....	2
1.3.1. Sistemas predecesores de los ERP .....	2
1.3.2. Historia del ERP de SAP .....	3
1.4. Productos que complementan el ERP de SAP.....	4
1.5. Requisitos actuales mínimos de un sistema ERP.....	5
1.6. Componentes de un ERP .....	5
1.6.1. Componentes del ERP de SAP.....	8
1.7. SAP Netweaver como plataforma de adaptación y preparación de los procesos de negocio ...	11
1.7.1. Modelo de SAP Netweaver como arquitectura de la empresa (people view, organizational view, information view, business process view, information technology view) .....	12
<b>2. Metodología de implementación .....</b>	<b>15</b>
2.1. Metodología ASAP para la implementación de la solución .....	15
2.1.1. Preparación del proyecto.....	17
2.1.2. Plan de negocio.....	18
2.1.3. Realización.....	19
2.1.4. Preparación final.....	20
2.1.5. Salida en productivo y soporte.....	21
2.1.6. Operar el sistema .....	21
2.2. Metodología RunSAP para la implementación de las operaciones .....	22
2.2.1. Valoración y alcance.....	25
2.2.2. Diseño de las operaciones.....	27
2.2.3. Configurar las operaciones .....	31
2.2.4. Transferencia a producción.....	31
2.2.5. Operaciones y optimización.....	32
2.3. SAP Solution Manager y las normas ITIL.....	33
2.4. Listo para SOA.....	37
<b>3. Estructura técnica del sistema SAP ERP .....</b>	<b>38</b>
3.1. Tecnología cliente/servidor.....	38
3.2. Arquitectura del software de SAP.....	38
3.3. Tecnología ABAP .....	39
3.3.1. Procesos de un servidor de aplicaciones SAP ABAP .....	40
3.4. Tecnología Java.....	41
3.4.1. Procesos de un servidor de aplicaciones Java.....	41
3.5. El término instancia en sistemas SAP .....	42
3.5.1. La instancia en un ambiente ABAP+Java.....	42

3.5.2. Tipos de servidor de aplicaciones SAP .....	43
<b>4. Implementación técnica del sistema .....</b>	<b>45</b>
4.1. Planeación .....	45
4.2. Preparación del sistema .....	47
4.3. Requerimientos de instalación .....	48
4.4. Tipos de instalación .....	48
4.4.1. Instalación local .....	48
4.4.2. Instalación de dominio .....	48
4.4.3. Configuración en alta disponibilidad.....	49
4.5. Componentes de software necesarios .....	50
4.6. Instancia central y servicios centrales .....	52
4.7. Actividades posteriores a la instalación del sistema .....	58
4.8. Configuración técnica del sistema.....	60
4.8.1. Herramientas de configuración .....	60
4.9. Diseño del ambiente de la solución.....	61
4.9.1. Sistemas de transportes .....	62
4.9.2. SLD.....	65
<b>5. Administración del sistema .....</b>	<b>67</b>
5.1. Iniciar y detener el sistema SAP ERP de forma adecuada.....	67
5.1.1. Proceso de inicio de un sistema SAP ERP.....	67
5.1.2. Inicio del sistema: Bitácoras.....	68
5.1.3. Proceso de apagado de un sistema SAP .....	69
5.2. Configuración del sistema.....	69
5.2.1. Configuración de los parámetros del sistema .....	69
5.2.2. Ajustar los parámetros del sistema .....	71
5.3. Mantenimiento de versiones .....	71
5.3.1. Importar “support packages” .....	71
5.3.2. Actualizar las herramientas.....	73
5.3.3. Importar notas de SAP .....	74
5.4. Administración de mandantes del sistema.....	77
5.4.1. Estructura de datos de un sistema SAP .....	77
5.4.2. Mandante .....	78
5.4.3. Opciones de cambios en mandante.....	79
5.5. Programación de trabajos en fondo .....	81
5.5.1. Calendarización y Monitoreo de Procesos de Fondo.....	82
<b>6. Administración del servidor de aplicaciones Java.....</b>	<b>83</b>
6.1. Herramientas de administración y configuración del servidor de aplicaciones Java .....	83
6.1.1. Config Tool .....	83
6.1.2. Visual Administrator .....	84
6.1.3. Configuración del servidor de aplicaciones SAP Java con la herramienta Config Tool .....	84
6.1.4. Configuración del servidor de aplicaciones SAP Java con la herramienta Visual Administrator.....	86
6.2. Administración de versiones .....	91
6.2.1. Instalar correcciones para el servidor de aplicaciones Java .....	91
6.2.2. Instalar correcciones para una aplicación Java .....	96
<b>7. Evaluación técnica para un buen rendimiento y escalabilidad para los sistemas SAP en base a la plataforma ABAP .....</b>	<b>99</b>

7.1. Plan para la optimización continua del rendimiento en el sistema.....	99
7.2. Análisis del rendimiento .....	101
7.2.1. Problemas de desempeño generalizados .....	104
7.3. Análisis de Carga en el sistema .....	123
7.3.1. Distribución de la carga de trabajo en los sistemas SAP ABAP .....	124
7.3.2. ST03N.....	125
7.4. Problemas de desempeños específicos .....	128
7.4.1. Ejecución de trazas y su utilización (ST12, SAT, SE30, ST05) .....	130
7.4.2. Optimización en la capa de Base de Datos (Oracle) .....	145
7.4.3. Optimización en la capa de ABAP con programación eficiente .....	159
7.5. Dimensionamiento del Hardware .....	176
<b>8. Tendencias .....</b>	<b>184</b>
8.1. Base de datos en memoria .....	184
8.2. Clientes móviles .....	185
8.3. SAP en sistemas virtualizados (cloud computing) .....	186
<b>Conclusiones.....</b>	<b>190</b>
<b>Bibliografía .....</b>	<b>191</b>
<b>Glosario .....</b>	<b>196</b>

## Agradecimientos

Quiero agradecer a mis padres, Germán R. Valdivia García y Graciela Ortega Chávez, porque siempre me procuraron una educación y en ningún momento han dejado de apoyarme, todo lo que he logrado se los debo a ellos. También agradezco a mi hermano, Iván R. Valdivia Ortega, quien ha compartido su conocimiento y experiencia conmigo, conforme hemos crecido. Sin duda, los demás miembros de la familia (abuelos, tíos, primos, etc.) también han sido parte de mi educación y les agradezco su ayuda.

Una pieza fundamental para la finalización de este logro, son todos los profesores que dedicaron su tiempo y esfuerzo en enseñarme y corregirme. En especial, quiero agradecer a los profesores de la UNAM. Y aprecio en gran medida, el apoyo, tanto intelectual como moral, proporcionado por nuestro asesor, el Ing. Román Osorio Comparan.

Por último, quiero agradecer también, a mis compañeros y amigos, Pedro Ruiz Espinoza y Luis Enrique Gutiérrez Zamora, por la disposición y experiencia aportados para completar este trabajo.

Erick Reynaldo Valdivia Ortega

Dedico esta tesis a mis padres Pedro Ruiz Rodríguez y Estela Espinoza Vargas por el apoyo incondicional que me han brindado durante todos estos años, este trabajo representa la culminación de mucho tiempo de estudio y dedicación continua para poder alcanzar este logro.

Agradezco a mis amigos y compañeros de este proyecto Erick Valdivia y Luis Gutiérrez, por todo el esfuerzo que demostramos durante este tiempo para culminar este trabajo, de la misma manera agradezco también al Ing. Román Osorio Comparan, por la dedicación y el tiempo que nos dedicó para dar inicio y conclusión a nuestra tesis.

Finalmente me siento muy orgulloso de expresar un agradecimiento a mi Universidad, la UNAM, mi alma mater, ya que esta institución me ha brindado las bases para formarme, crecer y desarrollarme profesionalmente y así poder alcanzar cada una de las metas y retos que se presentan día con día durante el ejercicio de mi carrera profesional.

Pedro Ruiz Espinoza

Existen momentos que se tienen que recordar y compartir, la culminación de una carrera profesional con su trabajo de tesis es uno de ellos. A mis padres Carlota y Salvador que son los fundamentos de todo lo que soy y seré; a mis hermanas que han compartido conmigo siempre sus ideas y reflexiones; a Viridiana por estar a mi lado; a todas las personas que me han dado algo de sí para formar una amistad, una charla, una discusión o una idea para continuar disfrutando de la vida y cumplir mis sueños.

Agradezco a mi casa de estudios la UNAM, a su profesores y alumnos por dejar ampliar mi horizonte de conocimientos e introducirme al esquivito mudo del saber.

Finalmente agradezco a mi compañeros de tesis Erick y Pedro que en conjunto con nuestro asesor Román logramos concluir este trabajo.

Luis Enrique Gutiérrez Zamora

## Objetivo

Transferir el conocimiento profesional adquirido durante los proyectos de implementación técnica para aplicaciones de SAP (Systeme, Anwendungen und Produkte) con el fin de tener un software estable y escalable para la plataforma tecnológica de ABAP (Advanced Business Application Programming).

Los estudiantes y egresados de ingeniería deben de tener un acercamiento real a las aplicaciones industriales que se utilizan para poder llevar a cabo los procesos de negocio de las grandes empresas. Es importante que los profesionistas mexicanos se mantengan competitivos globalmente, sin embargo en ocasiones esta meta se vuelve un desafío por la falta de integración con las empresas e instituciones que llevan la vanguardia en la tecnología.

La UNAM como universidad líder en Latinoamérica, debe aportar profesionistas con una visión globalizada de las tecnologías de la información y cómo influyen en las industrias.

SAP es uno de los líderes en software de negocio con una gran participación a nivel mundial en una gran variedad de industrias. SAP utiliza una serie de estándares globales abiertos, los cuales hacen que sus sistemas sean fáciles de integrar.

Existe un campo laboral amplio para trabajar con sistemas ERP de SAP en México. Es importante que los alumnos de ingeniería en computación se puedan integrar en estos proyectos de implementación, administración, mantenimiento y desarrollo de aplicaciones con conocimiento previo de las soluciones de SAP.

El contenido teórico de la tesis cubre algunos de los principales temas que un consultor técnico asociado certificado (C\_TADM – Academia Basis de SAP) debe conocer, a su vez se exponen algunas de las experiencias profesionales que pueden ayudar a la resolución de problemas en el ámbito profesional. Cabe resaltar que este trabajo no sustituye la documentación oficial ni los entrenamientos de SAP.

## Definición del problema

Los costos y tiempo de una implementación de un sistema ERP (Enterprise Resource Planning) de SAP usualmente se ven rebasados por la falta de experiencia previa de los ingenieros involucrados. En la parte tecnológica, el desconocimiento de la arquitectura y su integración, crea problemas en las fases tempranas del proyecto, los cuales se ven reflejados en etapas posteriores con problemas de estabilidad, escalabilidad y poco espacio para la innovación tecnológica.

Al ser SAP una empresa transnacional, en ocasiones es difícil obtener la capacitación necesaria o la transferencia de conocimiento por los altos costos del material y el entrenamiento. El contenido de esta investigación trata de brindar las bases para poder entender mejor esta tecnología. Esto permitirá a los ingenieros salir a la vida profesional con el lenguaje técnico adecuado y el conocimiento propio que requieren para trabajar con este proveedor de software como consultores o integradores.

Esta investigación representa una breve guía para que el lector tome de referencia los modelos y paradigmas que la empresa SAP publica como mejores prácticas. Los patrones de referencia que se exponen, pueden aplicar no sólo a las soluciones que involucran software con la tecnología de SAP, sino que se pueden extrapolar a cualquier otro tipo componente de software, e incluso tomarlos como referencia en el caso de que se construyan aplicaciones para complementar huecos propios de la localización geográfica o requisitos de los procesos de negocio.

Entender es el primer paso para poder empezar a crear.



## **Método**

La metodología que seguiremos será de carácter deductivo, con una técnica de análisis de contenido y documentación, de casos profesionales en los que nos hemos involucrado.

En la parte deductiva, partimos siempre de los conceptos generales de la arquitectura del software, para poder entender en fases posteriores, la justificación de las recomendaciones técnicas que realizamos. Además tratamos de homologar la información contenida en diversos documentos, para poder definir una directriz clara que unifica la experiencia y la documentación.

La población que manejamos en los ejemplos es de carácter confidencial, sin embargo representa los proyectos en los cuales hemos laborado, por esta razón las muestras de código, diagramas y esquemas son ejemplos representativos reales.

## Resumen

- El capítulo 1, “Fundamentos de un sistema ERP”, explica a modo introductorio la representación de un software de ERP en la actualidad y su evolución como una necesidad de industrialización moderna de procesos. Se explica en breve los componentes básicos de un ERP y cómo cada uno de ellos ayuda a las organizaciones complejas que tenemos en la actualidad. Finalmente, presentamos la plataforma tecnológica de SAP Netweaver como la base para productos de integración, comunicación, inteligencia de negocios y desarrollo.
- El capítulo 2, “Metodología de implementación”, está enfocado en presentar las metodologías ASAP y RunSAP para la implementación y operaciones de soluciones SAP. De una forma breve y clara, se expone las fases que compone cada una de ellas, así como quienes son los responsables y entregables asociados a cada una de las etapas. Durante el desarrollo del capítulo, se presentará la forma en la que se debe llevar a cabo un proyecto de implementación de ERP de la manera óptima.
- El capítulo 3, “Estructura técnica del sistema SAP ERP”, nos introduce a la arquitectura de los servidores de aplicación ABAP y Java. Se presentan los procesos que los conforman, así como la función de cada uno de éstos. Hacia el final de este capítulo se muestran las diferentes configuraciones en las que se pueden instalar los servidores de aplicación.
- El capítulo 4, “Implementación técnica del sistema”, describe las actividades necesarias para preparar la infraestructura técnica que soportará la operación de la solución SAP a implementar. Estas actividades, consideran la identificación y diseño de cada uno de los componentes de software que será necesario instalar, así como las actividades a ejecutar para habilitar dicha plataforma tecnológica.
- El capítulo 5, “Administración del sistema”, cubre las tareas de administración más importantes para operar el servidor de aplicación ABAP. Estas tareas incluyen la forma correcta de iniciar y detener el sistema, la configuración de parámetros, la implementación de actualizaciones a los componentes que forman el servidor de aplicaciones, entre otros. El capítulo introduce la administración de mandantes y finalmente se exponen los procesos de fondo y la forma de programarlos.
- El capítulo 6, “Administración del servidor de aplicaciones Java”, provee toda la información relacionada con las tareas y herramientas que nos ayudarán a una correcta administración de la parte Java de un servidor de aplicaciones. Una parte importante dentro del capítulo es explicar la forma en la que se aplican actualizaciones al servidor de aplicación Java.
- El Capítulo 7, “Evaluación técnica para un buen rendimiento y escalabilidad para los sistemas SAP en la plataforma ABAP”, expone de manera más especializada las prácticas que se recomiendan ejecutar técnicamente para poder asegurar un rendimiento optimizado de la solución. Se explican las herramientas adecuadas de monitoreo y algunas reglas generalizadas a seguir. Finalmente, se ejemplifican situaciones y experiencias basadas en el trabajo realizado en campo con sistemas reales. Se recomienda leer este capítulo ya que se tengan las bases de arquitectura y administración básica.
- El capítulo 8, “Tendencias”, nos presenta las nuevas tecnologías que se están incorporando en el mercado. Los objetivos de estas tendencias son proveer a los usuarios de una mayor movilidad; a los sistemas de capacidades de procesamiento más ágil y en general a las empresas con herramientas que permitirán la reducción de costos en el mantenimiento de sus aplicaciones. Entre las tecnologías que se comienzan a implementar y que se analizan en este capítulo, están las bases de datos en memoria, los clientes móviles y la computación en la nube.



# 1. Fundamentos de un sistema ERP

## 1.1. Definición de ERP y los sistemas ERP

Planeación de Recursos de la Empresa ERP (Enterprise Resource Planning por sus siglas en inglés), es una práctica para manejar un negocio. Usa las tecnologías de la información como herramienta para la integración de los procesos claves de negocio. El ERP debe ser un facilitador para alcanzar los objetivos de negocio o de una organización.

En un software ERP como lo llamaremos en este trabajo, sólo existe una base de datos a través de la cual todos los departamentos de una organización tales como producción, ventas, finanzas, logística, recursos humanos y servicio al cliente, manejan sus procesos de negocios.

Un ERP ayuda a evitar que las compañías tengan un sistema por cada departamento y ayuda a que la comunicación inter-compañía se simplifique, lo cual evita problemas de integración entre los sistemas de cada uno de los departamentos.

Cuando las áreas dentro de una compañía no están integradas, la información tiene que ser capturada cada vez que cambia o se transporta entre las áreas. Por ejemplo, si se manda una orden de compra de materiales para el almacén, el departamento de almacén tiene que enviar la propuesta al equipo de compras, que a su vez tiene que enviarlo a finanzas para su aprobación dentro del presupuesto. En este caso cada una de las áreas tiene que agregar la información necesaria de su departamento al sistema. Esto no sólo incrementa la oportunidad para introducir errores, sino que también aumenta el tiempo para obtener el material.

Otro punto importante, es que si se requiere cambiar el proceso de negocio para incorporar alguna mejora o requerimiento legal, implementar dichos cambios en los sistemas de cada departamento, representaría un mayor costo en tiempo y recursos.

En un sistema ERP se capturan los detalles una sola vez, una compra por ejemplo, y esta información debe estar disponible para todos los departamentos. Esto asegura que la comunicación inter-departamental, ocurra casi de manera instantánea y cada uno de los departamentos colabore con la misma información.

Lo mismo aplica en caso de que se quiera modificar un proceso de negocio, sólo un sistema tiene que ser modificado, y esto afecta a todos los departamentos involucrados. Esto además de simplificar la adaptación de procesos de negocio, ayuda a la consistencia de información para auditorías.

El problema principal para el software empresarial es que el departamento de tecnologías de la información de la empresa suele tener problemas en términos de integración y rediseño de procesos de negocio. Es por eso que la arquitectura del software debe ser orientada a procesos con el rol de un habilitador al negocio principal. El software debe adaptarse y crearse en base a este paradigma.

## 1.2. La necesidad de un ERP en las organizaciones actuales

En el ambiente actual, altamente globalizado y competitivo, es esencial ser flexible y ágil al mismo tiempo. Un diseño apropiado y la alineación de los sistemas empresariales con las estrategias de negocios, pueden permitir a los negocios crear una ventaja competitiva en los mercados globales.

La respuesta eficiente al cambio, aunque sean eventos impredecibles, debe ser una característica de las empresas modernas, y es por eso que se deben tener sistemas empresariales que faciliten estas características.

Uno de los intereses primordiales de las compañías actuales es hacer dinero. Las empresas producen productos o servicios, los cuales se venden a clientes para tener ganancias financieras. Al generar esos productos o servicios, se necesitan las materias primas y materiales operativos, las cuales se compran a los proveedores.

Para obtener un mejor margen de ganancia en esta interacción, se necesita el manejo adecuado de clientes y proveedores, además de todos los aspectos dentro del negocio como son empleados, cuentas de bancos, inventarios, entre otros.

Para hacer todo esto manejable, la compañía cuenta con varios departamentos, cada uno encargado de un aspecto particular. Entre los principales departamentos se encuentran finanzas, manufactura, recursos humanos, logística, etc.

Cada uno de estos departamentos, debe de contar con software que le permita manejar su actividad diaria y a su vez, comunicarse con los demás departamentos cuando sea necesario.

Al implementar un software ERP, se instala una solución en toda la empresa, la cual consiste en varios módulos de software y dependiendo de las características y/o requerimientos de cada empresa, el software ERP se adapta a sus procesos y necesidades. Cada departamento de la compañía puede usar uno o más módulos del software ERP para manejar sus actividades diarias.

Otro tema indispensable para la rentabilidad de la empresa es la creación de reportes cada determinado tiempo. Los más comunes son cada fin de mes/cuarto/año (fin de un periodo), mostrando el avance del negocio. Este tipo de reportes, ayuda a la toma de decisiones y al ajuste de la estrategia de la empresa, es decir, a optimizar los recursos con los que cuenta.

Para lograr lo anterior todos los módulos del ERP deben estar conectados, de esta manera tenemos una vista de la compañía como un todo y podemos observar las diferentes aristas que conforman la compañía.

### **1.3. Evolución de los sistemas ERP**

Con el cambio en los requerimientos de negocios, las soluciones de software de negocios han ido evolucionando a través de los años, en conjunto con las necesidades de los negocios y empresas. A continuación, comentaremos brevemente los principales sistemas predecesores de los ERP.

#### **1.3.1. Sistemas predecesores de los ERP**

MRP I. Planeación de necesidades de materiales.

En los años sesenta se inició la conceptualizó a los primeros sistemas MRP (Material Resource Planning) con el objetivo de sustituir los sistemas de información tradicionales de planeación y control de la producción [CZ1990]. La mayoría de los primeros MRP estaban basados en sistemas de software que fueron específicamente diseñados para cumplir con las expectativas de las industrias manufactureras, que luchaban por mantener los costos de producción bajos, debido al manejo ineficiente de los inventarios.

Su objetivo principal era la optimización del inventario, debido a que un inventario insuficiente afecta al calendario de producción, lo cual significa que las órdenes de los clientes no se cumplen y que el inventario excesivo expone a las organizaciones a pérdidas financieras por los costos de almacenaje y el envejecimiento de los productos, además de retener flujo efectivo en existencias sin usar.

Los MRP integraron todas las funciones de manejo de inventario para la reducción de costos de producción, pero uno de los principales problemas que enfrentaron las industrias con los sistemas MRP, fue la consistencia de información y la simplificación de cálculos en la planeación que tenía como consecuencia tiempos holgados de entrega [MCZ2005].

MRP II

A mediados de los años setenta, los negocios se vieron en la necesidad de coordinar por completo su proceso de manufactura para optimizar la productividad. Esto dio a la tarea de crear MRP II o MRP extendido, integrando todas las operaciones relacionadas a la manufactura dentro de la organización como la manufactura, ingeniería, finanzas, y mercadotecnia.

Ya integrado el proceso de compras con el de fabricación, el gran paso en este sistema fue integrar el área de finanzas. Este concepto de sistema de información que integró la producción, el inventario y las finanzas, fue bautizado por Ollie Wight como MRP II [MCZ2005]

Adicionalmente, el sistema de reportes de MRP II, ayudó a los gerentes a contestar preguntas especulativas para la toma de decisiones.

A principios de los años 70 cuatro ingenieros de IBM fundan SAP que se convierte en el proveedor más grande de software de negocios de hoy en día. A finales de la década se funda Lawson Software en 1975, Oracle en 1977, JD Edwards ahora Oracle en 1977 y Baan ahora Infor Global Solutions en 1978 [DK2012].

En 1979 Oracle ofrece el primer manejador de base de datos relacional SQL [EHT2012].

#### ERP

Los sistemas ERP también han evolucionado a través de los años. Los primeros ERP fueron diseñados para las empresas manufactureras, pero en la actualidad se ofrecen para cualquier tipo de organización e industria.

Para finales de los ochentas, con un crecimiento impredecible y más competencia que antes, las organizaciones se dieron cuenta que tenían que ser más eficientes en su función en general. Esto les ayudaría a adaptarse a los constantes cambios de negocio, así como a dar mejor servicio al cliente. La empresa PeopleSoft desarrolla un software de manejo de capital humano que se extiende a lo largo del mundo.

La empresa de investigación Gartner acuña el término ERP en los años noventa que envuelve los conceptos de MRP, MPR II y más aplicaciones del ciclo de vida del producto, logística, manejo del cliente, procesamiento de órdenes, finanzas y recursos humanos. Hoy en día el ERP es el mayor descriptor para aplicaciones de software empresarial [DK2012].

Los sistemas ERP han tenido mucha aceptación y éxito por su característica principal de integrar todos los procesos de negocios de una organización en un sistema. Esto asegura un flujo constante de información por toda la organización y hace eficientes sus funciones.

#### 1.3.2. Historia del ERP de SAP

Debido a que nuestro trabajo se enfoca al ERP de SAP, solo explicaremos a más detalle la historia de este producto en particular y no expondremos el resto de la historia de las otras compañías como JDEdwards, Baan u Oracle.

SAP es una compañía de origen alemán de desarrollo de software y corporativo de consultoría. El producto más conocido es el sistema SAP-ERP del cual describiremos la historia según el proyecto inspiración celebrando 40 años de SAP [PI2012] y el artículo [LRW2012].

SAP (Systeme, Anwendungen und Produkte, por sus siglas en alemán), es una empresa que se fundó en 1972, lo que la hace más vieja que Microsoft u Oracle, su principal competidor; con la visión de desarrollar una aplicación estándar para el procesamiento en tiempo real de negocios. SAP es actualmente la cuarta compañía en ventas de software a nivel mundial [VB2009].

En 1973 R/1 fue lanzado, seis años después se lanzó R/2. En 1981, empezó el rediseño por completo su plataforma tecnológica para el mercado y la industrialización.

R/2 es el primer ERP que integra todos los procesos de negocio a lo ancho de la empresa. Esta solución fue compatible con computadoras mainframe, como las de IBM y Siemens, y la interface de usuario estaba basada en caracteres.

Con el cambio de R/2 a R/3 en 1992, SAP siguió la tendencia de cambio de arquitectura de mainframe a cliente-servidor. Las compañías empezaron a utilizar diferentes tipos de tecnología, por lo cual SAP tenía que hacerse compatible con esta diversificación de hardware.

R/3 se empieza a hacer popular porque era compatible con una amplia gama de sistemas operativos, protocolos de comunicación y bases de datos. También la interfaz gráfica SAP GUI ayuda a expandir la base de sus clientes.

El nombre R/3 proviene de la "R" de Real time y 3-capas, al ser una arquitectura tipo: Base de Datos-Servidor de Aplicación-Cliente (SAPGUI). SAP introduce en el R/3, el componente Basis que es la columna vertebral tecnológica de las aplicaciones de negocio.

La última versión de SAP R/3 es la 4.7 Enterprise, la cual cambia a la nueva plataforma tecnológica llamada SAP Web Application Server (SAP Web AS), pensando ya en la adaptación de SOA y tiene un enfoque orientado a los servicios en línea. Esto hace posible que los usuarios accedan a servicios de la empresa por medio de un navegador web.

El ERP de SAP evoluciona de nombre a SAP ECC (ERP Central Component), el cual tiene los componentes principales de un ERP y puede ser manejado por varias industrias. SAP ECC puede manejar procesos de negocio mediante servicios web. La base tecnológica de SAP ECC es Netweaver y el Web AS pasa a ser uno de los componentes de Netweaver. Hay componentes extras de software que se le puede agregar al ECC y que son específicos de industria, estos van desde el comercio, petróleo y gas, manufactura discreta, aviación, banca, defensa y seguridad, sector público, sector educación e investigación, aeroespacial, etc.

SAP ECC contiene diferentes módulos que encapsulan los procesos de negocios de una organización. Cada módulo contiene un número de transacciones que permiten realizar una tarea dentro de un proceso.

#### 1.4. Productos que complementan el ERP de SAP

SAP actualmente soporta más de 40 idiomas, más de 50 tipos de moneda, más de 30 soluciones específicas de industria y más de 20 combinaciones de plataformas de hardware, base de datos y sistema operativo.

El rango de productos de SAP está clasificado en tres familias principales:

- SAP Business Suite
- Soluciones de pequeñas y medianas empresas
- Soluciones específicas de industria

Las soluciones en la familia de Business Suite, permiten que una organización integre todos los aspectos de su negocio y colabore con entidades externas para llevar a cabo su trabajo, como cliente o proveedor. La base técnica de esta familia es SAP Netweaver.

La familia de Business Suite incluye 5 soluciones que se pueden instalar juntas o por separado:

- Enterprise Resource Planning (ERP) - El producto principal de la compañía en su versión actual 6.00. Desde el punto de vista funcional, SAP clasificó este producto en cuatro áreas claves: finanzas, recursos humanos, operaciones y servicios corporativos. Cuando se adquiere el ERP sólo se compra la licencia en base a los módulos que se activan.
- Customer Relationship Management (CRM) - Ayuda a las compañías a adquirir y mantener a sus clientes, ganar mercado y conocimiento del cliente. Facilita la coordinación con todas las tareas relacionadas a los clientes, como ventas, mercadotecnia, servicios al cliente, entre otros.
- Product Lifecycle Management (PLM) - Ayuda a los fabricantes con información relacionada al producto. Ayuda en el ciclo de vida del producto, desde la etapa de diseño hasta la de mercadeo, con lo cual los productos pueden producirse con mayor calidad y con costos eficientes.
- Supply Chain Management (SCM) - Ayuda a las compañías con el proceso de asignación de recursos para la fabricación y procesos de servicios. Ayuda a la organización a alinear sus existencias con la demanda del mercado y rápidamente adaptar sus operaciones a su cadena de suministro.
- Supplier Relationship Management (SRM) - Permite a las compañías trabajar en colaboración con sus proveedores que son vitales para la empresa. Ayuda a simplificar el proceso de abastecimiento, por ejemplo cuando un contrato con un proveedor está a punto de vencer, informa a los usuarios pertinentes para evitar demoras.
- Netweaver - Es la plataforma de tecnología que tiene otra serie de productos para la inteligencia de negocios (BW – Business Warehouse), seguridad y auditoría (GRC –

Governance, Risk and Compliance), manejo de información y conocimiento (Portal y KM – Knowledge Management), Plataforma SOA (PI – Process Integration), etc.

Cada uno de estos productos, complementa los espacios de optimización dentro de las organizaciones, para ayudarlas a ser competitivas a nivel mundial.

### 1.5. Requisitos actuales mínimos de un sistema ERP

Los módulos típicos del software comercial que cumple un ERP son: compras, manufactura, ventas, finanzas, recursos humanos, servicios y reportes en general. Existen una serie de puntos que se pueden visualizar con en el uso de los sistemas ERP, a continuación mencionamos algunas según [VK2005]:

Mejores prácticas de negocio integradas y conocimiento de negocio:

- Integración de procesos e información.
- Calidad en la información.
- Autorización y autenticación central.

Sin embargo, existen también riesgos potenciales comunes en el uso de sistemas ERP:

- Un solo punto de falla.
- Problemas integrando diferentes sistemas ERP.
- Seguimiento de procesos complejos, aun cuando se operan en más de un sistema.
- Operado por personal con educación inadecuada en el manejo de ERP.

Un sistema ERP moderno generalmente consta de dos capas, la de tecnología y la de aplicación, este tipo de sistemas hasta el momento se presentan como sistemas monolíticos. Como la demanda de software empresarial moderno no se puede cumplir con este tipo de arquitectura, el cambio a una arquitectura más flexible como SOA (Service-oriented Architecture), es el estatus quo de las compañías modernas.

La plataforma de SAP Netweaver, se presenta como una arquitectura que cumple con esas características de SOA y la cual ha ido evolucionando a partir de la plataforma de R/3 Basis, como se muestra en la gráfica 1.5-1.

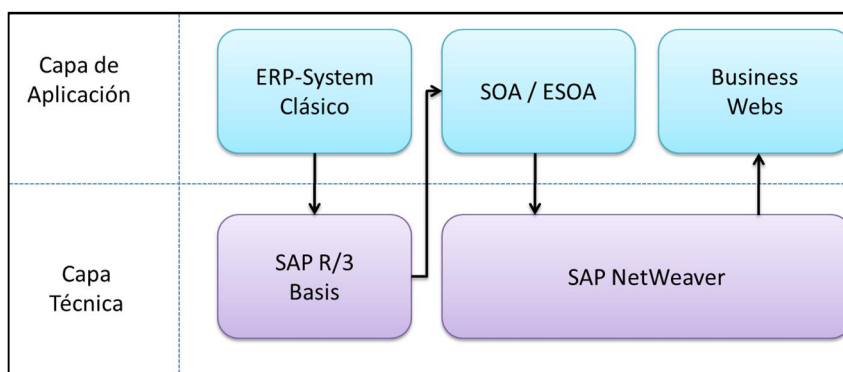


Fig. 1.5-1 Integración SOA con SAP

### 1.6. Componentes de un ERP

A continuación, introduciremos varios conceptos esenciales en el argot de los sistemas ERP [SAP01\_2009]. Estas definiciones deben ayudar a comprender mejor los ejemplos o referencias que se hacen en los siguientes temas, además de que introducen la arquitectura del sistema.



## Elementos Organizacionales.

Uno de los primeros pasos en la implementación de las aplicaciones de SAP, es retratar las estructuras organizacionales de la compañía en el sistema de SAP. Lo primero es analizar las estructuras y procedimientos de la compañía, para después empararlas con las estructuras de SAP.

- **Cliente.** El elemento cliente es el de mayor jerarquía en todos los elementos organizacionales, éste representa la empresa o la sede del grupo.
- **Código de compañía.** El código de compañía es una unidad de contabilidad legal independiente, que representa el elemento organizacional central de la contabilidad de finanzas. El código de compañía, también representa la vista de la ley de impuestos de una compañía, el calendario fiscal, la moneda local y los requerimientos de reportes de impuestos.
- **Planta.** Una planta puede manufacturar productos, distribuir productos, o proveer un servicio. La planta es la unidad organizacional central de producción.
- **Almacén.** El material en existencia en el manejo del inventario, se puede diferenciar dentro de una planta y se puede diferenciar de acuerdo al almacén.
- **Organización de ventas.** La organización de ventas, es el elemento organizacional central en el manejo de órdenes de venta que controla los términos de venta a un cliente.
- **Canal de distribución.** Los canales de distribución pueden ser usados para definir cómo diferentes productos llegan a los consumidores, por ejemplo, venta al mayoreo.
- **Unidad organizacional.** La unidad organizacional describe las unidades de negocio que existen en una empresa. Comúnmente las unidades se definen vagamente como departamento funcional o regional.
- **Posición.** Las posiciones son las asignaciones individuales de los empleados en la empresa.

En la figura 1.6-1 se muestra el ejemplo de la estructura organizacional de una empresa usando SAP. El software y la base de datos toman por tanto este modelo como base.

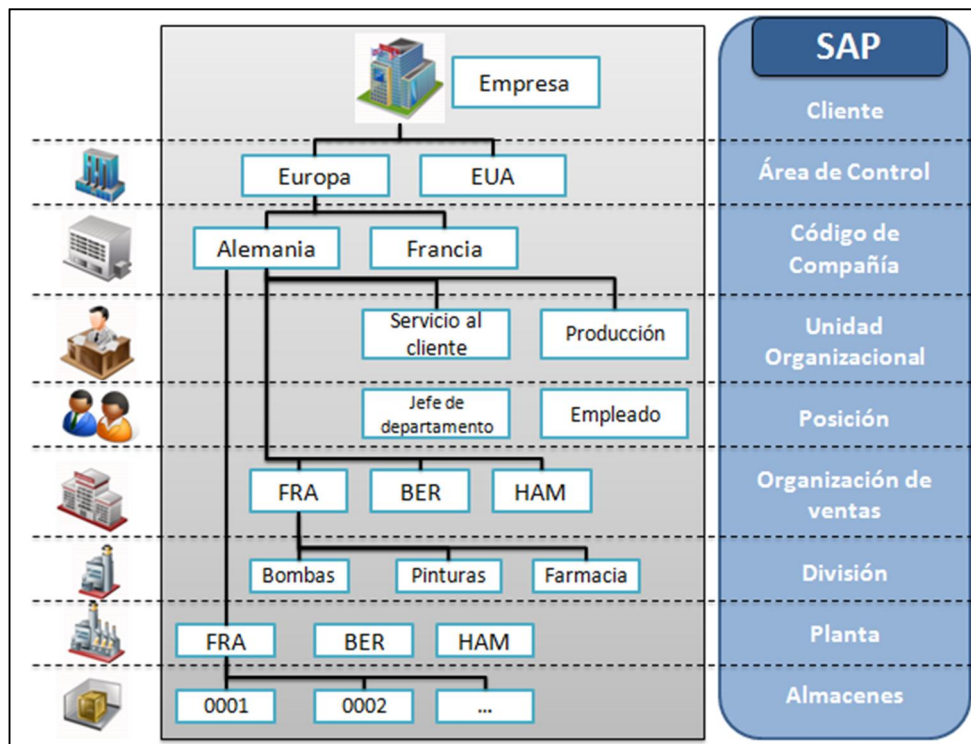


Fig. 1.6-1 Estructura Organizacional ERP SAP [SAP01\_2009]

**Datos Maestros.** Creados de manera centralizada y disponibles para todas las aplicaciones y todos los usuarios autorizados, los datos maestros tienen también un aspecto organizacional porque su información está ordenada en vistas que son asignadas a elementos organizacionales.

En el ERP de SAP los datos maestros son creados de manera centralizada y están disponibles de esta manera a todos los usuarios que tienen accesos a ellos de acuerdo a su perfil. El hecho de que estén centralizados significa que siempre están consistentes, actualizados y libres de redundancia.

- *Datos maestros del cliente.* Define la relación de negocios entre la compañía y su cliente y se representa en una estructura de tres partes:
  1. Información general
  2. Información para los códigos de compañía
  3. Información para ventas

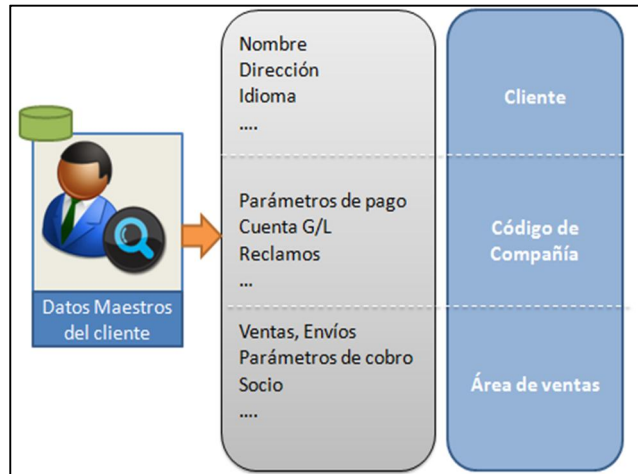


Fig. 1.6-2 Datos maestros del cliente

- *Maestro de materiales.* Contiene toda la información clave que una compañía necesita para el manejo de un material dentro de la compañía. Éste define entre otras cosas cómo el producto se vende, se compra, se manufactura, se maneja su inventario, contabiliza y se calcula. La mayoría de materiales se agrupan por nivel de planta.

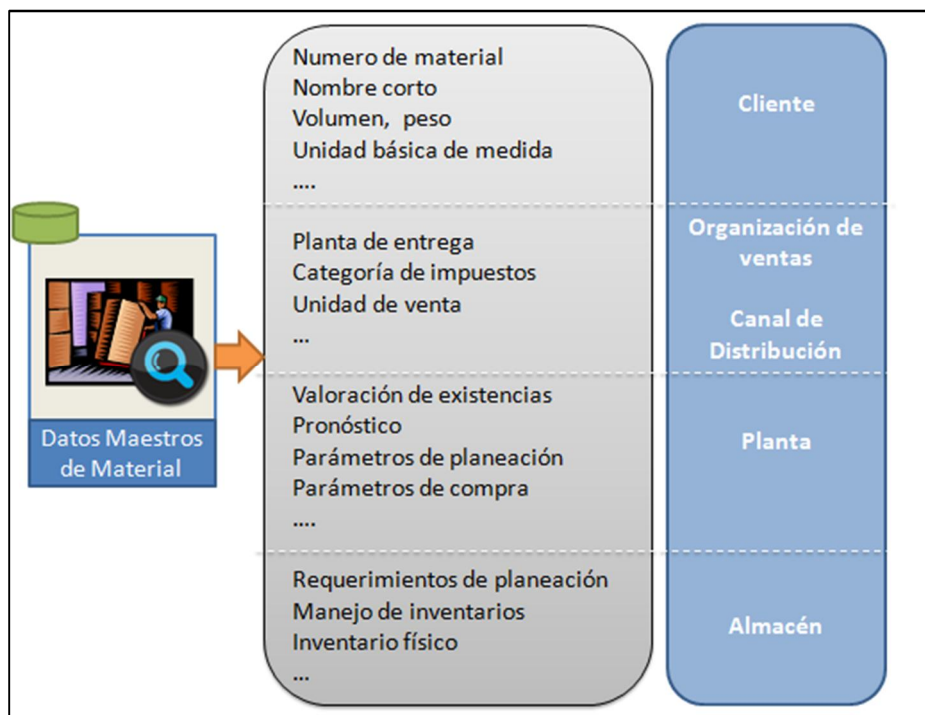


Fig. 1.6-3 Maestro de materiales

**Transacciones.** Las transacciones son programas de aplicación que ejecutan un proceso de negocio en el sistema ERP de SAP. Algunos ejemplos incluyen crear orden de venta, cambiar la información del cliente, o desplegar un reporte con la lista de las órdenes de venta.

Siempre que se ejecuta una transacción un documento se crea, este documento graba la información de la transacción, y contiene toda la información predefinida de los datos maestros y los elementos organizacionales.

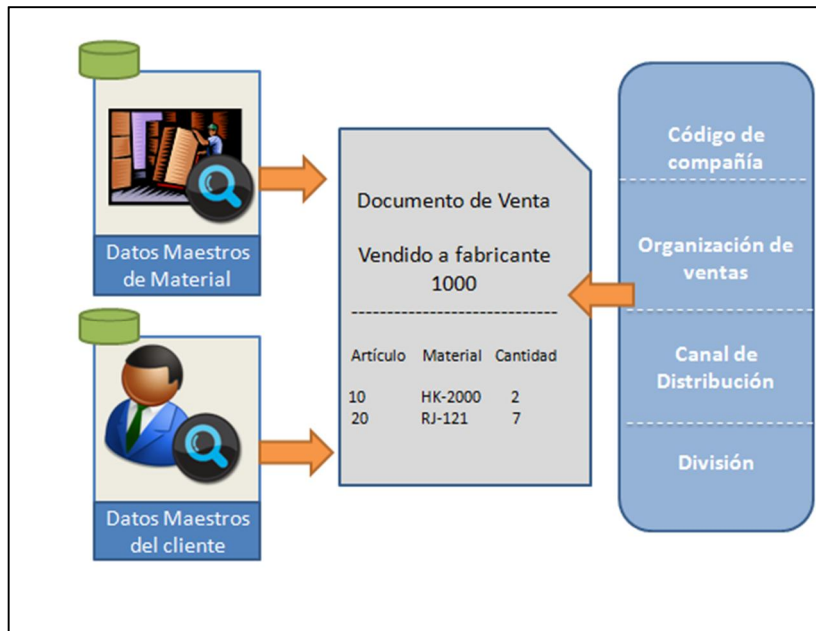


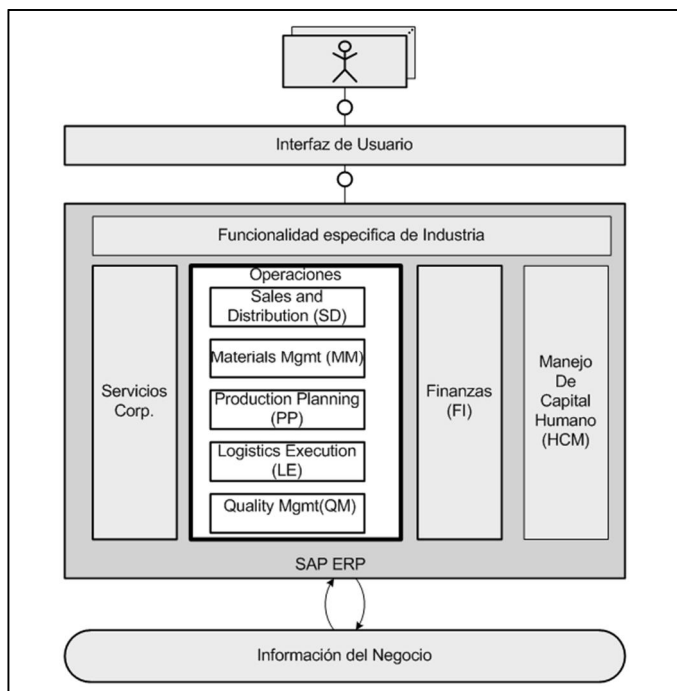
Fig. 1.6-4 Transacciones dentro de SAP

### 1.6.1. Componentes del ERP de SAP

Para lograr una coordinación exacta y una excelencia operacional los procesos de abastecimiento, logística, desarrollo de producto, manufactura, así como también en ventas y servicios tienen que trabajar de la mano. Para poder soportar estos procesos punto a punto por todas las áreas, SAP proporciona el SAP ERP Operaciones (SAP ERP OPS) como parte del SAP ERP, como se muestra en la figura 1.6.1-1. Además de SAP ERP OPS, el SAP ERP incluye las siguientes aplicaciones:

- Servicios Corporativos
- Finanzas (FI)
- Manejo de Capital Humano (HCM)
- Funcionalidad específica de industria

Todas las aplicaciones comparten una misma base de datos, sin embargo es común instalar ERP HCM en un sistema distinto con su propia base de datos. SAP ERP FIN, SAP ERP OPS y Servicios Corporativos siempre están dentro de un mismo sistema como se muestra en la siguiente figura:



**Fig. 1.6.1-1 Arquitectura de módulos en SAP ECC [SOA200\_2009]**

El ERP de SAP cuenta con una solución esencial llamada (SAP ERP OPS) para la obtención, venta, producción, almacenaje, envío, y transportación de material. SAP ERP OPS es parte del SAP ERP, que también incluye SAP ERP Finanzas y SAP ERP HCM. SAP ERP OPS consiste en los siguientes componentes [SAP01\_2009]:

- Ventas y Distribución (SD)
- Planeación de Producción (PP)
- Manejo de Materiales (MM)
- Logística (LE)
- Manejo de Calidad (QM)

Usando SD los productos se venden y mandan a los socios de negocios o ellos mismos realizan los servicios. Información acerca de los productos, servicios y socios de negocio es la base para el proceso de venta. El proceso de venta está basado en un conjunto de documentos de negocio como cuestionarios al cliente, cotizaciones, y lo más importante la orden de venta. Estos documentos los mantiene y guarda el módulo de SD. Las actividades que se deben de continuar se basan en ellos. El cálculo de precios para materiales y servicios está basado en la condición técnica. Ventas y distribución incluye la creación de órdenes de compra, revisar la disponibilidad de los productos requeridos, cálculo de precios, y finalmente la entrega y facturación.

PP ayuda al desarrollo y ejecución de planes efectivos de producción, los cuales toman en cuenta al almacén, material, y capacidad de producción como información, así como también la planificación de la venta. Cuando el plan de producción es transformado a una orden de producción para inicializar la ejecución, PP reserva y obtiene las materias primas usando MM. Además del maestro de materiales, la información maestra más importante para PP es el BOM (Bill of Material), el cual lista todos los componentes requeridos para manufacturar un cierto producto.

MM proporciona las funciones que son necesarias para trabajar con los flujos de mercancías y servicios entrantes. MM soporta el abastecimiento usando los documentos de negocio de solicitud de pedido y subsecuentemente la orden de pedido. La llegada de las mercancías ordenadas da como resultado la recepción de mercancías y es manejado por el manejo de inventario. MM mantiene el inventario completo de toda la compañía, lo cual significa que cualquier salida de mercancía también se notifica a MM. Manejo de Materiales incluye la creación de órdenes de compra para la obtención de materiales y servicios de proveedores externos, verificar las facturas entrantes y manejar el inventario.

LE controla y organiza el movimiento de materiales dentro de la compañía (manejo del almacén), y también la transportación entre empresas. Los documentos centrales de negocio, los cuales son actualizados y guardados por LE son la entrega, el documento de envío, y la orden de transferencia. Estos son utilizados para lanzar y controlar el movimiento de material.

QM permite planear y conducir inspecciones para revisar las características de calidad de un material. Está integrado en el abastecimiento, producción y proceso de ventas.

Para asegurar el proceso de negocio a través de los diferentes componentes de SAP ERP OPS basados en información consistente, estos están estrechamente integrados usando integración a nivel de base de datos. De esta manera la comunicación A2A (Application-to-Application) se usa la integración a nivel BD. Usando esta misma mecánica se comunica con ERP Finanzas. Todos los componentes comparten el mismo maestro de materiales.

En la imagen de arquitectura 1.6.1-2 observamos la interacción de los diferentes componentes de SAP ERP Operaciones con el proceso industrial generalizado. Esta generalización se debe y puede adaptar a los requerimientos del negocio y ahí es donde la consultoría y desarrollo para el cliente entra en acción. El producto ERP de SAP es sólo una generalización de los procesos de negocio que se tienen que adaptar para cumplir con las expectativas del cliente.

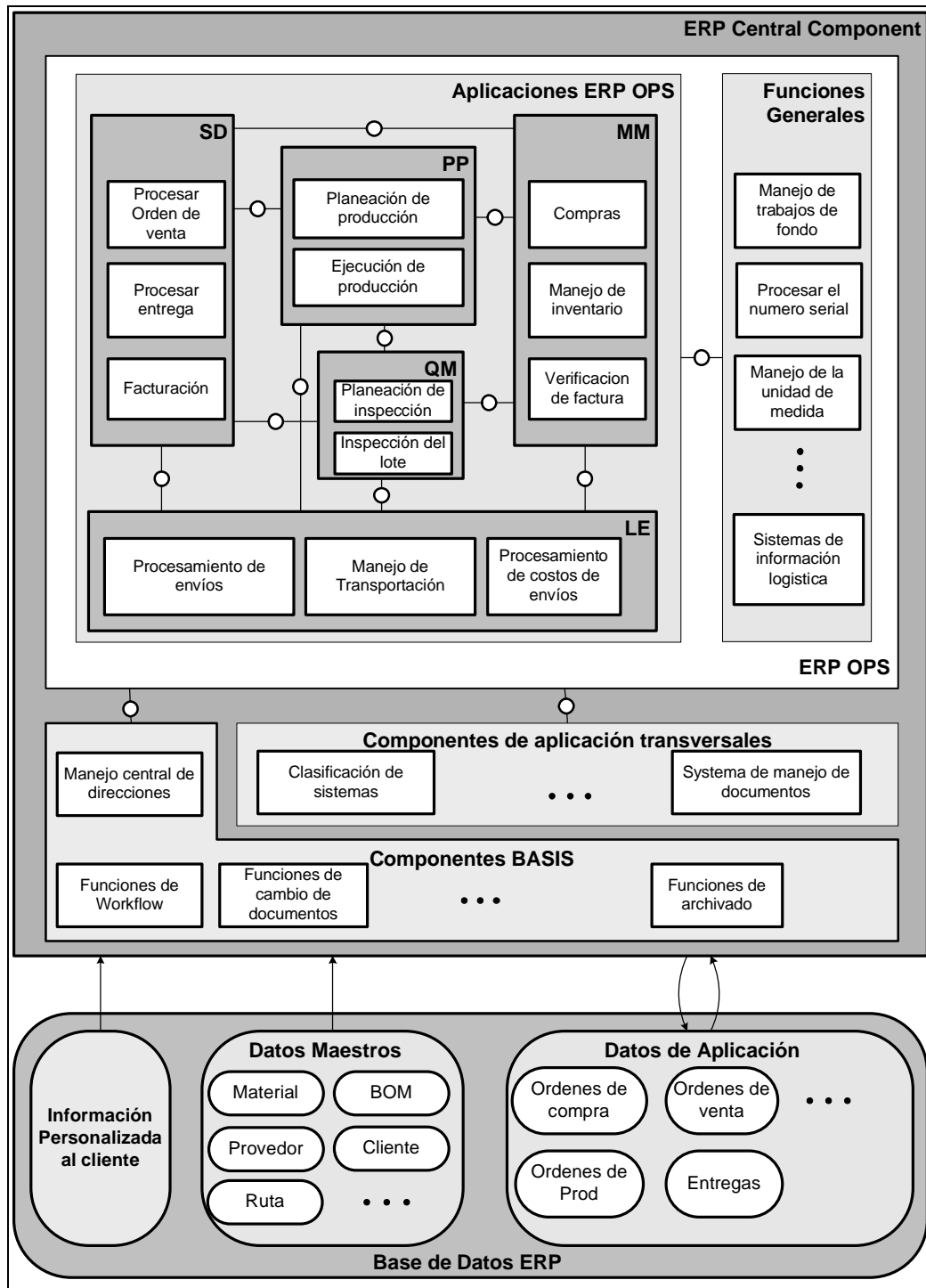


Fig. 1.6.1-2 Arquitectura ECC de SAP ECC para operaciones [SOA200\_2009]

### 1.7. SAP Netweaver como plataforma de adaptación y preparación de los procesos de negocio

La idea que se presentó sobre las arquitecturas orientadas a servicios y sus usos en el dominio de las empresas obliga a SAP al desarrollo de una plataforma tecnológica que permita la integración de esos diferentes servicios.

SAP desarrolló una plataforma para cumplir con ESOA (Enterprise Service-Oriented Architecture) llamada Netweaver, la cual representa la base técnica que habilita las aplicaciones de negocios de SAP, sus socios y clientes. Casi todos los productos de SAP están basados en el servidor de aplicación Netweaver de SAP (SAP Web AS por sus siglas en inglés).

SAP Netweaver es una plataforma basada en web de integración y aplicación. Sirve como base para una arquitectura SOA y que permite la alineación e integración de personas, información y procesos de negocio esparcidos por diferentes plataformas tecnológicas.

SAP Netweaver se desarrolla como requerimiento de los constantes cambios en las necesidades de negocio y fue una evolución de SAP Basis, luego SAP Web Application Server (SAP Web AS) para culminar en SAP Netweaver. El SAP Web AS sigue siendo la plataforma tecnológica para las soluciones de SAP, pero ahora ha sido combinado con otros componentes de integración para constituir Netweaver.

SAP Netweaver soporta tecnologías como .NET, Java 2 Enterprise Edition (J2EE) y Websphere. Soportando la integración de sistemas SAP con sistemas no SAP.

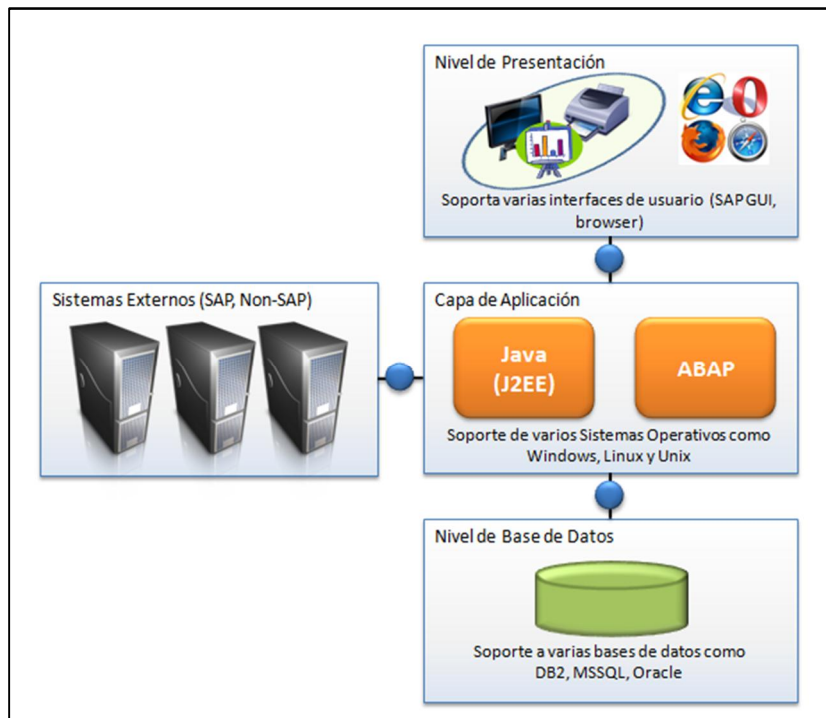


Fig. 1.7-1 Arquitectura de capa Netweaver

El Servidor de Aplicación de SAP Netweaver (SA) ofrece:

- Una arquitectura multinivel
- Un ambiente confiable y ampliamente probado para aplicaciones ABAP y Java
- Gran escalabilidad
- Independencia de sistema operativo y base de datos
- Un ambiente conveniente de desarrollo
- Soporte de varios protocolos (RFC, HTTP)
- Altos estándares de seguridad para procesos de negocio complejos

### 1.7.1. Modelo de SAP Netweaver como arquitectura de la empresa (people view, organizational view, information view, business process view, information technology view)

SAP Netweaver es también una plataforma de integración basada en estándares abierto de integración. SAP Netweaver habilita las siguientes prácticas de TI [SAP01\_2009]:

- Unificación de la información
- Manejos de la información de negocio
- Integración de procesos punto a punto
- Desarrollo del cliente
- Manejo del ciclo de vida unificado
- Consolidación
- Diseño e implementación ESOA
- Manejo de la seguridad y gobierno de las Aplicaciones

En el siguiente diagrama observamos los componentes principales de Netweaver:

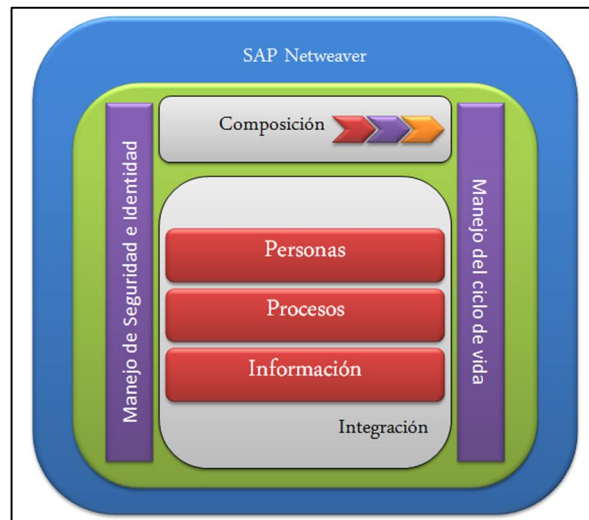


Fig. 1.7.1 Arquitectura de empresa Netweaver [SAP01\_2009]

SAP Netweaver tiene muchas capacidades, las cuales se pueden clasificar en cuatro habilitadores:

➤ Integración de personas.

La integración de personas asegura que los empleados tengan acceso a la información y funciones que ellos necesitan para realizar sus actividades de manera rápida y eficiente. Las funciones de SAP Netweaver Portal son las más importantes en este contexto. SAP Netweaver Mobile sirve como plataforma universal para la movilidad.

➤ Integración de información.

La capa de integración proporciona acceso a toda la información estructurada y no estructurada dentro de la compañía. Las áreas claves para la integración de información se conforman por SAP Netweaver BW (Business Warehouse), KM (Knowledge Management) y SAP MDM (Master Data Management) que ayuda a consolidar la información dentro de la organización a través de sus fronteras, es decir, que aunque la información esté guardada en diferentes lugares por motivos técnicos, tengamos sincronizadas las bases de datos con la información necesaria, sólo así se logra que todos los sistemas que se conectan obtengan la misma información. SAP BW ayuda a ver los brutos de información desde diferentes ángulos y hacer las decisiones apropiadas. BW y MDM son para información estructurada y KM para información no estructurada como archivos con información variada.

➤ Integración de procesos.

La integración de procesos asegura que los procesos de negocio trabajen a través de las fronteras en un ambiente heterogéneo de sistemas. SAP Netweaver PI (Process Integration) juega un rol crucial y conecta los sistemas SAP con los sistemas no SAP. Principalmente se usan estándares abiertos del mundo Java y XML.

Esta capa representa los elementos de "Integration Broker" y BPM (Business Process Management) Manejo de procesos de negocio lo cual forman el producto llamado PI. El "Integration Broker" es el encargado del intercambio de información en diferentes formatos y protocolos, además de esta conversión técnica también cambiar el contenido convirtiendo valores, por ejemplo Mujer por Sra. El Manejador de Procesos de Negocio, BPM, proporciona las funciones de manejo de flujos en los procesos a través de los sistemas. Se puede utilizar para automatizar procesos coordinando y disparando el intercambio de información entre los sistemas que pueden acceder al Integration Broker.



➤ Plataforma de integración

El Servidor de Aplicación de SAP proporciona una infraestructura probada para las operaciones de las aplicaciones de negocio que están basadas en las tecnologías claves como J2EE y ABAP. Aparte del ambiente de ejecución de J2EE y de ABAP, se proporcionan las herramientas correspondientes de desarrollo para apoyo a la implementación de las aplicaciones. Estándares abiertos, acceso a aplicaciones basado en navegadores web y servicios web complementan la plataforma de aplicación. En esta capa siempre se abstrae tanto el sistema operativo como la base de datos como parte elementas del servidor de aplicación.

Componentes verticales dentro de SAP Netweaver:

➤ Manejo del ciclo de vida

Soporta el diseño, desarrollo, implementación y manejo de cambios de la solución entera. Cubre todas las actividades para ejecutar aplicaciones empresariales, que incluye la preparación para la instalación, actividades de personalización del software, monitoreo, mantenimiento, actualización y tareas de migración. El software que soporta todas estas tareas es "Solution Manager" que proporciona parte de estas funciones como primer paso para una implementación que cumpla con ITIL, algunas de estas funciones se cubren el capítulo 2.

➤ Composición (Composite Application Framework)

Está integrado en SAP Netweaver y permite a sus socios de negocio utilizar herramientas, estructuras, reglas y métodos para crear nuevas aplicaciones que estén basadas en procesos de negocios a través de áreas funcionales. El CAF, por sus siglas en inglés, permite el desarrollo de aplicaciones en los servicios subyacentes, servicios web o módulos de funciones SAP. La implementación de una capa persistente así como el diseño de interfaces son también parte del desarrollo. CAF puede ser usado para que los desarrolladores generen aplicaciones basadas en Java.

## 2. Metodología de implementación

La decisión de una empresa que opta por iniciar un proyecto de implementación de ERP, se da como consecuencia de la identificación de múltiples variables y necesidades del negocio, que serán atendidos con esta solución.

Uno de los objetivos principales al iniciar un proyecto de este tipo, es determinar de entrada cuáles son las motivaciones que tiene por una parte la compañía para iniciar con un proyecto de tal magnitud, así como conocer también cuáles son sus principales necesidades, es importante identificar cuáles son los beneficios tangibles que traerá la implementación de la solución, así como cuál será el impacto a nivel organizacional que se tendrá una vez que el ambiente salga a producción y la operación del negocio esté corriendo sobre la nueva plataforma.

En la actualidad podemos encontrar un amplio conjunto de productos enfocados a dar atención a las diversas áreas de negocio de la empresa de manera independiente, estos paquetes están enfocados a capturar, almacenar, procesar y distribuir enormes cantidades de información para departamentos o áreas diversas de la organización. Como lo hemos identificado durante la experiencia profesional, este tipo de soluciones, ha traído consigo problemas para las grandes empresas asociados a la duplicidad de datos, la dificultad de acceso a los mismos, la falta de integridad, así como la imposibilidad de compartir esta información en línea y en tiempo real, lo cual impide el oportuno conocimiento y control de la operación del negocio para la gestión de la empresa, trayendo consigo que en ocasiones, la información solicitada no sea del todo oportuna y precisa, como el ambiente actual lo demanda [WISP2011].

Es aquí cuando la implementación de una solución ERP cobra valor, el hecho de tener la información disponible en línea de manera integral, unificada y accesible, ha dado gran valor a las organizaciones que deciden implementar un ERP.

La experiencia nos ha demostrado que este tipo de soluciones, se enfocan en apoyar a los negocios en la gestión empresarial, la integración de las diversas partes del negocio, así como servir de plataforma base y soporte para los aspectos más críticos de la administración, llámese finanzas, producción, ventas, recursos humanos entre otras áreas.

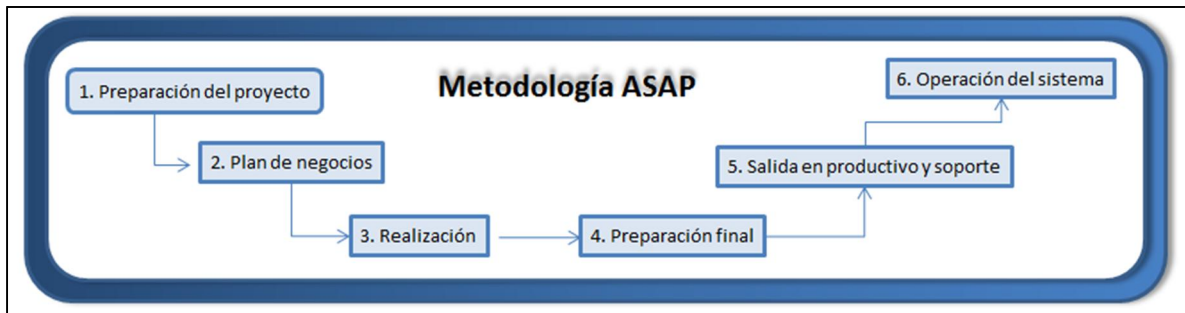
Las implementaciones de SAP ofrecen una amplia gama de productos enfocados a empresas de los más diversos giros, buscando entre sus principales objetivos, permitir la evaluación, implementación, automatización, integración y gestión de las diversas áreas de la empresa que permitan ejecutar y llevar a cabo las actividades del día a día.

A continuación, se realizará una descripción detallada de algunas de las principales metodologías utilizadas que se han desarrollado a lo largo de los años para realizar la implementación de SAP de la forma más organizada, óptima y exitosa.

### 2.1. Metodología ASAP para la implementación de la solución

ASAP o "Accelerated SAP", es una metodología desarrollada por la empresa SAP AG que establece los mecanismos para implementar una solución SAP en base a fases claras, entregables bien definidos, así como recomendaciones para ejecutar el proyecto de implementación de una forma eficaz en los menores tiempos posibles [SHAR2011].

Inicialmente ASAP fue desarrollada con el objeto de establecer las normas para la implementación de soluciones del tipo ERP únicamente. Actualmente, esta metodología va más allá y es utilizada para la implementación de otras tecnologías y soluciones tales como la Inteligencia de negocios, la implementación de portales de intranet y extranet, sistemas del tipo CRM, entre otras soluciones de negocio. La figura 2.1-1 muestra las principales etapas de las que se compone la metodología ASAP.



**Fig. 2.1-1 Metodología de implementación ASAP [SCR2011].**

Los principales beneficios de ASAP comprenden la reducción de costos totales del proyecto, mediante la incorporación de entregables bien definidos y un plan de trabajo modular con fechas y responsables claros para cada una de las actividades, así como el hecho de cubrir el ciclo de vida completo del proyecto, desde las fases iniciales de análisis y evaluación, hasta la ejecución y operación completa del sistema.

La figura 2.1-2 muestra algunos aspectos importantes que es necesario considerar como parte del ciclo de vida de un proyecto de implementación de TI, en general es indispensable considerar la etapa de preparación inicial del sistema, el equipo de trabajo así como la información misma que será procesada.



**Fig. 2.1-2 Ciclo de vida de un proyecto de TI.**

Las fases planteadas por la metodología ASAP se establecen por medio de un documento conocido como "Roadmap" u hoja de rutas, en el cual se detalla cada una de las actividades necesarias para llevar a cabo la implementación.

Las fases que comprende ASAP son las siguientes [SCR2011]:

1. Preparación del proyecto.
2. Plano de negocios, conocido también como "business blueprint".
3. Realización.
4. Preparación final.
5. Salida en productivo y soporte.
6. Operación del sistema.

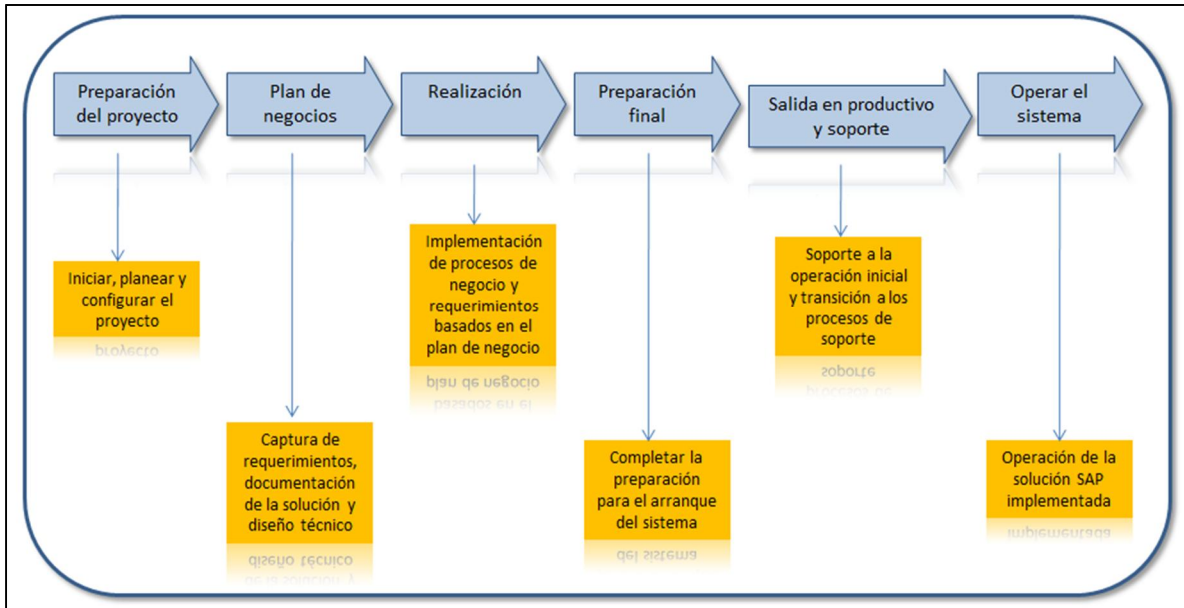


Fig. 2.1-3 Fases de la metodología ASAP

La figura 2.1-3 muestra cada una de las etapas comprendidas en esta metodología, en los capítulos siguientes, se describirá con mayor detalle en qué consiste cada una de ellas así como los principales aspectos que se deben tomar en cuenta, para el desarrollo de cada una de estas fases.

### 2.1.1. Preparación del proyecto

Esta fase sirve para iniciar con la planeación y preparación del proyecto de implementación [SCR2011]. Se busca entre otros aspectos, establecer cuáles serán las metas, alcances y objetivos del proyecto, así como la identificación de todos aquellos recursos que participarán tanto del lado del cliente, como por parte de la empresa proveedora de consultoría con especialización en la implementación de soluciones ERP SAP, como se muestra en la siguiente figura:

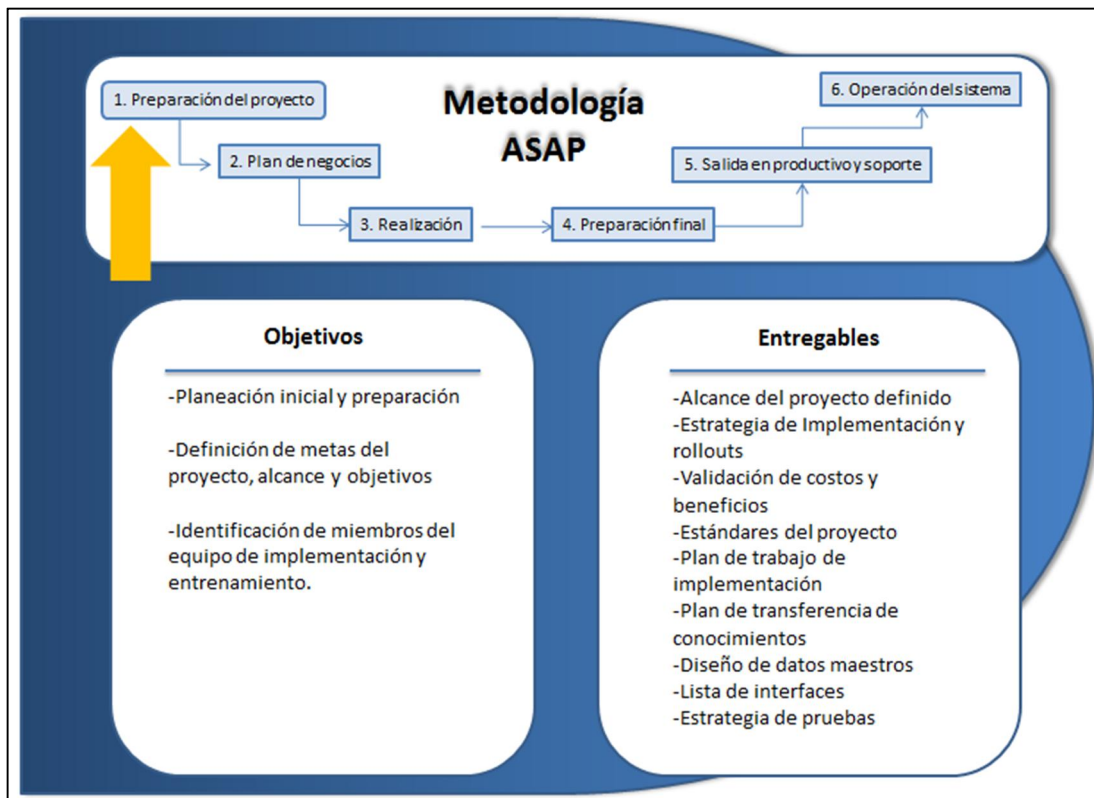


Fig. 2.1.1-1 Preparación del proyecto [SCAM2011].

Como parte de los entregables que se generará durante esta etapa, se encuentra la creación del plan de proyecto a un alto nivel, en el cual será necesario determinar los costos y beneficios asociados a la implementación de SAP, de la misma manera, se requiere iniciar con el proyecto de implementación de la infraestructura técnica que dará soporte a la plataforma, y que servirá como base para la instalación de cada uno de los componentes de cómputo requeridos para mantener la solución operando de la manera más adecuada.

Para dar inicio con el proyecto de manera oficial, se programará una reunión de arranque con todos los miembros que participarán en el proyecto, en esta junta se realizará el planteamiento de las metas y objetivos por alcanzar, así como la importancia de éstos en el futuro de la compañía.

De la misma manera, será necesario contemplar una fase inicial de entrenamiento enfocada a los empleados por parte del cliente, con el objeto de dar a conocer la plataforma SAP así como las tecnologías asociadas que se estarán utilizando durante el desarrollo del proyecto.

### 2.1.2. Plan de negocio

El objetivo de la fase de plano de negocios es entender cuáles son las metas del cliente, cuáles son los procesos de negocio que se implementarán en SAP, así como conocer de qué manera opera la empresa actualmente, para que a partir de este análisis, sea posible la generación de un plan de negocios o "business blueprint", por medio del cual se establecerá de qué forma operará el negocio una vez que se libere el sistema y se programe la salida en vivo de SAP [SCAM2011]. En la siguiente figura, se muestran los principales objetivos y entregables que es necesario alcanzar durante esta etapa:

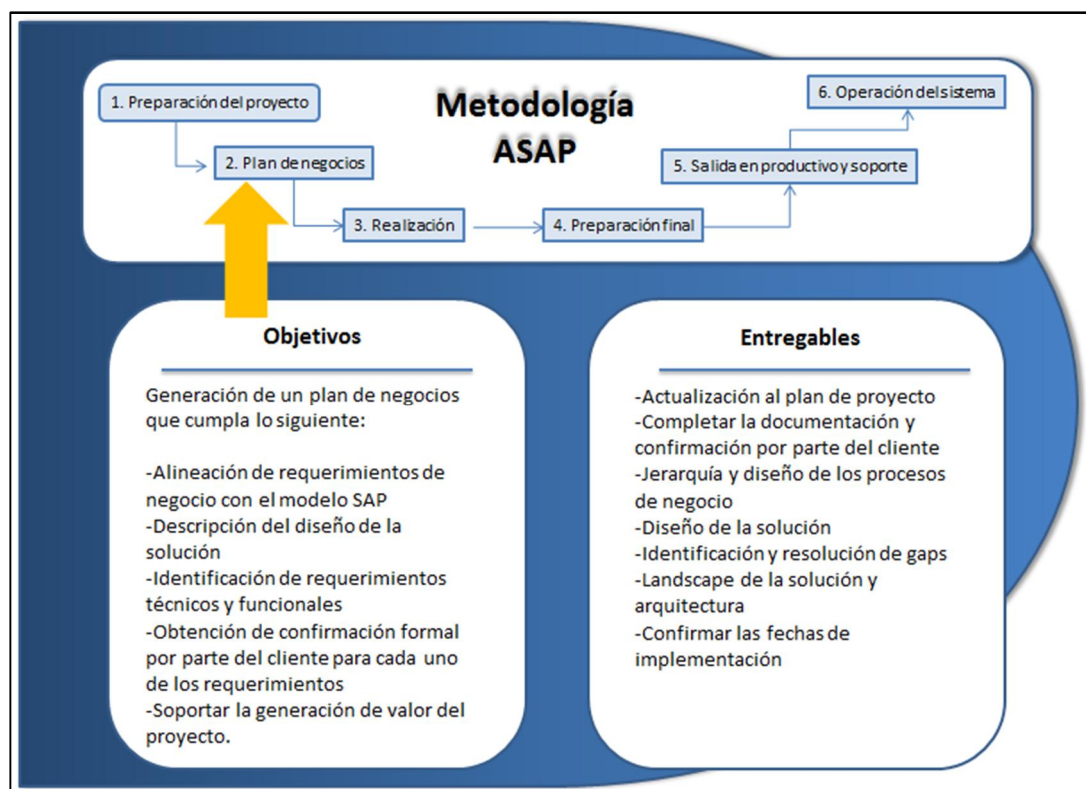


Fig. 2.1.2-1 Plan de negocios [SCAM2011].

Por medio de reuniones ejecutivas, se discutirán las metas del grupo, la estructura organizacional, así como también los procesos de negocio de alto nivel.

Posteriormente y una vez que se tengan identificados dichos procesos, se programarán reuniones con los líderes y responsables de estos procesos de negocio con el objetivo de conocer claramente los requerimientos funcionales, dichos requerimientos deberán ser firmados por los líderes del proyecto por parte del cliente, para en base a ellos considerar los tiempos necesarios de implementación que requerirá la empresa consultora, para que con estas fechas, sea posible plasmar el detalle de las actividades en el plan de trabajo.

De la misma manera, será importante buscar que cada uno de los procesos de negocio, sean estructurados y desarrollados en base a la funcionalidad estándar proporcionada por SAP, la empresa consultora deberá documentar el diseño y propuesta de solución para cada uno de los requerimientos, con el objetivo de que sean avalados y firmados por las áreas tanto de negocio como de tecnología por parte del cliente.

Como parte de las actividades, será necesario considerar un plan de comunicación por medio del cual se enviará un estatus de las actividades desarrolladas a todos los miembros del equipo, con el objetivo de mantenerlos actualizados con el avance en cada una de las fases del proyecto.

Esta etapa finalizará con la construcción, entrega y aceptación por parte del cliente del plano de negocio, el cual deberá ser generado tanto de forma escrita, como por medio de diagramas para exponer la estrategia de la manera más clara posible.

### 2.1.3. Realización

Como se muestra en la figura anexa, esta fase está enfocada a realizar la configuración, desarrollo e implementación de los requerimientos funcionales plasmados en el plano de negocios, de la misma manera, se programará la fase de pruebas de funcionalidad que incluirá una etapa inicial en donde se probará la solución de manera unitaria, para posteriormente iniciar con la ejecución de las pruebas integrales sobre todos los procesos de negocio [SCAM2011].

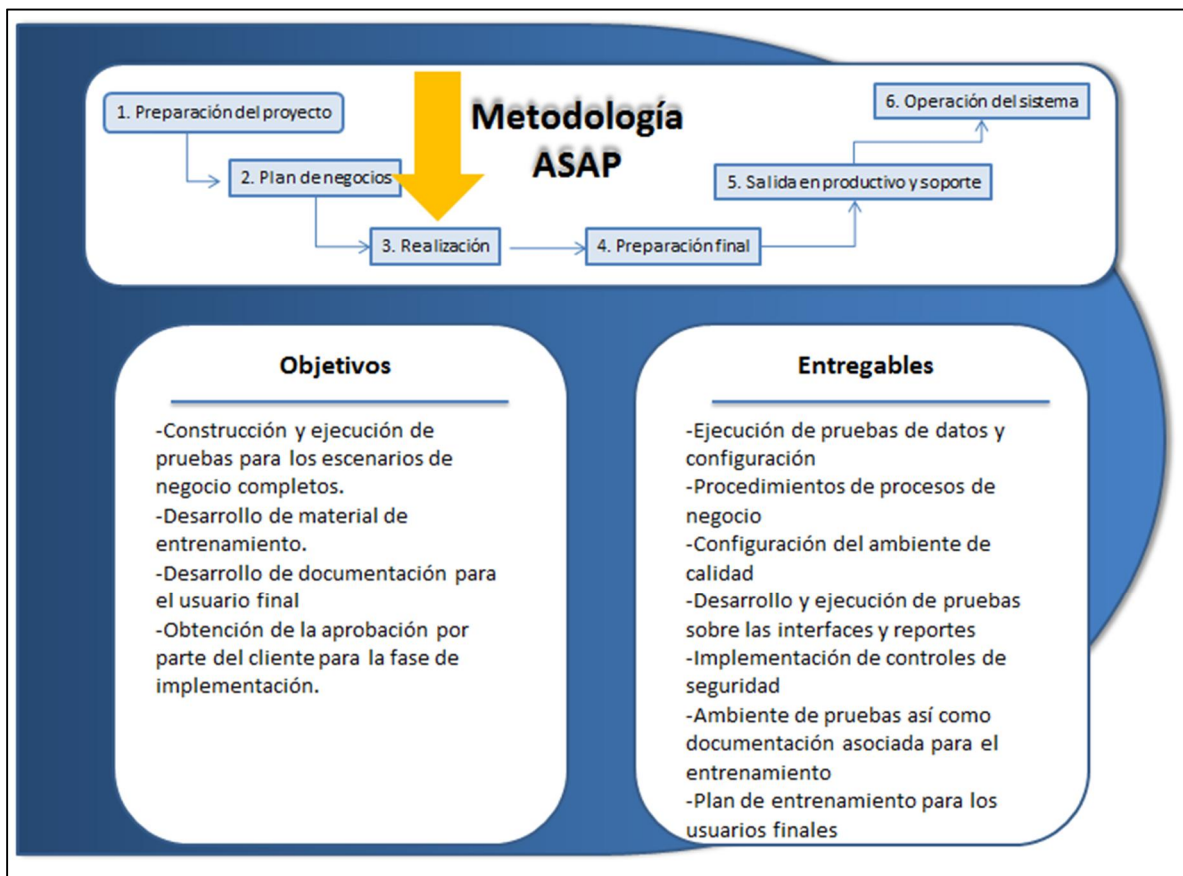


Fig. 2.1.3-1 Realización [SCAM2011].

El equipo de proyecto por parte del cliente, continuará bajo un esquema de capacitación de la solución, lo cual le permitirá adquirir un conocimiento más profundo de la nueva plataforma, y empezar a modelar los casos de negocio sobre SAP, la capacitación permitirá al equipo conocer con mayor detalle las herramientas, así como la integración intrínseca de sus diversos componentes.

Mientras los recursos del cliente continúan con la capacitación, el equipo de consultoría estará enfocado a la configuración de los procesos de negocio definidos en el sistema, empezando a reflejar ya la organización solicitada por el cliente.

Se realizará la carga de los datos maestros y se iniciarán con las pruebas en conjunto con los líderes de los procesos de negocio, para confirmar la funcionalidad establecida en el plano de negocios.

A la par de estas actividades, se estará trabajando en el desarrollo de reportes, interfaces y desarrollos solicitados por el cliente con funcionalidad fuera del estándar de SAP, se iniciará con la construcción de los escenarios y se trabajará en coordinación con el equipo de seguridad para realizar la creación de los roles y perfiles, para cada una de las aéreas de negocio, los cuales se utilizarán para establecer las restricciones de seguridad requeridas, en función de la posición que tendrán los usuarios finales que ingresen al sistema.

Durante las fases de pruebas, los consultores trabajarán conjuntamente con el equipo del cliente con el objetivo de iniciar con la transferencia de conocimientos, así como generar la documentación que será utilizada para la capacitación hacia los usuarios finales.

#### 2.1.4. Preparación final

El propósito de esta fase, será finalizar con los ciclos de pruebas integrales que confirmará la funcionalidad del sistema, adicionalmente será necesario programar una serie de pruebas en las cuales se buscará estresar los sistemas con el objeto de probar el performance de los ambientes instalados, en esta etapa se identificarán posibles áreas de oportunidad para la infraestructura de cómputo y con esto realizar las afinaciones necesarias para los sistemas ya sea a nivel aplicación, base de datos o sistema operativo [SCAM2011].

La figura 2.1.4-1 describe los principales objetivos que debe considerar el equipo de implementación durante el desarrollo de la preparación final.

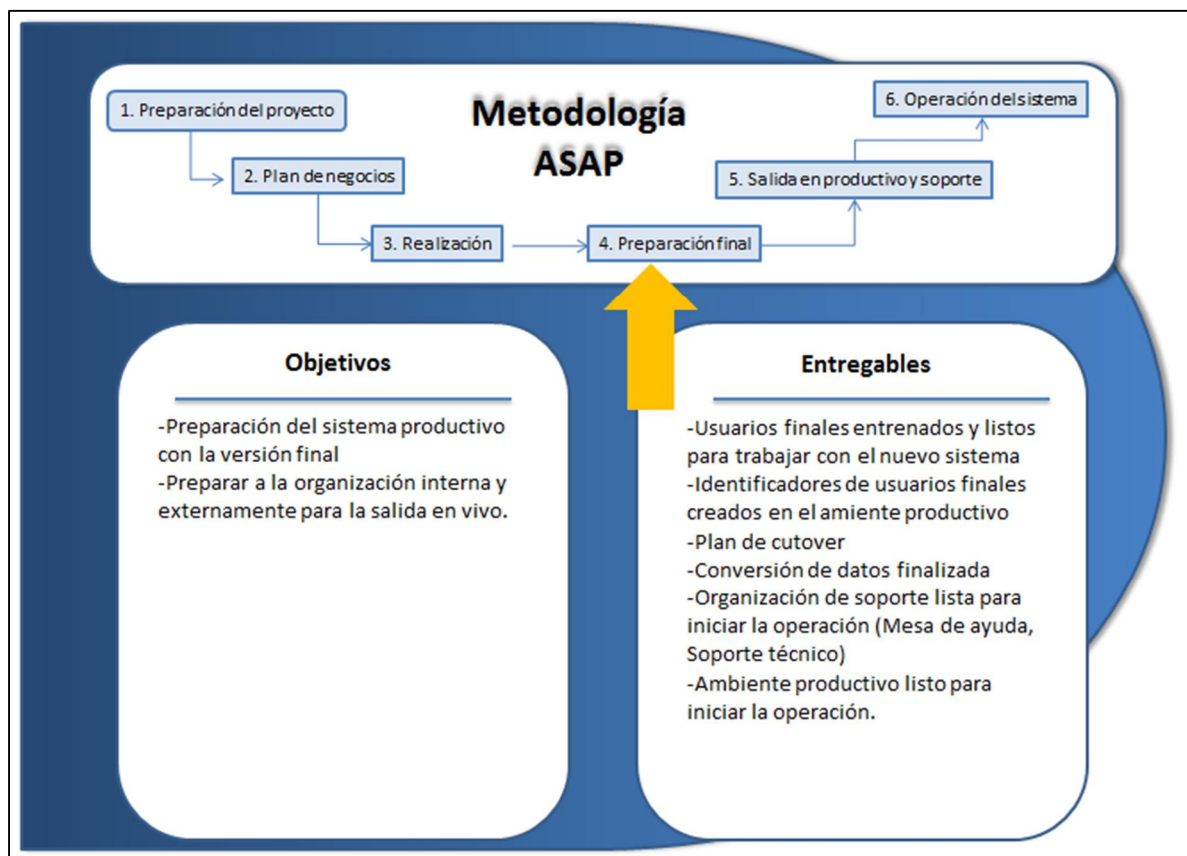


Fig. 2.1.4-1 Preparación final [SCAM2011].

Los usuarios finales que estarán trabajando con el sistema una vez que salga a operación serán capacitados durante este periodo, en este caso se seguirá la estrategia de capacitar a usuarios clave por parte del cliente, los cuáles impartirán a su vez la capacitación final al mismo personal de la empresa, para con esto ayudar a generar la confianza y aceptación del nuevo sistema, y

adicionalmente generar una base de conocimiento interna para el soporte e identificación de futuras mejoras a la solución.

El último paso será que el cliente apruebe el sistema y por medio de una reunión de “go/no-go” se dará el visto bueno para las fechas horarios para la puesta en marcha en base a un plan de “cutover”, en donde se planteará hora a hora las actividades y responsables asociados a éstas antes de la salida en producción de SAP. Es importante que esta actividad de cutover o de tareas de cierre se ejecute con precisión de lo contrario afecta las salidas en productivo y por tanto el tiempo-costado del proyecto.

### 2.1.5. Salida en productivo y soporte

Después de la puesta en marcha del sistema, como se observa en la figura 2.1.5-1, se deberá establecer un esquema de soporte de atención inmediata durante los primeros días, con el objetivo de dar atención de la manera más eficiente a todos los incidentes que se vayan generando durante la operación, así como estabilizar la solución recién implementada [SCAM2011].

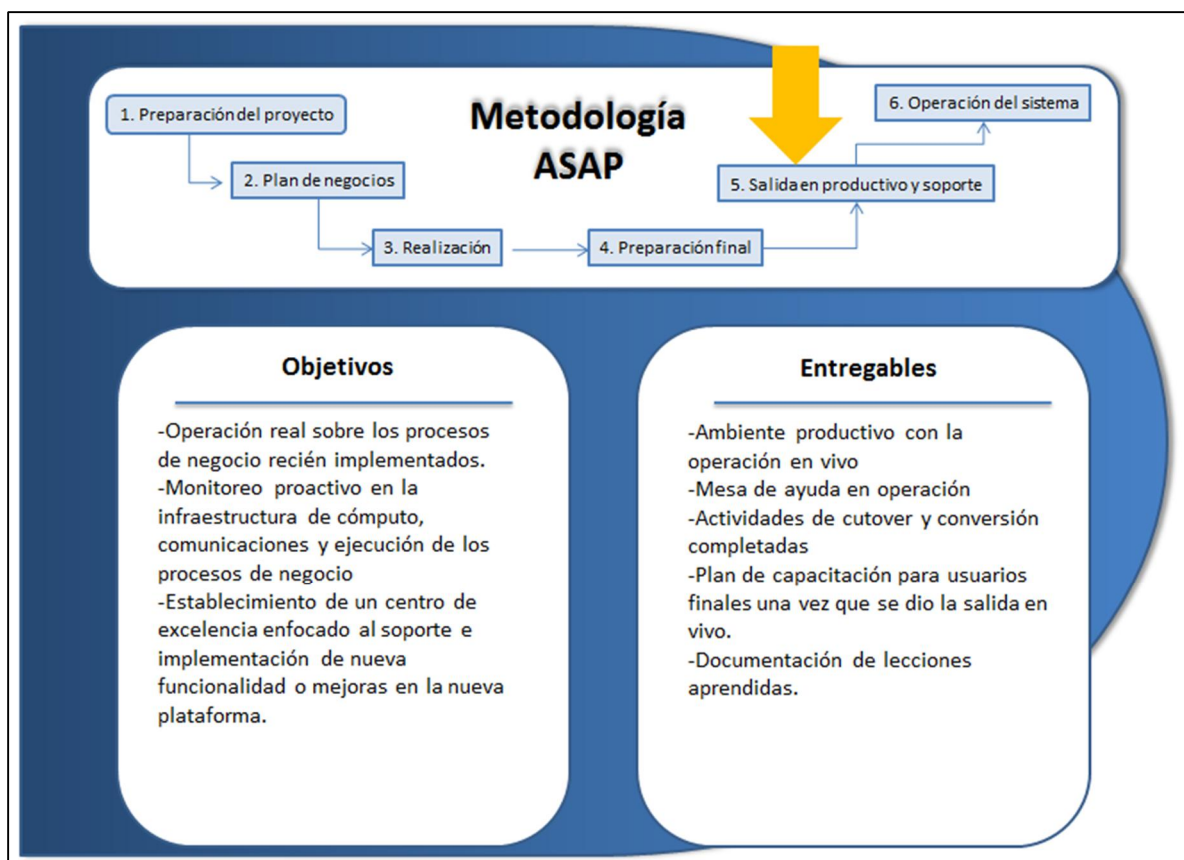


Fig. 2.1.5-1 Salida en productivo y soporte [SCAM2011].

De la misma manera, será necesario establecer un esquema de monitoreo proactivo, tanto a la infraestructura de cómputo utilizada, comunicaciones así como procesos de negocio asociados, con el objeto de minimizar los riesgos e identificar las áreas de oportunidad que necesiten ser atendidas, y minimizar con esto, posibles problemas que puedan impactar la operación del negocio.

### 2.1.6. Operar el sistema

La metodología ASAP establece una última fase enfocada a la operación del sistema, en la cual se busca que la operación del sistema se realice en base a los estándares planteados por SAP, en este caso como se observa en la figura 2.1.6-1, la recomendación es establecer un esquema de monitoreo a nivel técnico y de negocio en base a la herramienta “Solution Manager”, la cual se describirá con mayor detalle en los siguientes capítulos [SCAM2011].



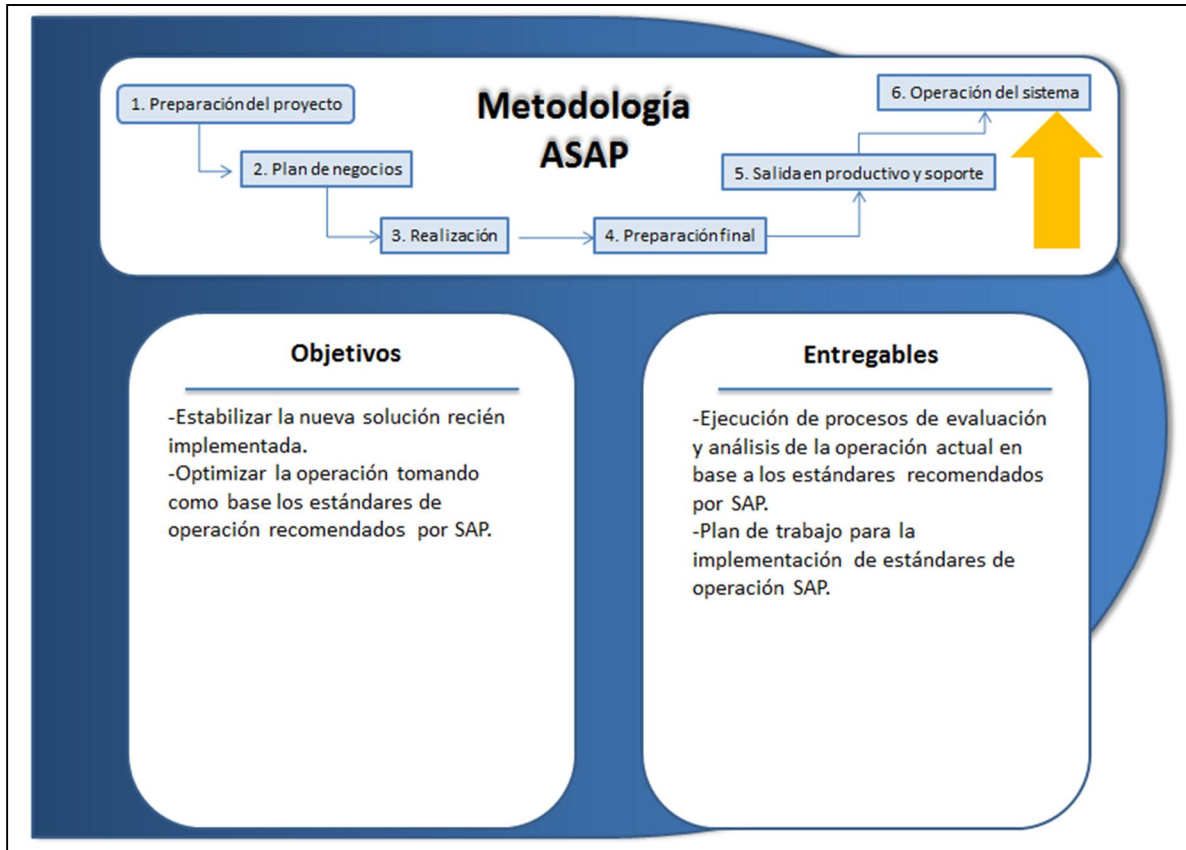


Fig. 2.1.6-1 Operación del sistema [SCAM2011].

## 2.2. Metodología RunSAP para la implementación de las operaciones

En la actualidad, los equipos de tecnología enfrentan una multitud de retos enfocados a garantizar la disponibilidad y la estabilidad de sistemas cada vez más complejos e integrados, buscando en todo momento, reaccionar de la manera más eficiente posible, ante los incidentes y los cambios que se presentan con la operación y el mantenimiento de los ambientes del día a día.

Uno de los objetivos principales de RunSAP como lo describe Gerhard Oswald en su obra “SAP Enterprise Support, ASAP to RunSAP”, es proporcionar las mejores herramientas y métodos necesarios para soportar la administración de las soluciones SAP, así como buscar la reducción del costo total de operaciones [GO2010].

La figura 2.2-1 muestra un ejemplo de cómo han ido creciendo con los años tanto en tamaño como en complejidad la infraestructura de operación de una solución que trabaja bajo una plataforma SAP.

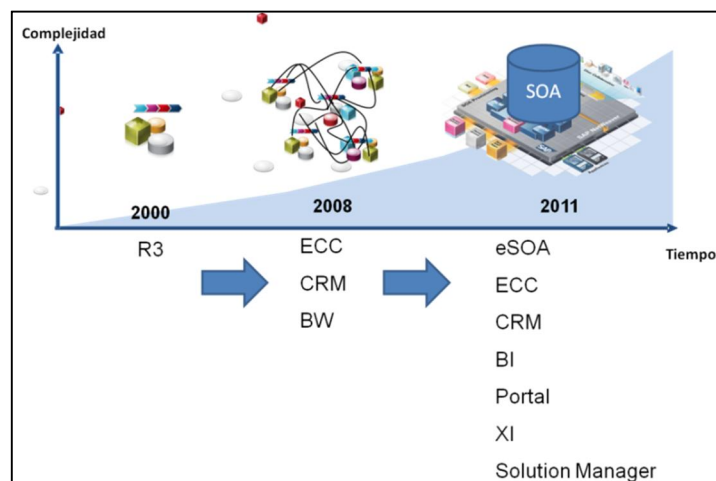


Fig. 2.2-1 Evolución de las soluciones de TI [GO2010].

La metodología RunSAP proporciona las mejores prácticas, procedimientos, y herramientas necesarias para la operación de las soluciones SAP, por medio de métodos de implementación estándar que han sido probados en numerosas instalaciones de ambientes ERP alrededor del mundo.

RunSAP se enfoca en la administración de soluciones SAP y la operación de procesos de negocio, establece los métodos de soporte necesarios para operar la infraestructura técnica necesaria y apoyar en el ciclo de vida completo de la solución, desde las fases de diseño hasta la operación y optimización del sistema, de la misma manera, establece recomendaciones como se observa en la figura 2.2-2 en las cuales se establece qué equipos de trabajo serán necesarios durante el proyecto de implementación, así como durante todo el ciclo de vida de la solución.

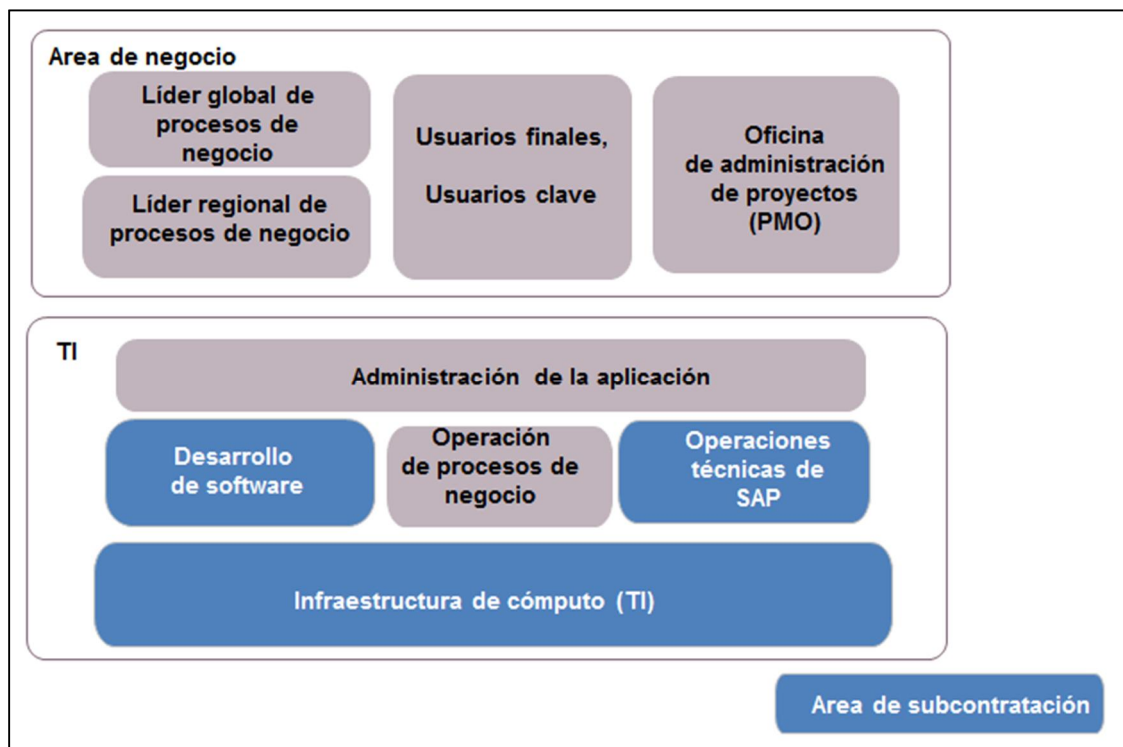


Fig. 2.2-2 Equipo de trabajo para la implementación [GO2010].

RunSAP toma como base la metodología ASAP y establece un modelo organizacional para los proyectos de implementación, en donde se divide claramente al equipo de negocio o funcional por un lado, y al equipo de tecnologías de la información por el otro.

En cada uno de estos equipos existen áreas específicas enfocadas a la operación y gestión de la solución, en este caso es necesario involucrar recursos directos por parte del cliente, pero también existe la posibilidad para algunas áreas de trabajar con proveedores de servicios externos en el equipo de tecnología, la recomendación inicial es evaluar la posibilidad de trabajar con un proveedor de subcontratación para las áreas de operaciones, desarrollo de software así como infraestructura técnica.

A continuación se describe con mayor detalle cada una de estas áreas, así como las funciones asignadas a cada una de ellas.

#### **Líder de procesos de negocio.**

Dado su profundo conocimiento en la operación del negocio, este recurso es el experto en el diseño y la administración de los procesos, requerimientos de implementación y mejora continua a los procesos existentes.

#### **Usuarios finales y usuarios clave.**

Son el primer punto de entrada para la identificación de problemas y retroalimentación al equipo de tecnología, la administración de incidentes se centra en este equipo.

#### **Oficina de administración del proyecto.**

Es el grupo central de administración en la unidad de negocio responsable de la planeación, implementación y mejora continua de la solución en todo el “landscape” y proyecto de implementación.

Todos los requerimientos de negocio deberán pasar por esta oficina, y deberán ser evaluados, autorizados e integrados en el plan de trabajo.

**Organización de administración de la aplicación.**

Este equipo es el área clave que interactúa como interface entre el equipo de negocios y el departamento de tecnologías de la información. Entre sus principales funciones se incluye el análisis de causa raíz para la atención de incidentes, la administración del control de cambios, entre otros.

**Operación de procesos de negocio.**

Es un área enfocada a los procesos de negocio críticos, su función es gestionar la generación y programación de nuevos procesos en el sistema, así como administrar, calendarizar y monitorear las herramientas encargas de realizar la ejecución de dichos procesos así como su dependencia con aplicaciones legadas o dependencias asociadas.

La figura 2.2-3 establece el modelo de operación para los equipos de trabajo, así como también algunas de sus principales funciones.

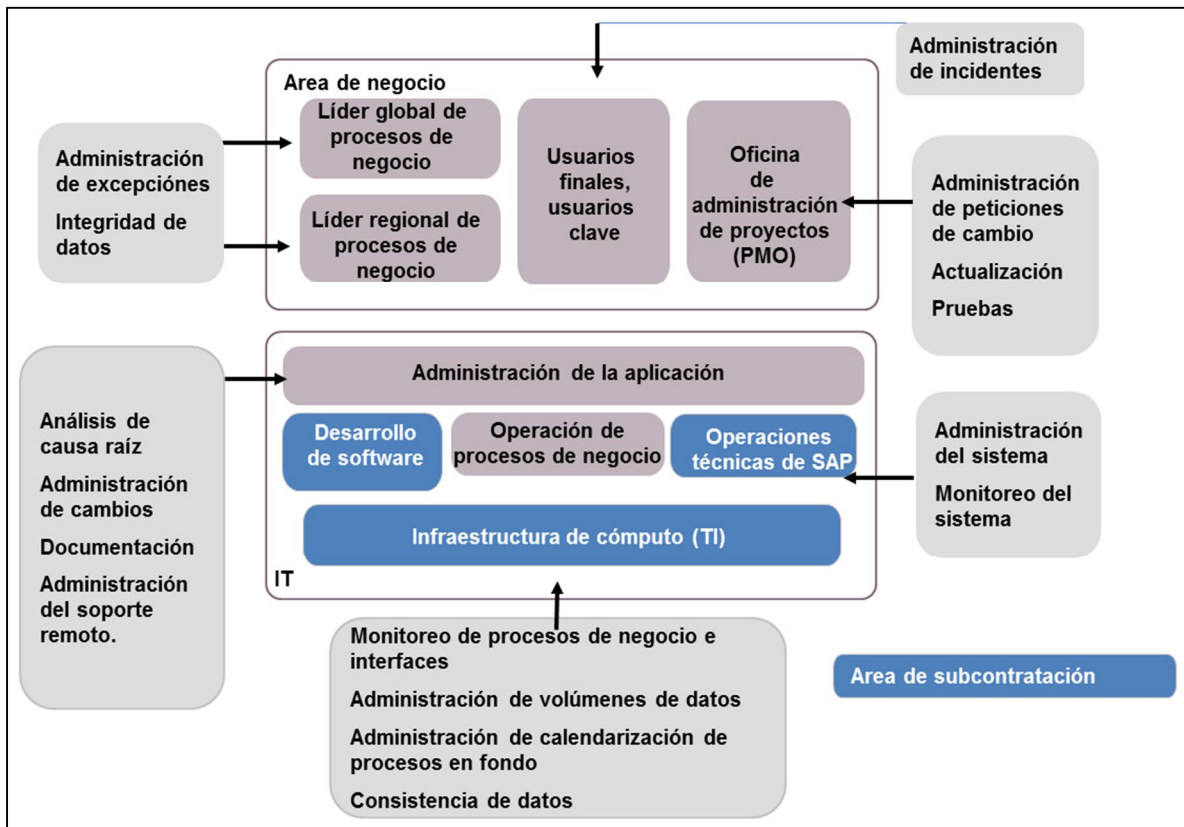


Fig. 2.2-3 Modelo de operación del proyecto [GO2010].

**Organización de operaciones SAP**

Se encarga de agrupar todas las actividades necesarias para administrar y monitorear el “landscape” de TI con el objetivo de mantener una operación exitosa de todos los componentes de la solución, el rol clave de esta área es la administración y monitoreo del sistema.

**Desarrollo de Software**

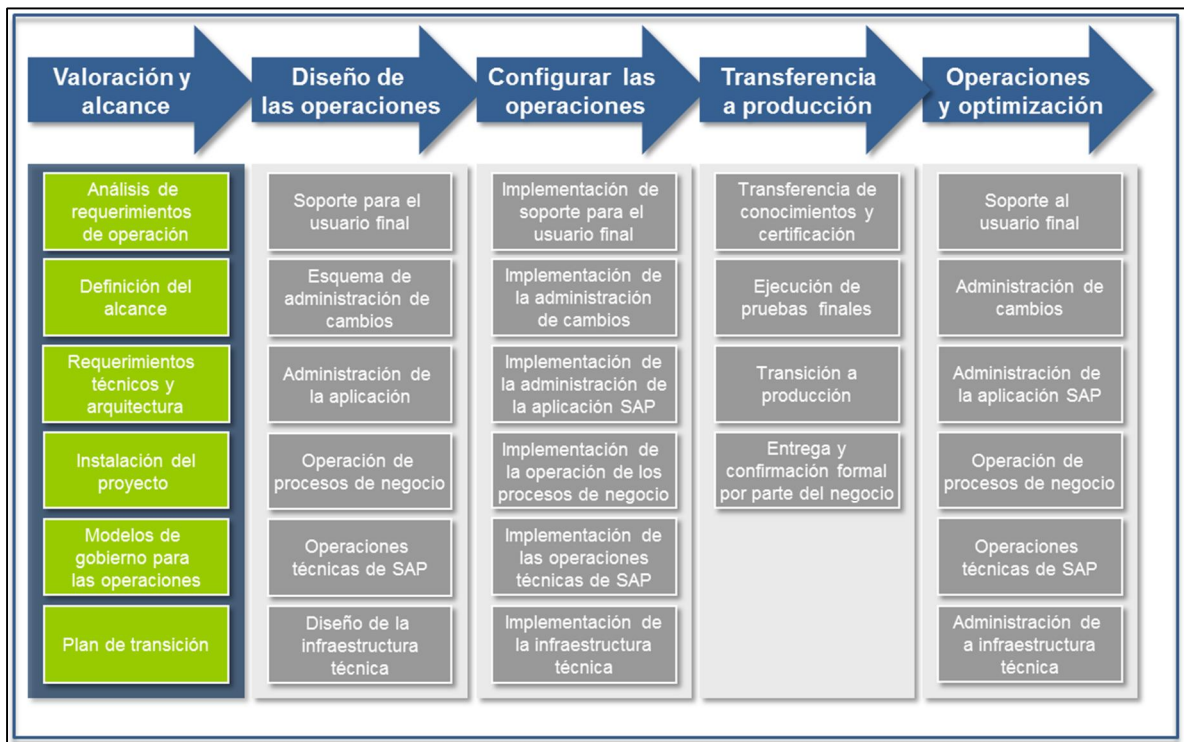
Su función incluye una variedad de tareas de desarrollo, incluyendo la programación de interfaces, la generación de reportes fuera del estándar, así como la creación de todo desarrollo enfocado a dar atención a algún requerimiento de negocio en particular.

## Infraestructura de TI

El equipo de infraestructura de TI, es el responsable de mantener la disponibilidad de toda la plataforma tecnológica en el nivel más técnico, en este equipo se ubican los especialistas encargados de realizar el mantenimiento y la administración de los servidores a nivel de Hardware y Software, particularmente la administración de Sistema Operativo, Base de Datos así como las todas las comunicaciones y enlaces de red asociados.

La metodología RunSAP está compuesta de las siguientes fases para realizar la implementación de una solución SAP, como se muestra en la figura 2.2-4 [SPMR2011]:

1. Valoración y alcance
2. Diseño de las operaciones
3. Configurar las operaciones
4. Transferencia a producción
5. Operaciones y optimización



**Fig. 2.2-4 Metodología RunSAP.**

En los capítulos siguientes se realizará una descripción detallada de cada una de estas fases, enfocándose en los puntos principales para llevar a cabo una implementación de SAP de manera exitosa.

### 2.2.1. Valoración y alcance

Esta fase representa la preparación para iniciar con el diseño de la implementación de las operaciones tomando en cuenta la nueva plataforma [SPMR2011].

Como se muestra en la figura 2.2.2-1, su principal objetivo es realizar una evaluación de los requerimientos técnicos y de negocio, que serán necesarios para realizar la implementación de los procesos de operación, ya sea que se traten de procesos nuevos, o por otra parte de procesos de negocio que desean ser ajustados o mejorados.

En la mayoría de los casos, las compañías cuentan ya con una estructura organizacional de soporte, enfocada a dar atención a las operaciones del negocio, pero al dar inicio con un proyecto de implementación ERP, es necesario realizar un ajuste e integración al modelo de operación actual, y dar cabida a la entrada de nuevos componentes.

En otras empresas es necesario iniciar con la definición de los procesos de soporte desde un inicio, ya que muchas veces, SAP llega a las empresas a sustituir la operación de otras soluciones que estaban enfocadas a la administración de ciertas áreas de la empresa.



Fig. 2.2.1-1 Etapa de valoración y alcance.

Como resultado de esta fase, se buscará conocer los tipos de procesos de soporte que serán implementados, así como los requerimientos de infraestructura técnica para llevarlos a cabo.

#### **Análisis de los requisitos operacionales.**

Los primeros pasos dentro de esta fase, están enfocados a entender los requerimientos de negocios, con el objetivo de garantizar el funcionamiento y operación estable de la solución, de tal manera que los procesos de negocio se ejecuten de la mejor forma y con el performance más adecuado, en este punto se identificarán las soluciones que requerirán un soporte de misión crítica con esquemas de soporte 7x24, lo cual requerirá mantener un monitoreo proactivo del sistema.

#### **Definición del alcance.**

Una vez que los requerimientos de operación se han identificado claramente, es necesario definir qué parte de las operaciones será necesario implementar o mejorar, con el objetivo de alcanzar los requerimientos de negocio.

Como parte de la definición de alcance, será necesario plasmar claramente los siguientes puntos:

- Las habilidades necesarias por parte del equipo responsable de administrar la solución.
- La estrategia de capacitación para el equipo que soportará la operación de la solución.
- Herramientas y procesos requeridos para la gestión de requerimientos del sistema.
- Las funciones necesarias que se implementarán en SAP Solution Manager como parte de la administración del "landscape".
- Será importante identificar de qué manera se asociarán los componentes de SAP e interfaces asociadas para dar soporte a los procesos de negocio que permitirán la operación de la empresa.

#### **Requerimientos técnicos y arquitectura.**

Para implementar las operaciones, se requerirá de una infraestructura técnica, esta infraestructura incluye las herramientas necesarias para dar soporte a la operación de los procesos tales como el monitoreo de la solución, el proceso de control de cambios, el proceso de atención y administración de incidentes generados por el usuario final, entre otros.

En esta fase será necesario identificar qué arquitectura técnica será utilizada para montar la plataforma SAP, tanto a nivel de Hardware como a nivel de Software, de la misma manera será necesario considerar un proyecto de dimensionamiento, en el que se determinará la capacidad y requerimientos de hardware necesario en los servidores, para operar la solución de una manera estable.

### Modelo de gobierno para las operaciones

Es necesario la definición de un modelo de gobierno, el cual se encargará de describir de forma clara cuál es la misión del equipo de Tecnologías de información, así como cuál es la organización, estándares y procesos sobre los cuales se regirá su operación.

Es importante la definición de este modelo para asegurar que las tecnologías de la información soporten y faciliten el desarrollo de los objetivos definidos por la empresa, aquí será necesario establecer el quién y el cómo se llevará a cabo la toma de decisiones así como los derechos y responsabilidades de todos los recursos del equipo.

### 2.2.2. Diseño de las operaciones

El objetivo de esta fase es realizar el diseño de las operaciones tomando como base los resultados obtenidos durante la fase de valoración y alcance [SPMR2011].

Es necesario realizar la creación de un plan de negocios en el cual se determinará los procesos de operación y se definirá que es necesario para mejorar o modificar los procesos de operación actual, para esta parte será necesario tomar en cuenta los siguientes puntos:

- ¿Cómo trabajarán los procesos sobre la nueva plataforma de ERP?
- ¿Qué herramientas son requeridas para implementar y soportar estos procesos?
- ¿De qué manera se realizará la capacitación al equipo de soporte para atender las nuevas tecnologías que serán implementadas?
- ¿Qué servicios de SAP serán utilizados?

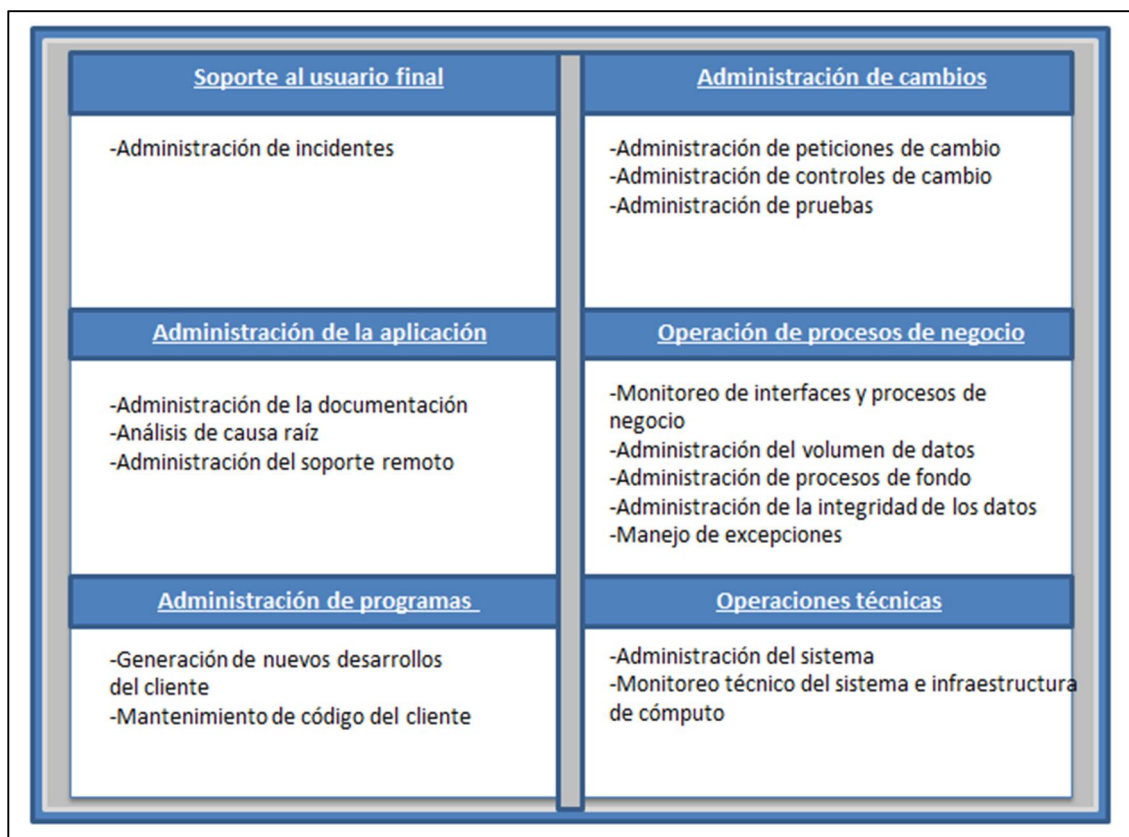


Fig. 2.2.2-1 Etapa de diseño de las operaciones [GO2010].

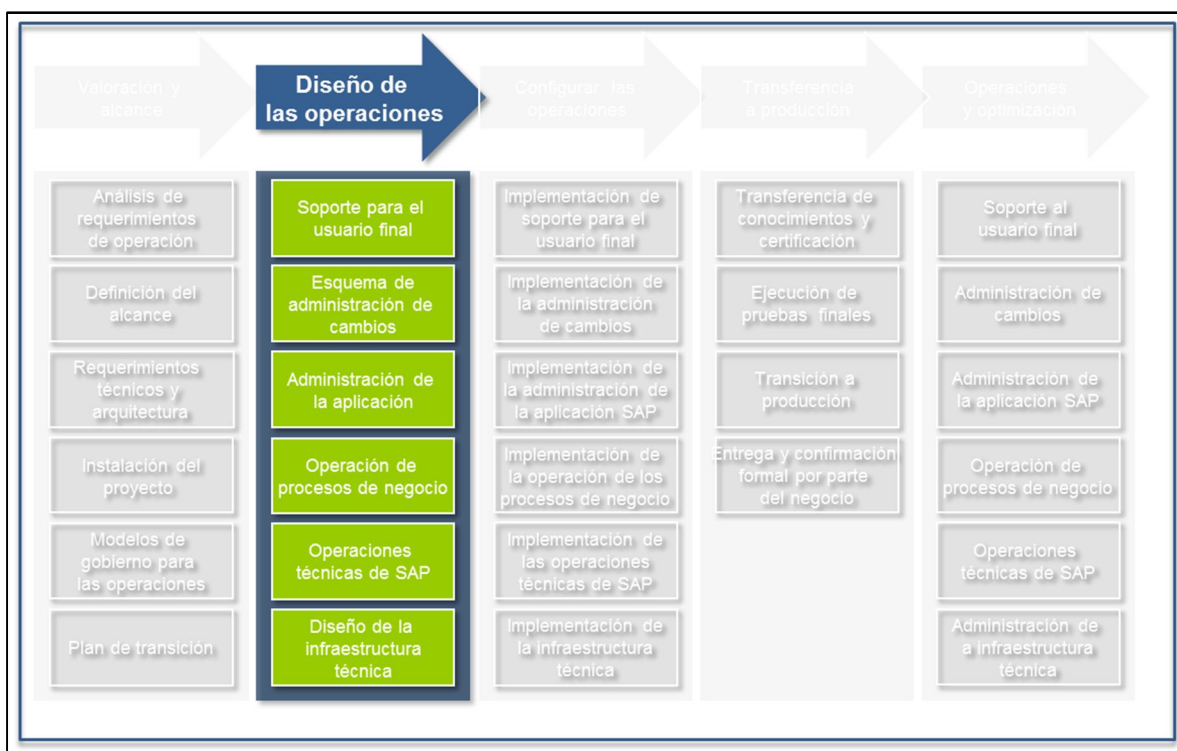
**Soporte para el usuario final.**

En la figura 2.2.2-1 podemos observar lo que se considera un esquema de soporte al usuario final, el soporte que se proporcione posterior a la salida en producción, es un punto clave para el éxito y para la aceptación de la nueva solución, así como también para mantener una operación estable de los procesos de negocio [SPMR2011].

Uno de los procesos más críticos de operación, es la atención y administración de incidentes hablando de sistemas de misión crítica como lo es SAP. La atención de incidentes se debe de manejar como un punto prioritario, ya que los problemas generados pueden ocasionar fuertes pérdidas económicas; así como el impacto directo a la operación del negocio si no se manejan y atienden de la manera adecuada, se identifican los problemas de raíz y se toman las acciones correctivas en tiempo y forma.

Generalmente las empresas cuentan con un proceso de atención de incidentes para los ambientes que tienen ya en operación. Al iniciar un proyecto de implementación ERP, la recomendación es alinearse al estándar de manejo de incidentes proporcionado por SAP, aquí se establecen las medidas necesarias a seguir, procesos, así como las herramientas recomendadas y equipos de colaboración involucrados, para dar atención en los tiempos más adecuados a todos los incidentes que se puedan presentar durante la operación diaria del sistema.

La figura 2.2.2-2 muestra algunos bloques principales que es necesario tomar en cuenta durante la fase del diseño de las operaciones.



**Fig. 2.2.2-2 Diseño de las operaciones.**

**Proceso de administración de cambios.**

El proceso de administración de cambios, establece los métodos, planeación y administración necesaria para atender los requerimientos de cambio que se vayan generando en el tiempo, buscando en todo momento minimizar los riesgos asociados a estos cambios de configuración.

El manejo adecuado de las peticiones de cambio, deberá garantizar la estabilidad de la solución. SAP recomienda como primera etapa nunca ejecutar un cambio en los ambientes productivos de manera directa [SPMR2011].

La ejecución de cualquier cambio, deberá iniciar desde un ambiente de Desarrollo en el cual, se validará el cambio de manera unitaria para analizar de qué manera impacta al sistema. Posteriormente, dicho cambio deberá ser validado en un ambiente de Calidad, en el cual se ejecutarán las pruebas correspondientes de manera integral, y si en este caso se determina que el cambio atiende el requerimiento solicitado sin impactar la solución, se deberá generar la solicitud y autorización correspondiente para transportarlo hacia el ambiente productivo. La figura 2.2.2-3 mostrada a continuación muestra un esquema de configuración del sistema de transportes para cumplir con esta tarea.

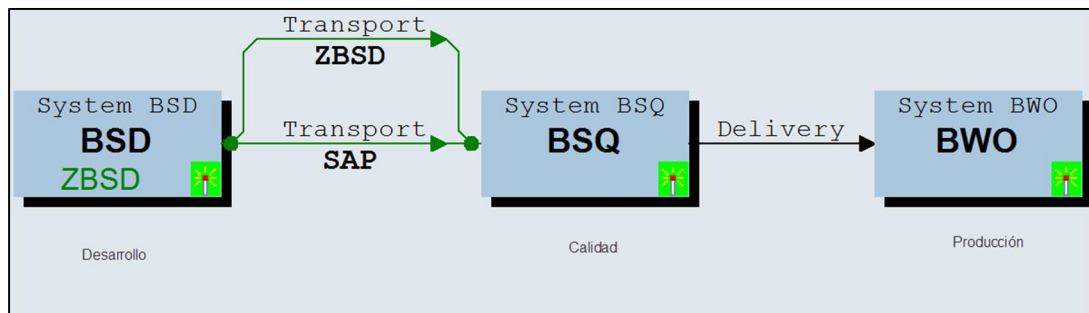


Fig. 2.2.2-3 Administración de cambios.

### Administración de la aplicación.

El objetivo de la administración de la aplicación, es mantener el sistema operando en las condiciones más óptimas, así como proporcionar el apoyo necesario para analizar y resolver los incidentes y problemas generados durante la operación del día a día.

La administración del sistema incluye una multitud de tareas tales como, la planeación del "landscape", la implementación de nueva funcionalidad, el manejo y ejecución de cambios provenientes, ya sea de las unidades de negocio, o del equipo de implementación como tal.

Este equipo se encargará de generar la documentación del sistema en base a estándares, así como asegurar y administrar el proceso de soporte remoto, para aquellos temas e incidentes que no sea posible resolver de manera local, y sea necesario involucrar al proveedor de software para generar un plan de acción para solucionarlos.

La documentación que se deberá generar incluirá un diagrama completo del "landscape" implementado. Se deberá mantener registrados los procesos de negocio que estarán corriendo en el sistema con todo el detalle posible, así como las características técnicas y de funcionalidad de las interfaces. Estos puntos serán un factor clave para la atención y resolución de incidentes y problemas, y será un mecanismo también para agilizar los análisis de casusa raíz que requieran atenderse para resolver los problemas de fondo.

### Operación de procesos de negocio.

Con el objetivo de asegurar la disponibilidad de los procesos de negocio, es necesario contemplar los siguientes puntos:

1. Administración de procesos de negocio, monitoreo y manejo de excepciones.

La administración de procesos de negocio deberá contemplar un monitoreo proactivo a cada uno de los procesos que se estarán ejecutando de manera concurrente en la solución SAP, será necesario definir la prioridad de cada uno, así como documentar claramente a los responsables, los horarios y días de ejecución, la duración estimada de cada uno de ellos así como definir un esquema de soporte para atender todos los problemas asociados a la cancelación o ejecución fallida de ellos, en donde es importante considerar el método de reacción asociado a una falla.

SAP ha desarrollado herramientas tales como Solution Manager, enfocadas a dar atención a cada una de estas tareas.



## 2. Administración de volumen de datos

La tendencia actual de mantener sistemas altamente interconectados, así como la necesidad por parte de las empresas, de mantener la información en línea y accesible en todo momento, ha traído consigo un incremento exponencial en los volúmenes de información que se almacenan día con día, por esta razón, es importante mantener una estrategia de monitoreo continuo sobre la disponibilidad de espacio en las unidades de almacenamiento, así como también definir estrategias para la depuración de información temporal y el archivado de datos que permita extraer la información de las BD, y almacenarla directamente en disco con el objetivo de no impactar en la medida de lo posible, el rendimiento y tiempo de respuesta de las Bases de Datos, como consecuencia del manejo de volúmenes de información muy elevados.

## 3. Administración de la calendarización de procesos.

La administración de la calendarización de procesos, establece de qué manera y en qué secuencia se deberán de ejecutar los procesos de negocio, ya sea en línea o en fondo, con el objetivo de optimizar los recursos de infraestructura existentes y evitar la presencia de cuellos de botella, que puedan impactar el rendimiento de los sistemas asociados a la saturación de recursos en horarios pico de operación [SPJS2011].

Para llevar a cabo esta tarea, será necesario mantener un monitoreo continuo de los recursos tanto de memoria, CPU así como las comunicaciones e interfaces asociadas.

La administración de la calendarización de procesos, incluye la definición de roles y responsabilidades, así como la definición de los procesos de planeación, control, calendarización y monitoreo de todos los procesos de negocio. La recomendación por parte de SAP, es realizar estas tareas de manera centralizada por medio de "Solution Manager" o por otra parte, haciendo uso de herramientas de terceros como lo son por ejemplo "CPS" de "Redwood" [SHSM2012].

### **Desarrollo de software personalizado por parte del cliente.**

Frecuentemente es necesario considerar el desarrollo de programas y reportes fuera del estándar de SAP, con el objetivo de complementar la funcionalidad, o dar atención a algún requerimiento de negocio particular que no contempla la solución.

Para este tipo de desarrollos, es importante establecer de qué manera se gestionará la generación de estos programas, como serán implementados estos objetos en el sistema, de qué forma se monitoreará su ejecución, así como cuál será el proceso para dar atención a las excepciones o problemas de funcionalidad reportados.

Todos estos procedimientos deberán ser generados como parte de la fase de diseño, de tal manera que puedan ser implementados durante la fase de operación productiva del ambiente.

### **Administración de la infraestructura de TI.**

La administración de la infraestructura de TI, involucra conceptos asociados a la gestión y operación de los siguientes recursos principalmente:

- Administración de servidores, mantenimiento y soporte tanto a nivel de Hardware como de Software.
- Administración de comunicaciones y enlaces de red.
- Gestión de la seguridad y controles, enfocado a la generación de accesos, así como a la identificación de vulnerabilidades y mitigación de riesgos.
- Administración de servicios de directorio
- Administración de redes de almacenamiento y archivado de datos
- Soporte a los usuarios vía mesas de ayuda o de servicios.

El equipo de infraestructura de TI tendrá como parte de sus principales objetivos, mantener la disponibilidad de los aplicativos y recursos técnicos sobre los cuáles se montará la plataforma SAP, con la finalidad de permitir una operación estable que facilite la ejecución del sistema de la forma más eficiente posible.

### 2.2.3. Configurar las operaciones

Una vez que el diseño se ha completado, será necesario iniciar ya con la implementación de los nuevos procesos y operaciones como se muestra en la siguiente figura, será necesario tomar como base la documentación generada hasta el momento, así como la información recabada en las fases anteriores [SPMR2011].

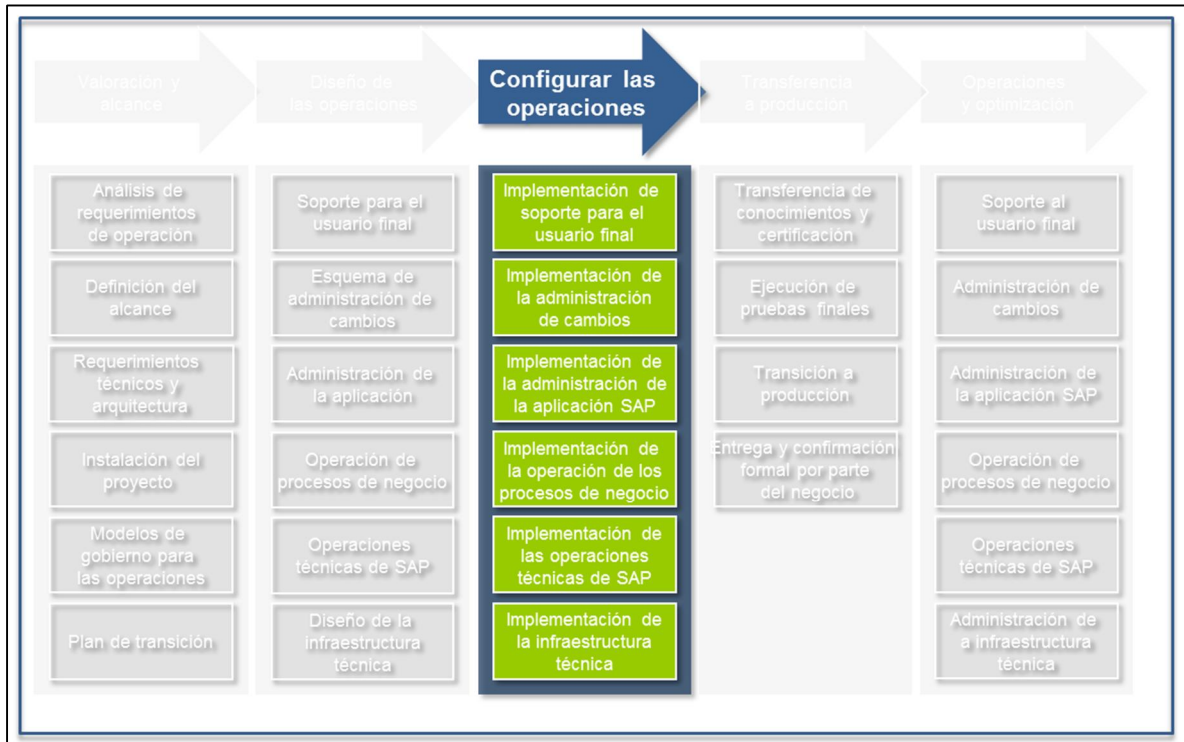


Fig. 2.2.3-1 Configuración de las operaciones.

Esta fase se enfocará a la configuración e implementación de todos los escenarios de negocio, en los cuales, se probarán los procesos de operación existentes así como todos los procesos de negocio que estarán soportados por SAP, básicamente se realizará la implementación de los siguientes puntos:

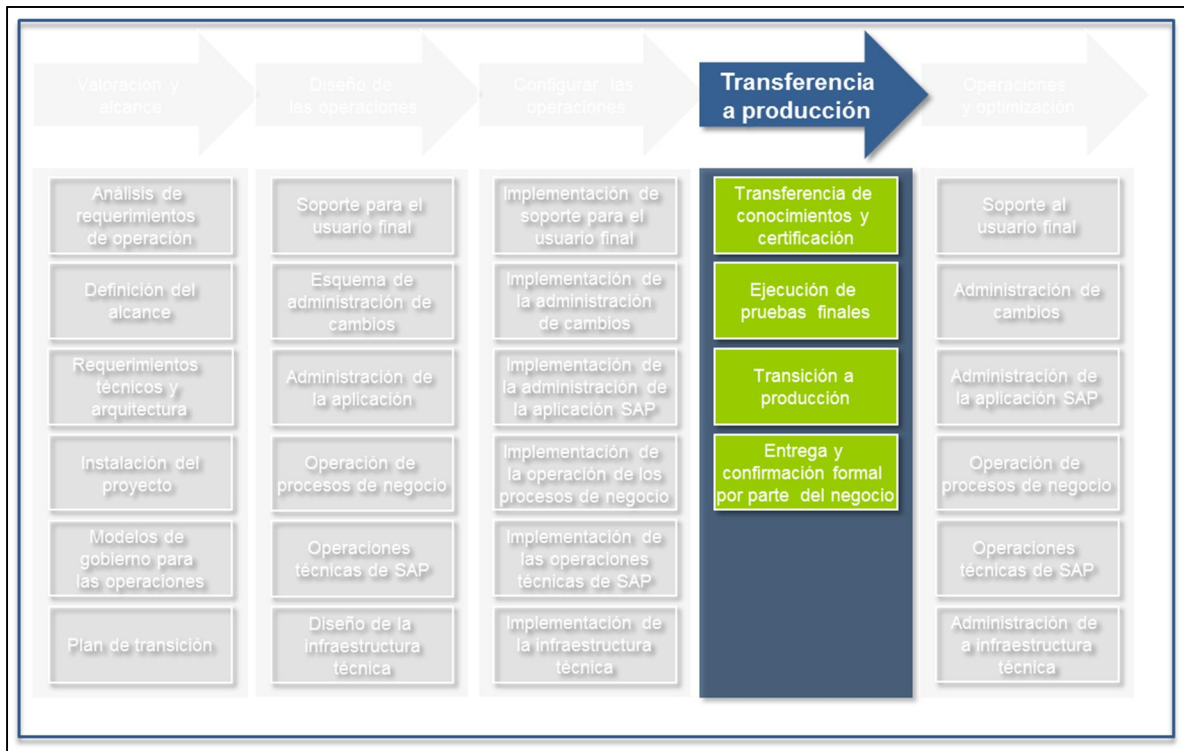
1. Esquema de soporte al usuario final.
2. Proceso de administración de cambios.
3. Proceso de administración de la aplicación.
4. Operación de procesos de negocio.
5. Desarrollo de aplicaciones personalizadas fuera del estándar de SAP
6. Infraestructura técnica que soportará la solución.

Para esta fase será necesario incluir un plan de trabajo de pruebas, el cual buscará validar la funcionalidad del sistema, así como para verificar los privilegios otorgados por medio de los roles y perfiles creados para los accesos asignados para todos los tipos de usuarios, dicha fase se conocerá como "Pruebas integrales", debido a que durante esta etapa, se probará la solución de forma integral con escenarios de negocio previamente diseñados, tomando como base la forma en la que operará el sistema una vez que esté en producción.

Como resultado de esta fase, se tendrá probada de manera integral toda la solución y deberá estar lista para la operación y salida en vivo.

### 2.2.4. Transferencia a producción

La fase de transferencia a producción y salida en vivo del sistema, se enfocará a trabajar en todas las actividades requeridas para iniciar la operación de la solución ERP, la siguiente figura muestra los principales puntos que se deben considerar durante la transferencia a producción [SPMR2011].



**Fig. 2.2.4-1 Transferencia a Producción.**

Entre los principales objetivos de esta etapa, se buscará asegurar que el equipo que soportará la solución se encuentra listo y debidamente capacitado para dar atención a todos los requerimientos, incidentes y cambios generados con la nueva plataforma que tomará la operación del negocio.

De la misma manera, se realizarán las pruebas finales para garantizar que todas las herramientas y procesos de negocio implementados, se encuentran listos para asegurar una salida en vivo exitosa, y también se confirmará que la plataforma cumple con los acuerdos de servicios generados respecto al performance de los procesos, tiempo de respuesta del sistema e interacción con las interfaces asociadas.

Finalmente la empresa consultora realizará la entrega formal de la solución al cliente, y el negocio por su parte, firmará la toma del sistema y fecha acordada para la salida en vivo de SAP.

### 2.2.5. Operaciones y optimización

El objetivo primordial de la fase de operaciones y optimización, será verificar que la solución está operando de manera estable después de la salida en vivo, cumpliendo de manera adecuada los requerimientos de negocio [SPMR2011].

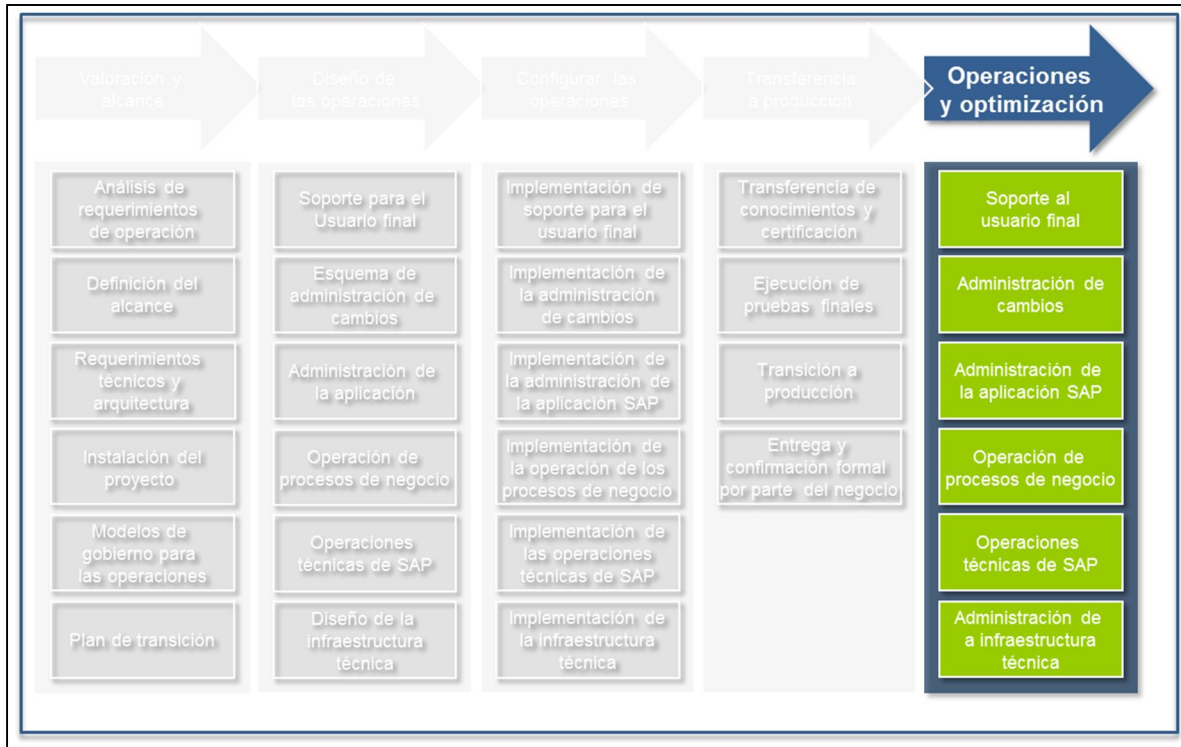


Fig. 2.2.5-1 Operaciones y optimización.

Para alcanzar esta meta, será necesario establecer un esquema de monitoreo proactivo sobre todos los procesos y herramientas asociadas como se describe en la figura 2.2.5-1, con el objetivo de estabilizar el sistema durante los primeros días, los cuáles suelen ser un punto clave para la aceptación del sistema por parte de los usuarios finales.

En esta etapa se buscará también analizar el comportamiento del sistema y encontrar las áreas de oportunidad que sean factibles de optimizar, ya sea en el aspecto técnico hablando del performance y tiempos de respuesta del sistema, cuestiones referentes a la seguridad y a la posible asignación de privilegios adicionales e innecesarios, así como mejoras a los procesos de negocio como tal, que permitan en términos generales, evitar en la medida de lo posible problemas de indisponibilidad que impacten la operación del negocio, así como también buscar la reducción de los costos de mantenimiento de la solución.

### 2.3. SAP Solution Manager y las normas ITIL

En la actualidad las grandes empresas se ven obligadas a enfrentar una serie de retos, enfocados a mantener operando de la manera más óptima su infraestructura de sistemas.

La integración a la que se ha llegado hoy en día con las nuevas tecnologías, implica en muchas ocasiones la ejecución de procesos de negocio que involucran operaciones globales, el hecho de mantener una misma solución a la que acceden usuarios de diferentes países y en diferentes idiomas, así como con aplicaciones de diversos proveedores incluyendo los desarrollos en casa, es un gran reto que las organizaciones enfrentan día con día.

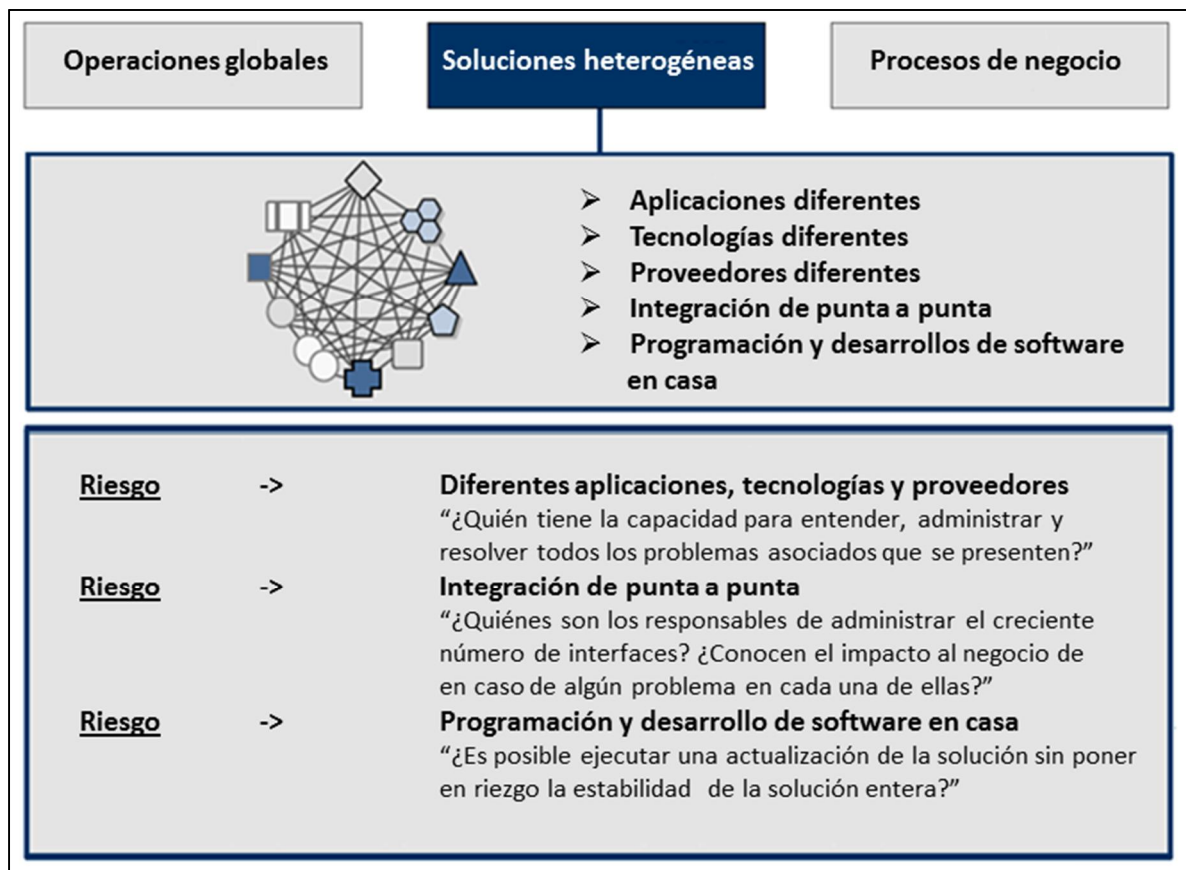


Fig. 2.3-1 Integración de procesos de negocio [SHSM2012].

Como consecuencia de esta gran integración, el número de interfaces entre los sistemas ha crecido de una manera considerable, trayendo con esto “landscape” cada vez más complejos, así como la necesidad de contar con un equipo de trabajo cada vez más especializado, que deberá ser capaz de administrar y dar soporte a esta plataforma.

Para llevar a cabo esta tarea, será necesario que el equipo mantenga una documentación clara referente al flujo y atención de todos los procesos de negocio de la empresa, que estarán corriendo desde SAP.

Entre los puntos principales que deberán ser incluidos en la estrategia de documentación, se encuentran los siguientes:

- Roles y responsabilidades de todo el equipo encargado de administrar la solución.
- Procedimientos de monitoreo para todos los procesos, manejo de excepciones así como la creación de una matriz de escalación asociada.
- Procedimientos de programación de procesos, incluyendo la planeación así como el control y dependencias asociadas a estos procesos.

La figura 2.3-2 muestra un ejemplo claro en donde podemos identificar un alto número de instancias involucradas en la implementación de un proceso particular de negocio, así como el riesgo asociado.

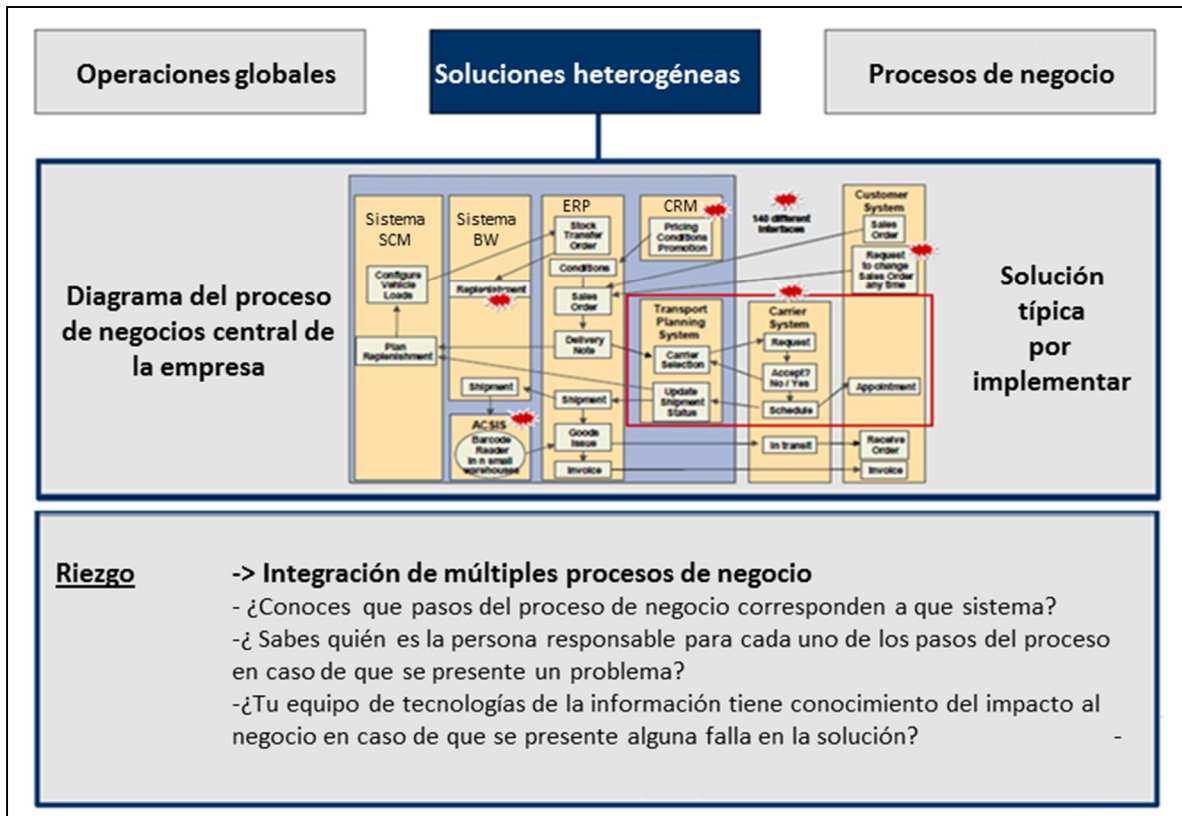


Fig. 2.3-2 Implementación de procesos de negocios [SHSM2012].

SAP desarrolló la herramienta Solution Manager con el objetivo de administrar la solución de manera completa abarcando todo el ciclo de vida del sistema, desde las fases de valoración y diseño, hasta la salida en producción y operación del sistema, tomando como base los mecanismos establecidos en las normas de ITIL como principio fundamental [SCSM2011].

Como se describe en la figura 2.3-3, ITIL es un marco de referencia utilizado a nivel mundial, el cual cuenta con una serie de librerías que establecen las mejores prácticas que se han venido desarrollando y documentando a lo largo de estos años, para llevar y administrar la entrega de servicio de Tecnologías de la Información en una empresa de la manera más óptima y eficiente.

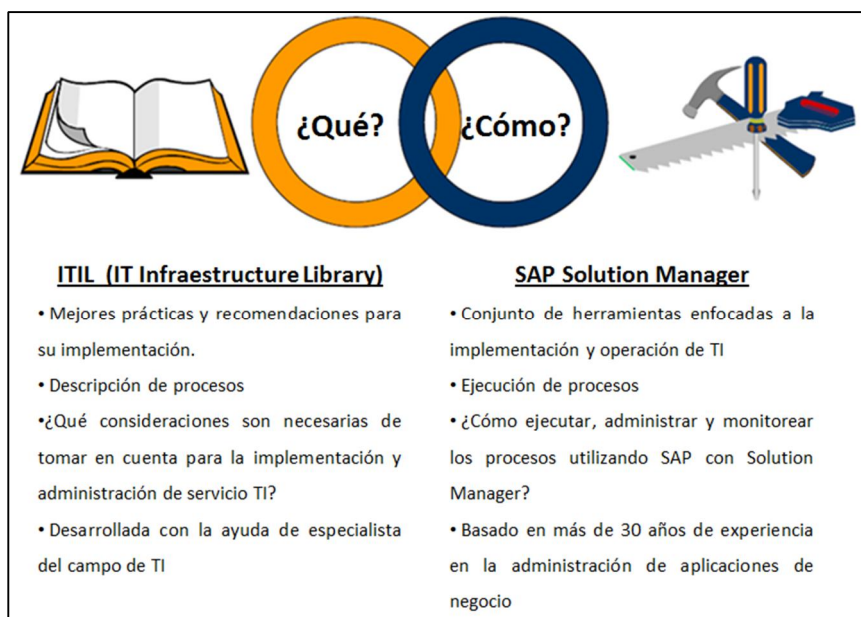


Fig. 2.3-3 ITIL [SCSM2011].

ITIL establece cuales son los procesos que deben de implementarse para realizar la entrega de servicio de TI, por su parte Solution Manager plantea de qué manera se deberán implementar y administrar estos procesos, enfocados tanto a la parte de infraestructura así como al desarrollo y operación de TI [SCSM2011].

ITIL tiene sus orígenes desde principios de los años 80 en el CCTA (Central Computer and Telecommunications Agency, por sus siglas en inglés) con sede en Inglaterra, pero se empezó a utilizar desde principios de los años 90, tomando como base para estas recomendaciones algunos estándares de TI, así como marcos de trabajo como son COBIT, la ISPL (Information Services Procurement Library, por sus siglas en inglés) y el DSDM (Dynamic System Development Method, por sus siglas en inglés), entre otros [WIIT2012].

ITIL versión 3 es la última versión publicada a finales del 2006 por el Reino Unido, e incluye principalmente los siguientes libros:

1. Diseño de servicios de TI
2. Introducción a los servicios de TI
3. Operación de los servicios de TI
4. Mejora de los servicios de TI
5. Estrategia de los servicios de TI

Solution Manager ofrece una serie de herramientas enfocadas a dar atención a cada uno de los servicios establecidos por ITIL, entre ellos podemos encontrar como se describe en la figura 2.3-4, herramientas de monitoreo enfocadas al proceso de disponibilidad, la generación de reportes de EarlyWatch Alerts, utilizados para mantener un monitoreo continuo de la capacidad, así como de la sanidad de los sistemas, el módulo de "Service Desk" o servicio de escritorio cuya principal finalidad es dar atención a los procesos de cambios, atención de incidentes, problemas, entre otros [SCSM2011].

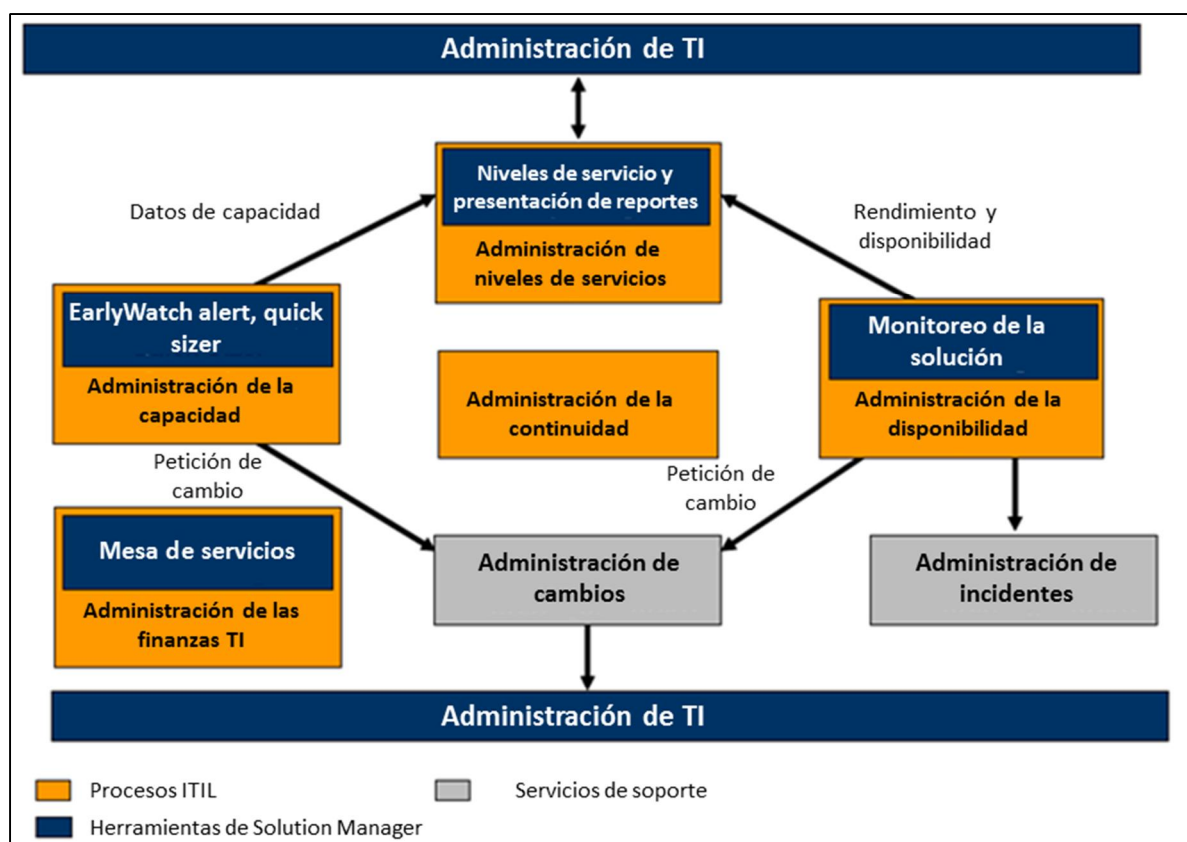


Fig. 2.3-4 Herramientas de ITIL [SCSM2011].

Solution Manager cuenta con un importante número de módulos que permiten a las empresas administrar el proyecto de implementación así como la solución completa, la figura 2.3-5 describe algunos de estos.

Por otra parte, esta herramienta ofrece ventajas tales como, la posibilidad de realizar una administración eficiente de toda la solución de manera centralizada, proporcionando además la facilidad de realizar análisis de rendimiento en los sistemas, proporcionar soporte a SAP de manera remota, así como también la oportunidad de realizar análisis de causa raíz para todos los problemas que se van presentando con el día a día de la operación, por medio del módulo de análisis de causa raíz.

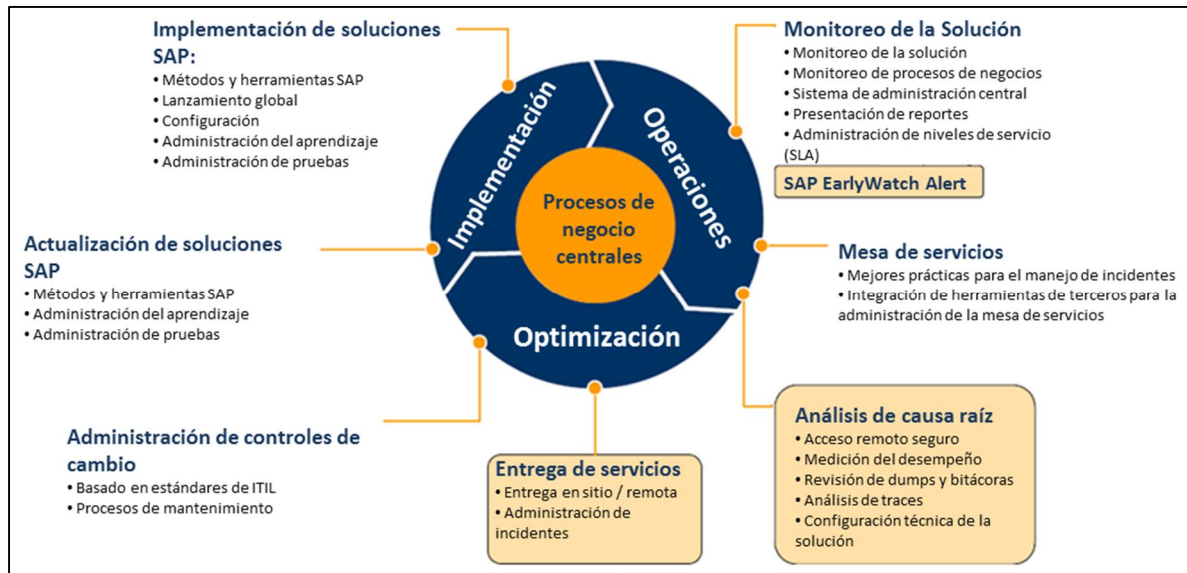


Fig. 2.3-5 Módulos de ITIL [SCSM2011].

## 2.4. Listo para SOA

SOA es una metodología de modelado y diseño de sistemas que establece la funcionalidad de los aplicativos en base a servicios bien definidos, que buscan dar atención a los requerimientos del negocio. Generalmente estos servicios se implementan en ambientes de Internet o Intranet utilizando servicios web, en los cuáles los nodos de la red ponen a su disposición sus recursos, con el objetivo de permitir la comunicación con otros nodos en la red [SCSO2011].

Esta comunicación se establece por medio de lenguajes y protocolos estándar de desarrollo como son WSDL (Web Services Description Language) y SOAP (Simple Object Access Protocol). Cada servicio diseñado, está enfocado en atender una acción en particular, tal como realizar una compra de un boleto de autobús, solicitar una consulta a un banco, o cualquier requerimiento de negocio que sea solicitado.

De esta manera, SOA le permite a las empresas ofrecer a los usuarios el acceso a la información de una manera más simple y oportuna como se muestra en la figura 2.4-1, así como el hecho de contar con una plataforma de desarrollo abierta basada en estándares, que les permitirán atender los requerimientos de negocio de una manera más ágil y a la velocidad que el ambiente actual demanda el acceso a la información.

SAP utiliza la plataforma SOA para la atención de los requerimientos de negocio, conceptualmente esta arquitectura se puede representar en el modelo cliente-servidor, en donde por un lado se cuenta con un servicio proveedor de cierto proceso de negocio, y por el otro se cuenta con una unidad consumidora de dicho servicio.

La plataforma desarrollada por SAP ofrece toda una gama de servicios empresariales por medio del ESR (Enterprise Service Repository), en la cual se cuenta ya con la definición de interfaces, servicios y objetos de negocio estándar para soportar la implementación de la nueva solución.



### 3. Estructura técnica del sistema SAP ERP

Este capítulo da a conocer la arquitectura de los sistemas SAP. Basada en la configuración cliente/servidor, se presentan los procesos que conforman un servidor de aplicaciones web SAP (SAP Web AS, por sus siglas en inglés) y así mismo se define el término instancia. En este capítulo se muestran también las diferentes opciones de instalación de un sistema SAP ERP.

#### 3.1. Tecnología cliente/servidor

La arquitectura cliente/servidor consiste básicamente en un proceso (cliente) que realiza peticiones a otro proceso o grupo de procesos (servidor) que se encarga de dar respuesta a las peticiones del primero [WICS2011], como se muestra en la figura 3.1-1.

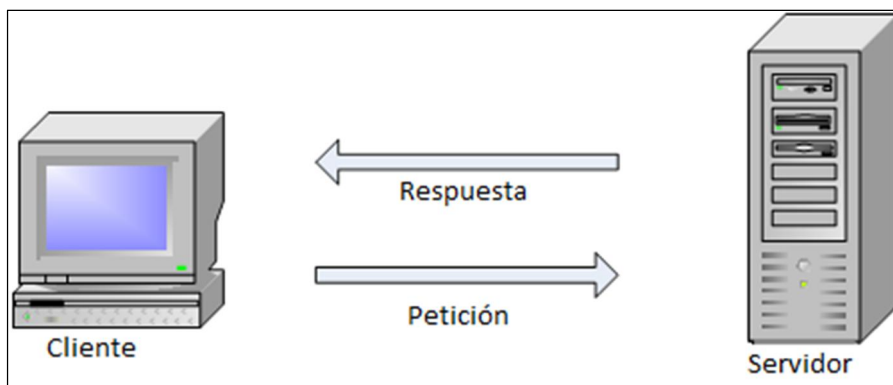


Fig. 3.1-1 Arquitectura cliente/servidor

#### 3.2. Arquitectura del software de SAP

Los siguientes procesos son necesarios para la operación de un software de aplicaciones de negocios:

- Procesos de presentación.
- Procesos de aplicación.
- Procesos de la base de datos.

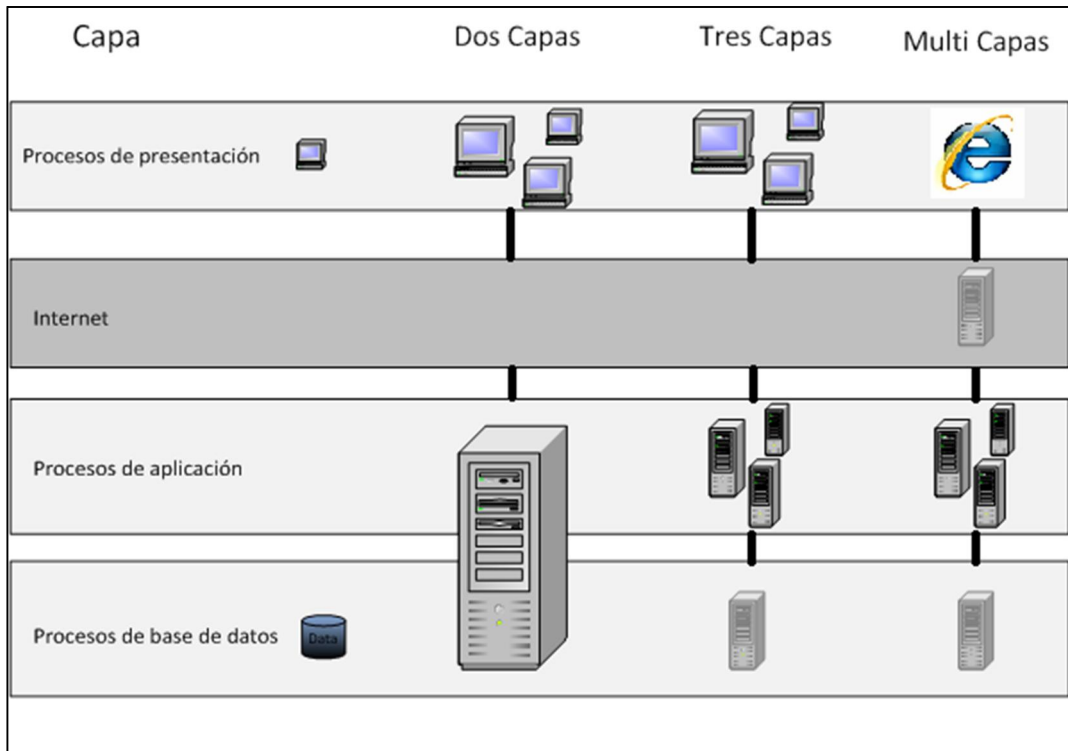
Durante la configuración de un sistema SAP, es necesario decidir cómo se van a distribuir los procesos entre el hardware que hay disponible.

En los años que hemos trabajado como consultores de tecnología en SAP, hemos conocido las configuraciones que se detallan más abajo.

En una configuración de una capa, todas las tareas (base de datos, aplicación y presentación) son procesadas por una sola computadora. Este tipo de configuración es utilizado principalmente con fines de prueba o demostración.

Una configuración de dos capas, generalmente, separa los procesos de presentación, de aquellos de aplicación y base de datos. Una configuración de este tipo provee un buen desempeño para un número grande usuarios sin incrementar sustancialmente los costos de hardware.

Para lograr una configuración de tres capas, cada una de las tareas correrá en diferentes equipos. La figura 3.2-1 ejemplifica las configuraciones que acabamos de mencionar.



**Fig. 3.2-1 Configuraciones cliente/servidor**

Una ventaja importante de la configuración multicapas, es el grado de distribución que pueden tener los procesos del lado del servidor. Esta distribución puede contribuir mucho al mejoramiento del rendimiento de la aplicación. Por ejemplo, los procesos de aplicación pueden distribuirse en varias computadoras.

Durante los años de ejercicio, hemos notado que resulta muy conveniente separar los procesos de la base de datos de los procesos de aplicación, es decir, dedicar un servidor para la base de datos y ejecutar los procesos de aplicación en otro servidor, de esa manera los procesos de la base de datos pueden aprovechar al máximo todos los recursos del servidor, en lugar de compartirlos con los procesos de la aplicación. Otra ventaja, es que, al contar con una capa específica para los procesos de aplicación la escalabilidad del sistema se facilita, porque sólo se agregan los servidores de aplicación necesarios, sin que esto signifique un cambio para el servidor de la base de datos.

### 3.3. Tecnología ABAP

ABAP (Advanced Business Application Programming – Programación Avanzada de Aplicaciones de Negocio) es un lenguaje de programación desarrollado por SAP. Muchas de las aplicaciones de un sistema SAP están desarrolladas en lenguaje ABAP. El lenguaje de programación ABAP ha sido optimizado para el desarrollo de aplicaciones de negocio escalables. Los consumidores pueden utilizar el banco de trabajo ABAP (“ABAP Workbench”) para desarrollar aplicaciones completamente nuevas, así como para mejorar o modificar aplicaciones estándar de SAP. El servidor de aplicaciones web de SAP provee la plataforma de ejecución para los programas escritos en ABAP.

De acuerdo al modelo de programación ABAP, arquitecturas multicapas pueden ser implementadas para aplicaciones empresariales, mientras la lógica del programa es ejecutada en el servidor de aplicación ABAP. La figura 3.3-1 muestra un diagrama de cómo son procesadas las aplicaciones ABAP [TADM10\_12005].

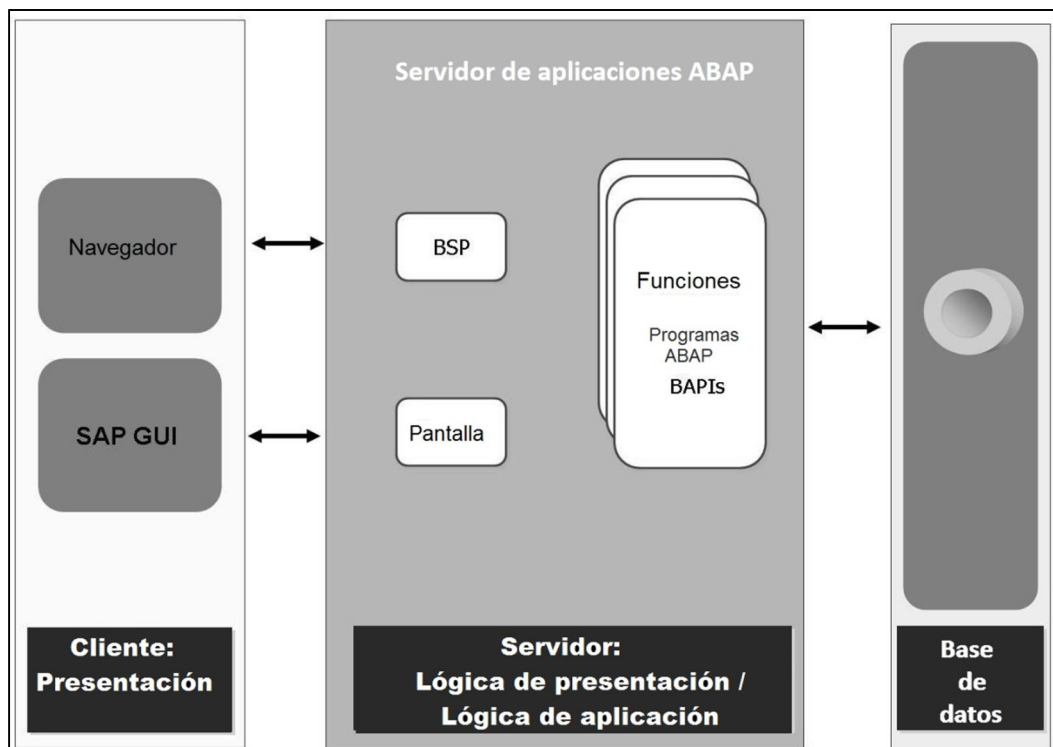


Fig. 3.3-1 Arquitectura de aplicaciones ABAP [TADM10\_12005]

El servidor de aplicación ABAP procesa las peticiones de los clientes, ejecuta la lógica de aplicación y regresan los resultados a los clientes. La información es accedida a nivel base de datos. La lógica de aplicación es, por ejemplo, implementada en forma de programas ABAP o de módulos de funciones, en el lado del servidor la lógica de presentación consiste de pantallas o “dynpros” (dynamic programs – programas dinámicos) o de páginas de servidor de negocios (BSP por sus siglas en inglés).

### 3.3.1. Procesos de un servidor de aplicaciones SAP ABAP

En un entorno de ejecución ABAP existen los siguientes procesos:

- El despachador ABAP: Se encarga de distribuir las peticiones a los procesos de trabajo ABAP.
- Los procesos de trabajo de diálogo satisfacen todas las peticiones que corren en primer plano solicitadas por un usuario. Cada despachador requiere al menos dos procesos de trabajo de diálogo.
- Los procesos de trabajo “spool” o impresión se encargan de pasar el flujo de datos secuencial a las impresoras. Cada sistema SAP necesita al menos uno.
- Los procesos de trabajo de actualización ejecutan las peticiones de actualización de datos en las tablas. Así como los procesos de trabajo impresión, se requiere al menos uno por cada sistema SAP.
- Los procesos de trabajo de segundo plano tienen la tarea de ejecutar programas que no requieren la interacción del usuario. Es necesario contar con dos procesos de trabajo de este tipo en cada sistema SAP.
- El proceso de trabajo de “enqueue” (bloqueo o encolamiento) administra la tabla de bloqueos en la memoria compartida. La tabla de bloqueos contiene los bloqueos lógicos que ocurren en la base de datos de un sistema SAP. Sólo se requiere de un proceso de trabajo de este tipo.

Además de los procesos de trabajo antes mencionados, el servidor de aplicaciones ABAP cuenta con los siguientes servicios:

- El servidor de mensajes, que se encarga de manejar la comunicación entre los diferentes despachadores del sistema. Este servicio sólo se configura una vez por cada sistema.
- El servidor “gateway” habilita la comunicación entre sistemas SAP, o entre sistemas SAP y sistemas de aplicaciones externos. Existe uno por cada despachador.
- El gestor de comunicación por internet permite a los sistemas SAP comunicarse directamente con internet. Este servicio recibe todas aquellas peticiones que llegan a través de internet y

las distribuye al despachador ABAP o al despachador Java, dependiendo del tipo de petición [TADM10\_12005].

### 3.4. Tecnología Java

El lenguaje de programación Java fue introducido por primera vez en 1995 por Sun Microsystems Inc. Java es un lenguaje de programación orientado a objetos e independiente de plataforma, es decir, un mismo programa puede correr en un sistema AIX, HP-UX o Windows.

La introducción del concepto de Java en sistemas SAP, permite el desarrollo de un largo rango de diferentes tipos de aplicaciones, desde aplicaciones clásicas hasta “applets” utilizados en sitios web para aplicaciones cliente/servidor.

De acuerdo a las especificaciones J2EE, las arquitecturas cliente/servidor de tres capas pueden ser implementadas, el servidor de aplicaciones J2EE se encarga de hacerlo funcionar. La figura 3.4-1 ilustra la arquitectura de las aplicaciones Java en un Servidor de Aplicaciones.

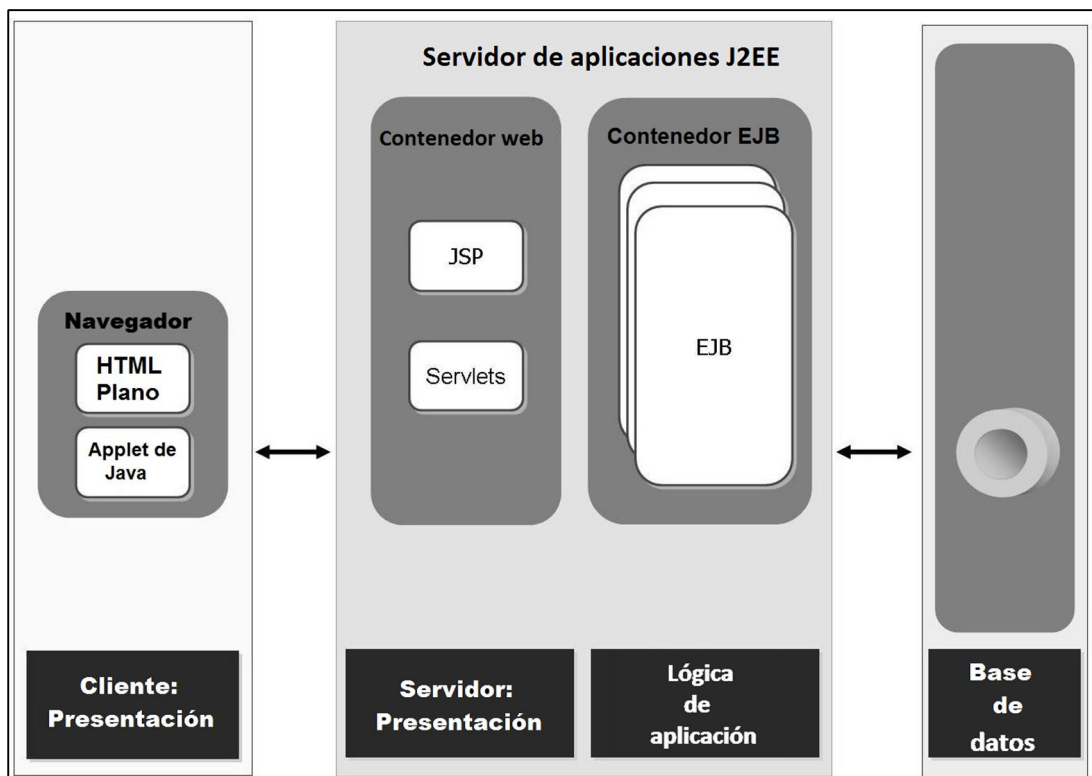


Fig. 3.4-1 Arquitectura de aplicaciones Java [TADM10\_22005]

El servidor de aplicaciones J2EE procesa las peticiones de los clientes, ejecuta la lógica de la aplicación y regresan los resultados a los clientes. La lógica de aplicación está implementada en forma de EJB (Enterprise Java Beans, por sus siglas en inglés), los cuales son ejecutados en un contenedor EJB.

La comunicación entre el cliente y el servidor J2EE está basada en estándares web, tales como, HTTP, HTML y XML. El servidor J2EE utiliza JSP (Java Server Pages) y Java “servlets” para generar páginas HTML e información en código XML. Los JSP y los Java servlets son ejecutados en un contenedor web, que forma parte del servidor de aplicaciones Java. Dentro de las páginas HTML que son enviadas al navegador de red, se pueden integrar java “applets”, éstos son ejecutados por el navegador de internet.

#### 3.4.1. Procesos de un servidor de aplicaciones Java

Los siguientes procesos conforman el ambiente Java:

- El despachador Java distribuye las peticiones Java a los procesos de servidor Java.
- Los procesos de servidor Java son multi-hilo y se encargan de ejecutar las aplicaciones Java. A diferencia de los procesos ABAP, pueden atender varias peticiones al mismo tiempo.

- El servicio de mensajes Java es el encargado de controlar toda la comunicación en el ambiente Java. Cuenta con una lista de todos los despachadores y procesos de servidor Java.
- El servicio de "enqueue" o servicio de bloqueos de Java, controla los bloqueos lógicos utilizados por la aplicación que corre en dado momento en los procesos de servidor.
- El conector SAP Java es responsable de la comunicación entre el ambiente ABAP y el ambiente Java [TADM10\_22005].

### 3.5. El término instancia en sistemas SAP

Una instancia es una unidad administrativa que combina varios componentes del sistema SAP para proveer uno o más servicios. Los servicios que provee una instancia son iniciados o detenidos al mismo tiempo. Para controlar todos los componentes de una instancia, se ajustan los parámetros contenidos en el perfil de la instancia.

La capa de aplicación de un sistema SAP generalmente consiste de varias instancias. Sin embargo, si la capa de aplicación está constituida por una sola instancia, entonces, todos los procesos necesarios para operar un sistema SAP deben ser configurados en esta instancia. Una instancia que cuenta con todos los elementos mínimos necesarios se le conoce como **instancia central**. En la figura 3.5-1 se representan diferentes tipos de instancia.

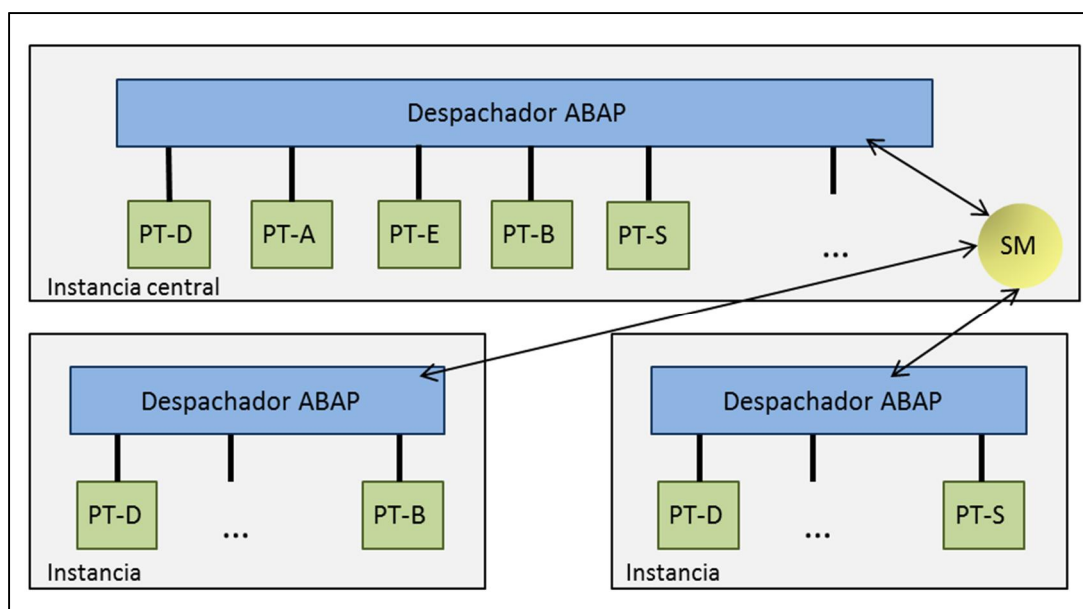


Fig. 3.5-1 La instancia

Los servicios que un servidor de aplicaciones puede proveer están determinados por los procesos de trabajo con los que cuenta. Así, por ejemplo, una instancia puede tomar varios roles, tales como: servidor de diálogo y simultáneamente servidor de procesos de segundo plano si cuenta tanto con procesos de trabajo de diálogo y procesos de trabajo de segundo plano. Pero también, puede ser un servidor de diálogo, si únicamente cuenta con procesos de trabajo de diálogo.

#### 3.5.1. La instancia en un ambiente ABAP+Java

Hasta ahorita, hemos hablado de los servidores de aplicaciones ABAP y Java como dos entidades separadas. Sin embargo, también es posible tener sistemas que utilizan ambos servidores de aplicaciones al mismo tiempo. Cuando hablamos de un sistema con estas características, el término instancia se refiere a los procesos ABAP, así como a los procesos Java. Entonces, la instancia de un sistema ABAP+Java incluirá todos los siguientes procesos y servicios: el despachador ABAP y sus procesos de trabajo, el gestor de comunicación por internet ("ICM", por sus siglas en inglés), el despachador Java y sus procesos de servidor Java.

En la siguiente figura (fig. 3.5.1-1) se muestra la arquitectura de un sistema SAP con varias instancias utilizando los ambientes ABAP y Java.

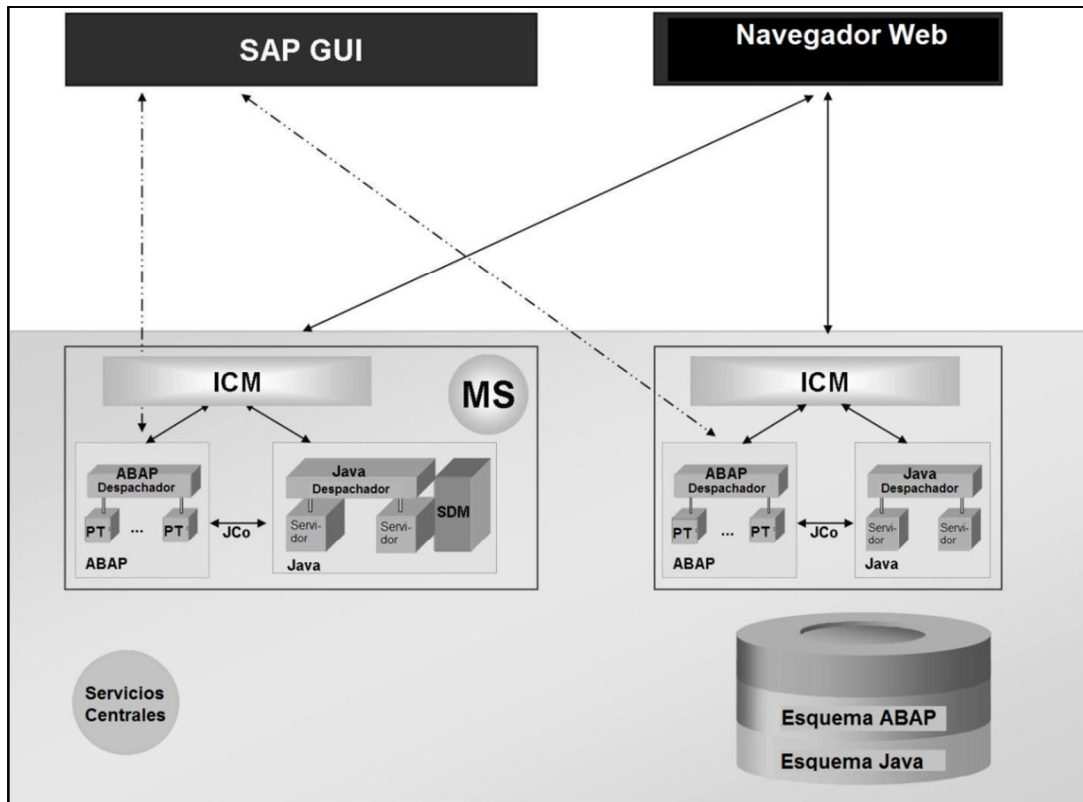


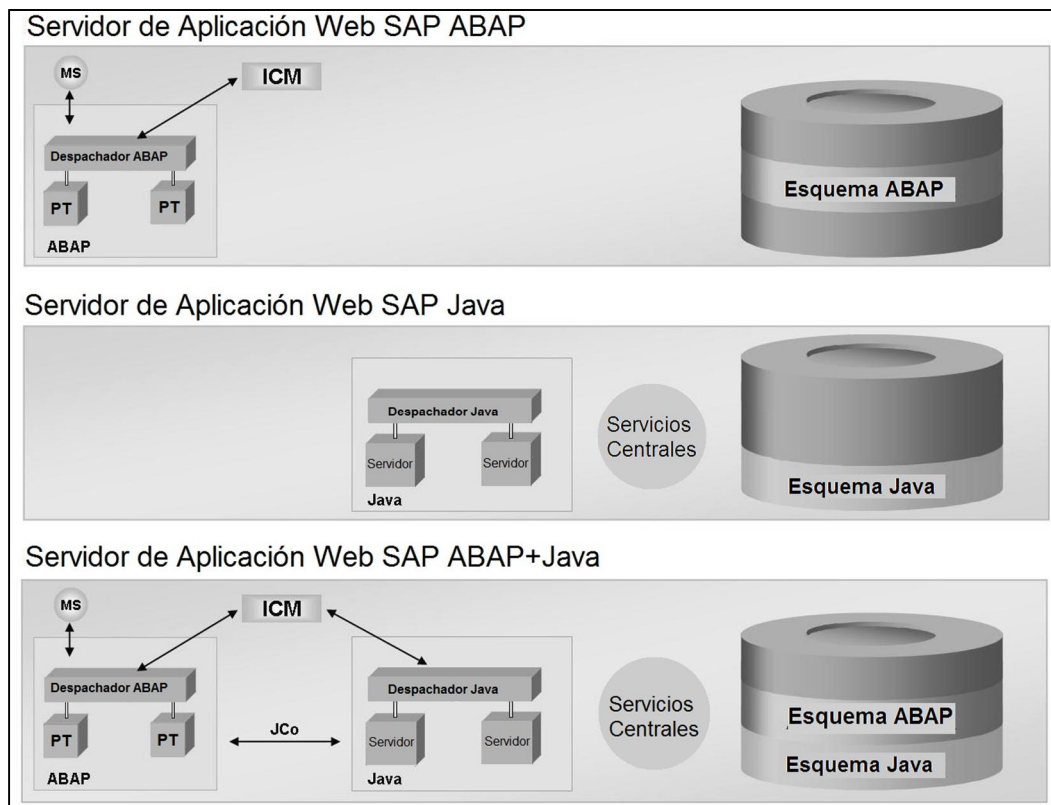
Fig. 3.5.1-1 Estructura de un servidor de aplicaciones con varias instancias [TADM10\_22005]

El gestor de distribución de software (SDM, Software Deployment Manager) es la herramienta estándar con la cual se instalan los componentes J2EE en el servidor de aplicaciones Java. Es parte del entorno de ejecución de Java y siempre es instalado como parte de la instancia central de Java.

Los servicios centrales forman una instancia aparte. Consisten del servicio de mensajes y del servicio de encolamiento (“enqueue”) y representan la base de la comunicación y sincronización para el entorno de ejecución de Java [TADM10\_22005].

### 3.5.2. Tipos de servidor de aplicaciones SAP

Dependiendo de los requerimientos de los procesos de negocio, los servidores de aplicaciones SAP se pueden configurar de las siguientes formas como se muestra en la figura 3.5.2-1.



**Fig. 3.5.2-1 Posibles tipos de servidor de aplicaciones web SAP [TADM10\_22005]**

- Servidor de aplicaciones web SAP ABAP: Infraestructura completa en la cual las aplicaciones basadas en ABAP pueden ser usadas y desarrolladas.
- Servidor de aplicaciones web SAP Java: Infraestructura completa para el uso y desarrollo de aplicaciones basadas en Java.
- Servidor de aplicaciones web SAP ABAP+Java: Infraestructura completa con la cual se pueden desarrollar y usar aplicaciones basadas tanto en ABAP como en Java. Este tipo de instalación hace notar la perfecta integración entre ABAP y Java.

Una de las principales características del servidor de aplicaciones SAP es que las tablas, programas e información de aplicación ABAP son guardados en el esquema ABAP de la base de datos mientras la información de Java se almacena en el esquema Java. El entorno de ejecución ABAP se encarga de acceder al esquema ABAP de la base de datos, en tanto que el entorno de ejecución Java se encarga de acceder el esquema Java de la base de datos. El conector Java de SAP (SAP Java Connector en inglés, JCo) tiene como tarea establecer la comunicación entre los ambientes ABAP y Java [TADM10\_22005].

## 4. Implementación técnica del sistema

La implementación técnica de SAP conlleva una serie de actividades y prerequisites que están enfocados a preparar la instalación y configuración técnica del "landscape", entendiéndose por "landscape", la infraestructura técnica que soportará la operación de los sistemas, durante esta etapa se instalarán los ambientes para dar inicio con la implementación y configuración de los requerimientos de negocio por parte del equipo funcional [SCTI2012].

Como parte de las actividades iniciales, será necesario realizar la definición de qué arquitectura y plataforma se utilizará para realizar la instalación inicial del sistema, será necesario establecer las versiones e infraestructura técnica asociada a los servidores, sistema operativo, base de datos así como los productos y versiones de SAP que serán instalados.

La infraestructura técnica utilizada se deberá analizar en un proyecto de dimensionamiento, en donde en base a la carga de trabajo y cantidad de usuarios que se tenga estimada a futuro, se deberá establecer la cantidad de recursos de hardware requeridos para soportar la nueva plataforma. Este análisis deberá establecer a nivel de detalle qué tanta memoria física como virtual deberá ser configurada para cada uno de los ambientes, la capacidad de CPU requerida, el espacio en disco para cada uno de los sistemas, el ancho de banda de la red necesario para garantizar un tiempo de respuesta óptimo entre el centro de datos, así como cada una de las localidades del negocio que tendrán acceso al sistema.

### 4.1. Planeación

Para llevar a cabo la implementación de la nueva plataforma, será necesario diseñar el escenario que será necesario instalar de acuerdo a los requerimientos de negocio que se establezcan desde un inicio, tomando como base los alcances y objetivos iniciales que busca la empresa con la implementación de SAP, para esto es indispensable determinar qué sistemas son requeridos y para qué se utilizará cada uno de los productos que serán instalados [AS2004].

Una vez que se tengan identificados las instancias de SAP que se implementarán, será necesario trabajar en conjunto con el proveedor de hardware e infraestructura para identificar los prerequisites técnicos, así como la infraestructura requerida que cumpla con las características de compatibilidad y capacidad para soportar la solución.

En la figura 4.1-1 se ilustra un "landscape" convencional que incluye la implementación de diversos productos tales como PI, BW así como un portal, entre otros.



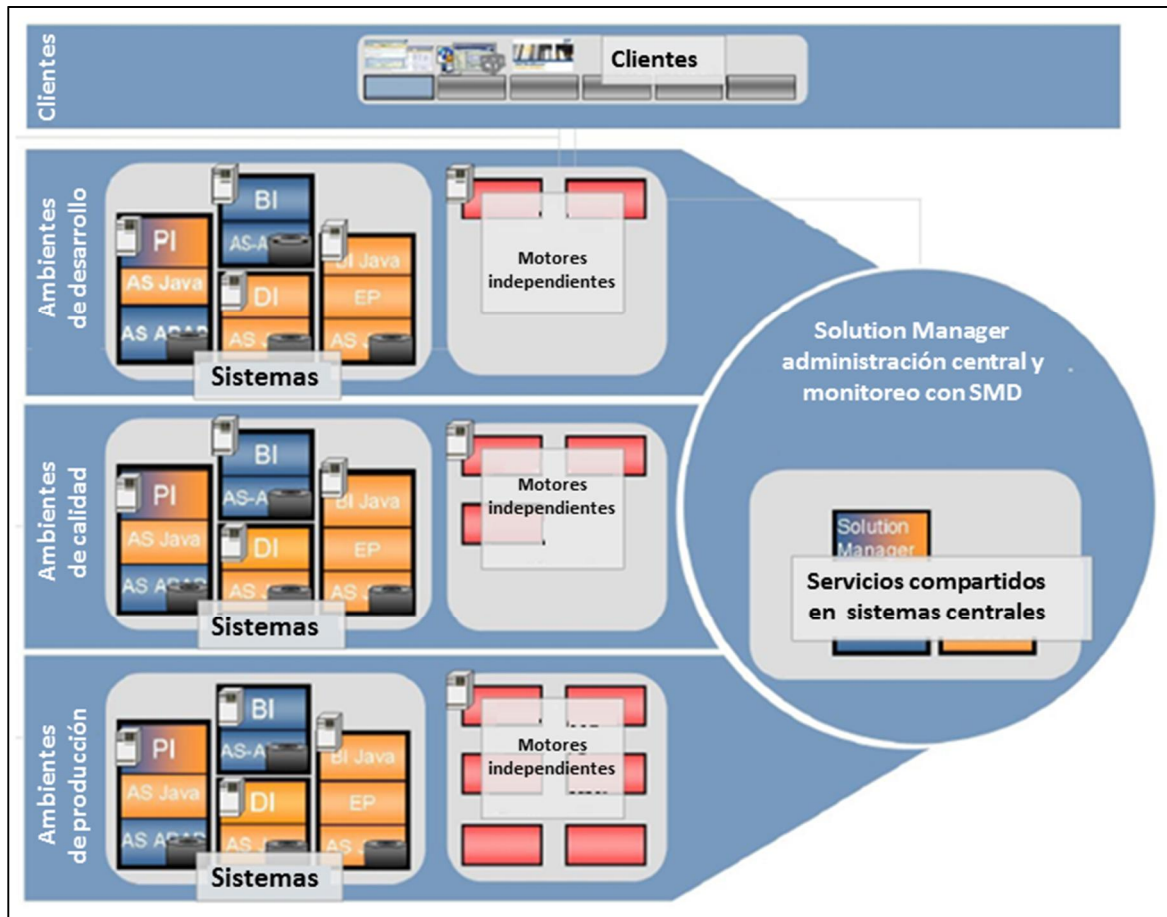


Fig. 4.1-1 Implementación de soluciones [AS2004].

A continuación se describen los puntos clave que se deberán tomar en cuenta para el diseño de la solución:

1. Determinar los escenarios de TI que se desearán implementar, tomando como base la plataforma Netweaver.
2. Determinar las unidades de software requeridas que serán instaladas en atención a los escenarios de TI definidos, entre ellos se incluyen los "usage types" que hacen referencia al producto asociado, ya sea un servidor de aplicación ABAP, uno de Java, componentes de "business intelligence" (inteligencia de negocios) o portales, entre otros. De la misma manera, es necesario realizar la definición de los clientes necesarios para establecer la conexión con el principal de negocio (ERP, SCM, CRM, APO) llamado "back-end", así como los tipos y versiones asociadas a la interface con el usuario final llamada "front-end". En la figura 4.1-2 se muestra un ejemplo de estos componentes, así como la relación existente entre estos y cada uno de los clientes del sistema.

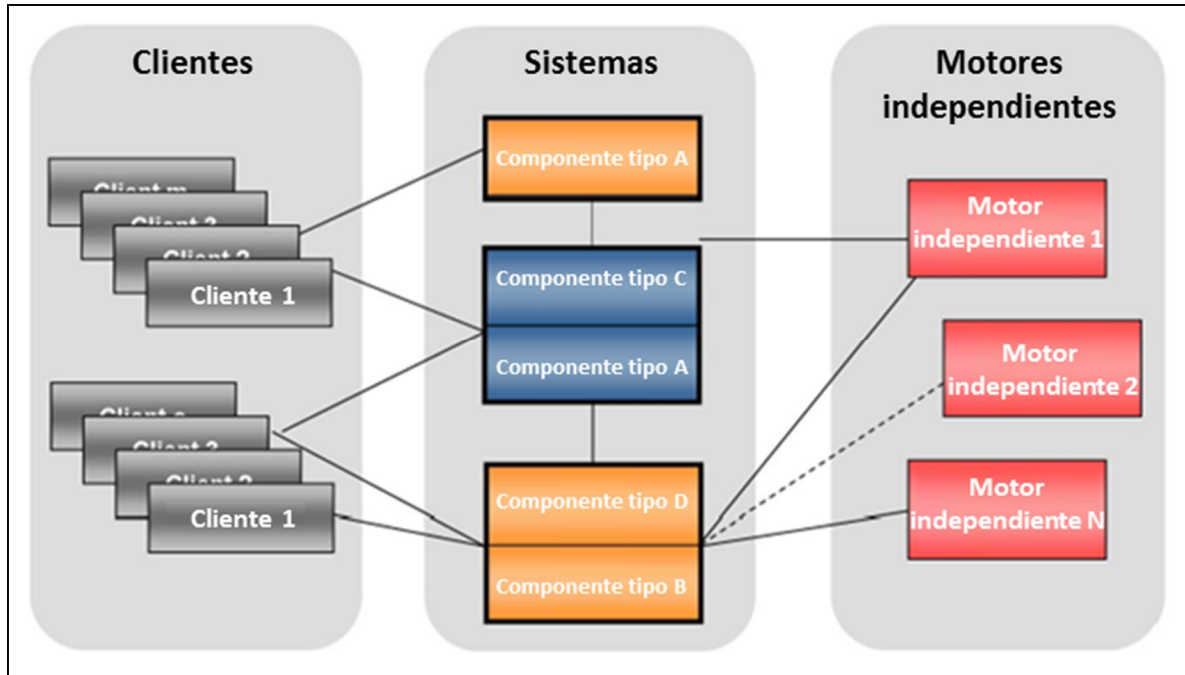


Fig. 4.1-2 Componentes de Software [AS2004].

- Determinar los servicios compartidos que serán necesarios para la implementación de la solución.
- Definir puntualmente el "landscape", esto es, se decidirán cuántas instancias de SAP deberán ser instaladas y para qué se utilizará cada una de ellas, generalmente SAP recomienda mantener como base una instancia de Desarrollo, Calidad y una instancia Productiva independiente. Dependiendo del tamaño de la implementación en algunos proyectos se acostumbra instalar adicionalmente una instancia llamada comúnmente "Sandbox" para las pruebas genéricas asociadas a actualización de versiones instalación de parches de software entre otros, así como un ambiente de Preproducción, enfocado a las pruebas de estrés y performance antes de la salida en vivo del sistema.

#### 4.2. Preparación del sistema

De acuerdo a nuestra experiencia, hemos identificado que antes de proceder con la ejecución e instalación técnica del sistema es necesario tomar en cuenta algunas actividades, con el objetivo de garantizar una instalación exitosa, así como un ambiente lo suficientemente robusto para garantizar la operación y estabilidad del sistema, a continuación se describen los principales puntos que se deberán tomar en cuenta antes de iniciar con la instalación:

- Es necesario verificar claramente los requerimientos tanto de software como de hardware, tomando como base el proyecto de dimensionamiento realizado para determinar los recursos necesarios en base a la carga de trabajo estimada.
- Verificar la estructura del sistema de archivos en la cual se realizará la distribución de servicios y componentes de la base de datos, ya sea que se trate de una instalación sobre plataforma UNIX, o por otra parte, las particiones de disco ya sean físicas o virtuales, en dado caso que se realice una instalación sobre una plataforma con Windows.
- Identificar si se realizará una instalación de tipo local o de dominio para el caso de la plataforma Windows, esto con el objetivo de garantizar la aplicación de políticas de seguridad en el dominio.
- Verificar que los usuarios de instalación generados, cuenten con los privilegios y autorizaciones necesarias para realizar la instalación del sistema ya sea del tipo local o de dominio, en el caso de que se desee realizar una instalación de dominio, es indispensable que los usuarios de instalación cuenten con privilegios de administración en el dominio, para la generación de cuentas y grupos de dominio necesarios para la administración de los servicios de SAP.

5. Identificación del directorio de transportes así como del servidor destinado para esta tarea.
6. Considerar un plan de trabajo para la instalación del cliente de software o “front-end” en las terminales de los usuarios finales, con el objetivo de permitir la conectividad con SAP.
7. Asegurar la descarga y copia de las medias de instalación desde el “marketplace” o portal central de soporte de SAP hacia cada uno de los servidores en donde sea requerido, de la misma manera es importante que se considere la instalación de “Solution Manager” como una de las primeras implementaciones ya que para las nuevas versiones de “Netweaver”, las herramientas solicitarán una llave generada por este sistema para permitir la instalación de SAP.

### **4.3. Requerimientos de instalación**

La experiencia nos ha demostrado que es indispensable verificar que las versiones de SAP que serán instaladas en la nueva plataforma sean compatibles con la versión de sistema operativo y base de datos que se está contemplando, así como la arquitectura del servidor elegida ya sea de 32 bits o de 64 bits, de otra manera se pueden presentar posteriores asociadas a temas de compatibilidad que impedirán continuar con los siguientes pasos de la instalación.

En el caso de que no se cumplan los requerimientos básicos, es muy probable que se presenten problemas que no permitan concluir con la instalación de las instancias exitosamente, por este motivo SAP cuenta con un portal conocido como PAM (Product Availability Matrix, por sus siglas en inglés) o matriz de disponibilidad de productos, el cual cuenta con una base de datos con todas las versiones que han sido liberadas con el objetivo de consultar los productos y versiones soportadas para cada uno de ellos.

El ingreso al PAM se realiza por medio de la URL “<http://service.sap.com/pam>”, pero en este caso es indispensable contar con un usuario registrado y autorizado por SAP para acceder a este sistema.

### **4.4. Tipos de instalación**

Una vez que se verifica la compatibilidad de los componentes tanto de hardware como de software que serán implementados, es indispensable identificar el tipo de instalación que se llevará a cabo. En el caso particular de plataformas Windows, es indispensable identificar si se realizará una instalación de tipo local, en la cual los servicios y usuarios de comunicación se instalarán en el equipo seleccionado únicamente, o en dado caso definir si será necesario realizar una instalación de dominio.

Existe la posibilidad de realizar una configuración instalación en alta disponibilidad, por medio de la cual se realizará una distribución compartida de los servicios tanto a nivel aplicación, base de datos como sistema operativo, con el objetivo de balancear la carga entre diferentes servidores y con esto buscar mitigar los puntos de falla en el ambiente. Con esta configuración se buscará incrementar la disponibilidad de los sistemas, cabe señalar que la alta disponibilidad, generalmente se establece en ambientes productivos de operación crítica para el negocio, por cuestiones principalmente de la complejidad y los costos que están asociados a dicha configuración [BS2011].

#### **4.4.1. Instalación local**

Con las instalaciones que hemos realizado durante nuestra experiencia profesional, hemos constatado que para las instalaciones locales que se realicen sobre plataforma Windows, el usuario de instalación deberá formar parte del grupo de administradores locales del servidor asociado, en este caso todas las cuentas de servicio e información generada para la creación de servicios, se almacenarán de manera local y no serán visibles para los demás ambientes restantes del “landscape”.

#### **4.4.2. Instalación de dominio**

En el caso que se elija realizar una instalación de dominio sobre plataformas Windows, es indispensable que el usuario generado sea parte del grupo de administradores tanto local, así como también administrador del dominio elegido para la instalación, todos los ambientes que alojarán la solución SAP de este sistema llámese desarrollo, calidad y producción, deberán formar parte del mismo dominio.

En caso de que se desee realizar una instalación en alta disponibilidad en la cual los servicios que darán soporte a la aplicación estarán instalados en diversos servidores, todos ellos deberán ser parte del mismo dominio, de la misma manera se recomienda utilizar la misma contraseña para todos los usuarios de servicios que sean creados, con el objetivo de evitar posibles problemas de permisos asociados a la administración del sistema.

#### 4.4.3. Configuración en alta disponibilidad

La configuración en alta disponibilidad busca en todo momento reducir al mayor grado posible el tiempo de indisponibilidad que se pueda presentar en aplicaciones de operación crítica, el objetivo será establecer un esquema que permita mantener la operación del sistema antes situaciones tales como fallas de hardware, software, errores humanos o inclusive actividades de mantenimiento programadas en los equipos, como la actualización de componentes de software entre otros [BS2011].

En el ámbito de las tecnologías de la información se utiliza el concepto “clúster” para hacer referencia al componente asociado para mantener la alta disponibilidad, existen tres tipos de clúster en general:

1. Clúster a nivel de hardware: Encargado de mantener la alta disponibilidad a nivel de componentes físicos del servidor.
2. Clúster a nivel de aplicación: El cual se encarga de realizar la distribución de los componentes de software a nivel SAP en todos los equipos involucrados
3. Clúster a nivel base de datos: Encargado de mantener las instancias de base de datos disponibles para el procesamiento de información.

Esta configuración generalmente establece el hecho de mantener un servicio disponible en diversos equipos físicamente, el cual se proporciona a los clientes por medio de un despachador de peticiones o un balanceador de cargas, que se encargará de distribuir las peticiones entre diversos aplicativos, así como de mantener un monitoreo constante sobre los servicios y redirigir las peticiones sobre los aplicativos disponibles.

Generalmente se busca establecer un método de redundancia para los componentes de hardware también, por medio de la inclusión física de servidores adicionales así como esquemas en los cuales se configuran varias interfaces de red, para mantener la disponibilidad a nivel de comunicaciones, esto con el objetivo de que si el sistema identifica una falla en algún componente en particular, el componente de respaldo entre automáticamente en operación para no afectar la disponibilidad del servicio hacia el usuario final.

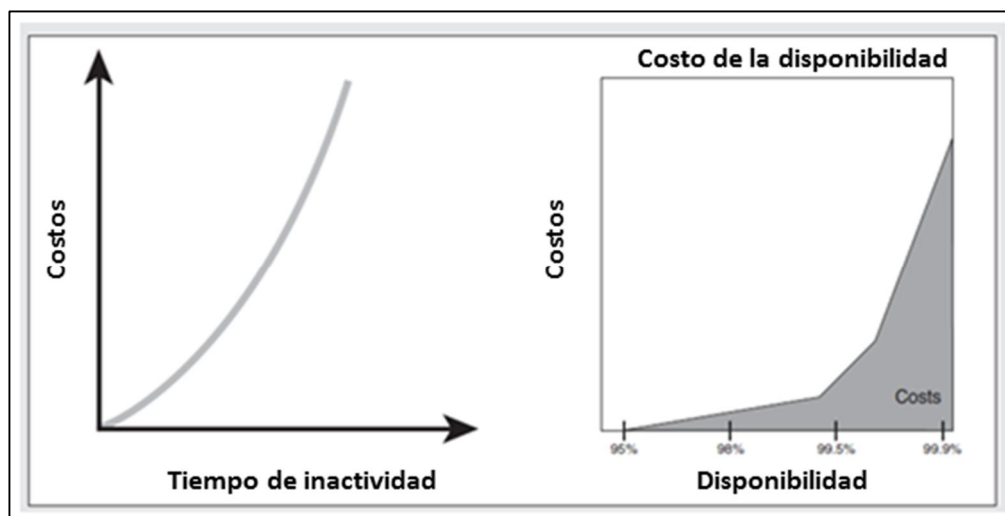


Fig. 4.4.3-1 Implementación de una solución de alta disponibilidad.

Como se mencionaba anteriormente, la implementación de un ambiente en alta disponibilidad incrementa fuertemente los costos del proyecto, la lámina 4.4.3-1 ilustra el comportamiento asociado

tanto a costos de implementación por porcentaje de disponibilidad buscado, entre mayor disponibilidad se quiera tener, el costo de implementación crecerá de manera significativa, por esta razón es indispensable realizar una evaluación que permita poner en balanza los costos asociados a esta implementación por un lado, así como los posibles costos o pérdidas financieras asociadas al tiempo de indisponibilidad de un ambiente productivo de operación crítica para el negocio.

A continuación en la figura 4.4.3-2, se muestra la arquitectura de un clúster de SAP así como la distribución asociada de servicios a nivel aplicación:

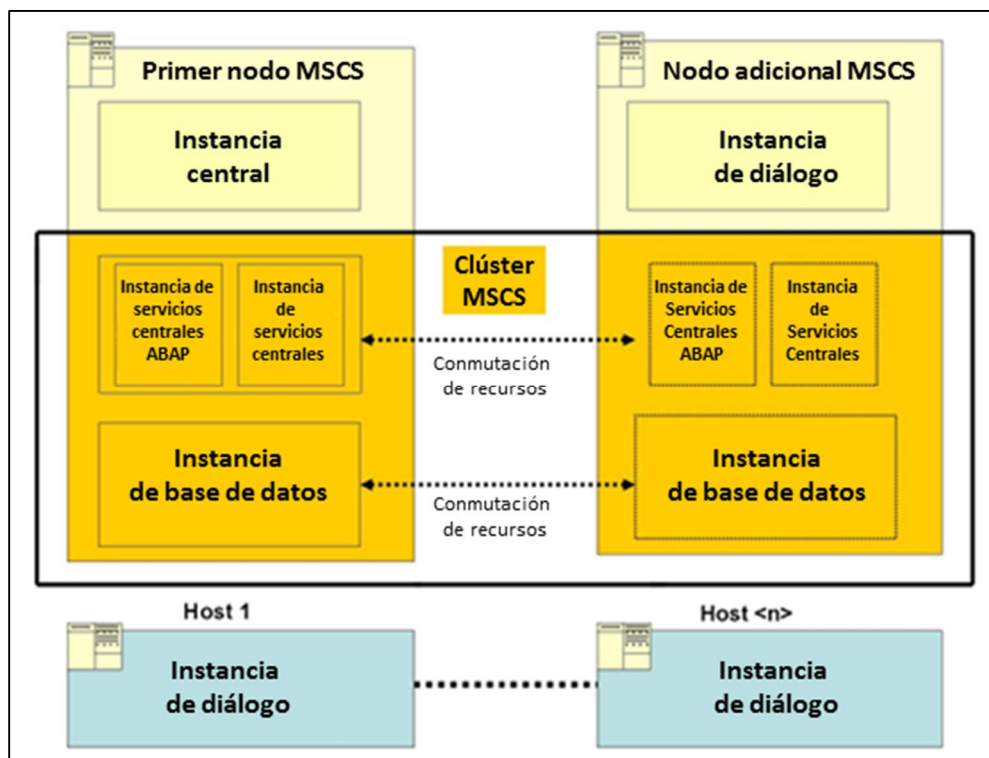


Fig. 4.4.3-2 Distribución del clúster [BS2011].

Como se muestra en la imagen, el diseño de este ambiente de alta disponibilidad está montado en dos servidores físicamente.

El motor de base de datos se encuentra como parte de un recurso compartido del clúster para en caso de alguna falla en el nodo primario, poder realizar el desplazamiento de recursos o "failover" hacia el nodo secundario.

Por parte de la aplicación, es necesario realizar de entrada la instalación de los servicios del clúster como tal para la administración de la herramienta. Adicionalmente se requiere contar con la instalación de los servicios centrales de ABAP así como los servicios centrales de instancia para permitir la comunicación y procesamiento interno de SAP.

De manera complementaria y con el objetivo de mantener la disponibilidad, esta arquitectura establece la instalación de una instancia central en el nodo primario, así como de al menos una instancia de diálogo en el nodo secundario, es importante señalar que es posible considerar la inclusión de "n" número de instancia de diálogo o servidores aplicativos, con el objetivo de incrementar la disponibilidad y escalabilidad del sistema por un lado, así como realizar un balance de cargas más eficiente en proporción a la actividad que tendrá el sistema.

#### 4.5. Componentes de software necesarios

La plataforma de Netweaver está enfocada en la integración de la gente, los procesos de negocio y la información bajo una misma estructura tecnológica capaz de mantener la información en línea y disponible en todo momento como el ambiente empresarial actual lo demanda, esta plataforma cuenta con la implementación de un amplio rango de integración con diversas tecnologías tales como Microsoft .NET, Java J2EE así como IBM Websphere entre otros [AS2004].

La implementación de SAP involucra realizar la definición de los escenarios de TI, tomando como base los componentes de software que será necesario instalar de un inicio, para dar atención a los requerimientos generados por parte del cliente.

Estos componentes o unidades de software deberán ser identificados y mapeados con el escenario de negocio que será implementado, con el objetivo de dar atención a los requerimientos de negocio como se muestra en la figura 4.5-1.

Escenario TI	Variante del escenario	Tipo de componente
<b>Portal empresarial</b>	Todas las variantes del escenario TI	-AS Java - EPC - Infraestructura de desarrollo
<b>BW empresarial</b>	Todas las variantes del escenario TI	- AS ABAP - AS Java - BI - EPC - EP - BI Java
<b>Administración de procesos de negocio</b>	Todas las variantes del escenario TI	- AS ABAP - AS Java - PI
<b>Movilidad en procesos de negocio</b>	Ejecución de aplicaciones móviles en línea	- AS Java

**Fig. 4.5-1 Unidades de software [AS2004].**

SAP Netweaver se compone básicamente de los siguientes tres tipos de unidades de software:

1. Sistemas que son configurados para dar atención a un caso en específico de negocio, los cuales se componen por un conjunto de componentes conocidos como "usage types".
2. Motores de aplicación independientes o "standalone", enfocados a dar atención a una función específica en combinación con una o más aplicaciones de Netweaver.
3. Clientes instalados de manera local en los equipos de los usuarios, enfocados a permitir el acceso al "landscape" de SAP.

La figura 4.5-2 muestra un ejemplo de esta arquitectura.

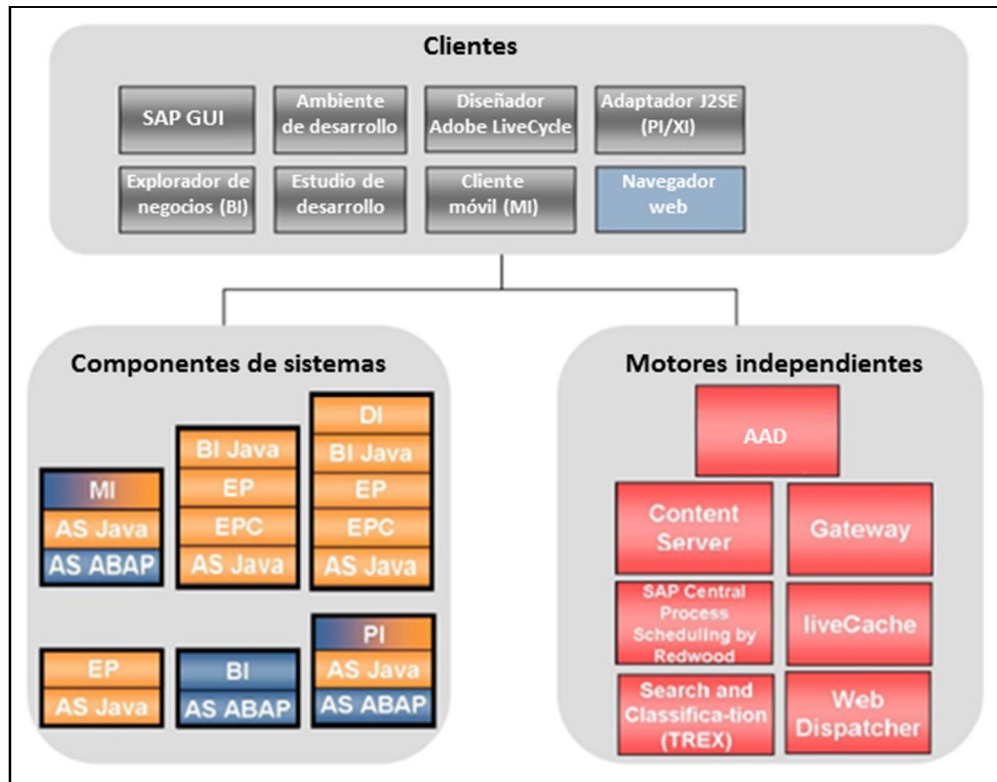


Fig. 4.5-2 Arquitectura de la solución [AS2004].

Los componentes de software conocidos como “usage types” de SAP representa el bloque principal a partir del cual se construye una solución sobre la plataforma de Netweaver, estos elementos son la estructura técnica esencial para la solución SAP y permiten mapear la lógica de un escenario de TI basado en SAP estableciendo los roles y funcionalidad general de cada instancia del “landscape”.

#### 4.6. Instancia central y servicios centrales

La implementación de SAP conlleva realizar la instalación de una serie de instancias centrales las cuales son distribuidas a lo largo de todo el “landscape”, como ambientes independientes que se encargan de proveer servicios compartidos a los usuarios, y que además se encuentran interconectados entre sí.

El hecho de realizar la instalación de SAP en base a instancias independientes con características únicas, permite reducir la complejidad de implementación del proyecto y realizar una administración y ejecución de cambios de una forma más ordenada. En una implementación típica de SAP, generalmente es común encontrar la instalación de las siguientes instancias [AS2004]:

1. ERP (Enterprise Resource Planning)
2. Portal empresarial (Enterprise Portal)
3. Inteligencia de negocios (Business Intelligence)
4. Integrador de procesos (Process Integrator)
5. Tecnología móvil (Mobile)
6. Infraestructura de desarrollo (Development infraestructure)
7. Solution Manager, herramienta utilizada para la gestión e implementación del proyecto, entrenamiento, pruebas, monitoreo y administración de la infraestructura, así como también utilizada para la gestión y administración de cambios e incidentes.

El concepto de instancia central como tal, es un concepto utilizado por SAP el cual hace referencia a los recursos tanto de hardware como de software, por medio de los cuales se proporcionan los servicios necesarios para mantener el sistema en operación, la instancia central se compone entonces del servidor físico así como de numerosos componente de software y servicios asociados entre los cuales se encuentran principalmente los siguientes [SHSE2011]:

1. Instancia de Base de Datos, responsable del almacenamiento y administración de los datos.

2. Servidor de mensajes, encargado de establecer la comunicación entre los servicios internos de la instancia.
3. Procesos de actualización, responsables de la ejecución de actualizaciones en la Base de datos.
4. Procesos de bloqueo, encargados de la administración de bloqueos para la protección de registros y consistencia de datos.
5. Procesos de diálogo, enfocados a la ejecución de los pasos de diálogo para los programas así como la atención de las peticiones interactivas de los usuarios.
6. Procesos de fondo, utilizados para la ejecución de reportes y tareas de mantenimiento programadas en fondo.
7. Procesos de spool, utilizados para el formateo y administración de las órdenes de impresión.

La figura 4.6-1 muestra un ejemplo en donde se ilustran algunos de estos componentes.

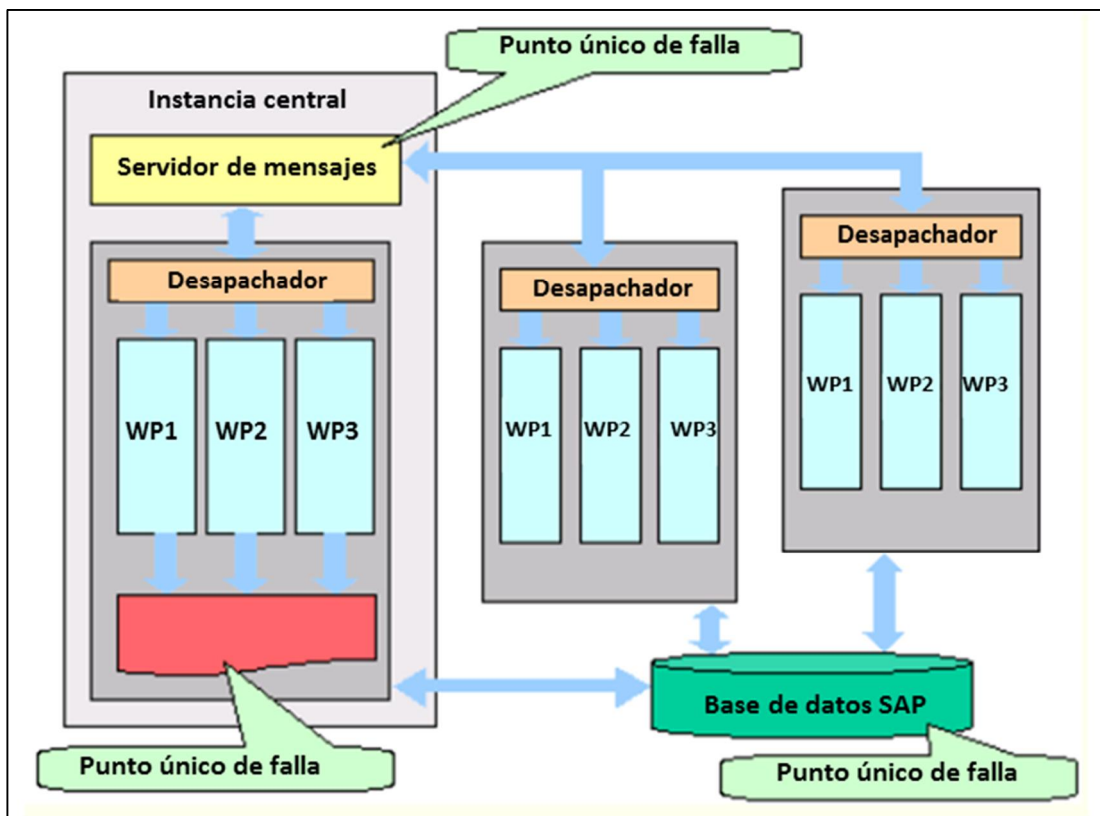


Fig. 4.6-1 Componentes del sistema [SHSE2011].

Con la experiencia hemos constatado que por medio de la transacción SM50, es posible visualizar los procesos de trabajo de una instancia central como se muestra en la figura 4.6-2:



No	Ty.	PID	Status	Reasn	Start	Err	Sem	CPU	Time	Report	Cl.	User	Action
0	DIA	5804	running		Yes					SAPLTHFB	000	SAPADMIN	
1	DIA	15916	waiting		Yes								
2	DIA	2832	waiting		Yes								
3	DIA	8356	waiting		Yes								
4	DIA	5628	waiting		Yes								
5	DIA	8328	waiting		Yes								
6	DIA	8312	waiting		Yes								
7	DIA	8304	waiting		Yes								
8	DIA	4456	waiting		Yes								
9	DIA	9144	waiting		Yes								
10	DIA	5096	waiting		Yes								
11	DIA	4472	waiting		Yes								
12	UPD	9136	waiting		Yes								
13	ENQ	8340	waiting		Yes								
14	BGD	21496	waiting		Yes								
15	BGD	9044	waiting		Yes								
16	BGD	9616	waiting		Yes								
17	BGD	19416	waiting		Yes								
18	BGD	8400	waiting		Yes								
19	BGD	16608	waiting		Yes								
20	SPO	20344	waiting		Yes								

Fig. 4.6-2 Procesos de trabajo.

El administrador del sistema conocido comúnmente como administrador de tecnología “base” o “basis” de sus siglas en inglés, por soportar la plataforma base, este recurso es el responsable de contrarlar el número de procesos de trabajo que se definirán en cada uno de los servidores aplicativos, la distribución se deberá realizar tomando en cuenta los recursos de hardware con los que cuenta el ambiente, así como también la carga de trabajo identificada en el sistema.

De la misma manera, el administrador tendrá la capacidad de generar los modos de operación, por medio de los cuáles se controlará el número y tipo de procesos de trabajo que estarán disponibles durante ciertos horarios en función a la carga que tenga el sistema, generalmente la recomendación es mantener un alto número de procesos de diálogo durante los horarios laborables que suelen ser durante el día, y posteriormente programar un cambio en el número de procesos que se asignarán durante la noche, incrementando los procesos de trabajo en fondo para la ejecución de los reportes con mayor carga, así como los procesos de mantenimiento que se ejecutan en el sistema para mantenerlo operando en condiciones óptimas.

Por medio de la transacción RZ04 es posible administrar los horarios de operación así como la distribución de cada uno de los procesos de trabajo, como se muestra en la figura 4.6-3:

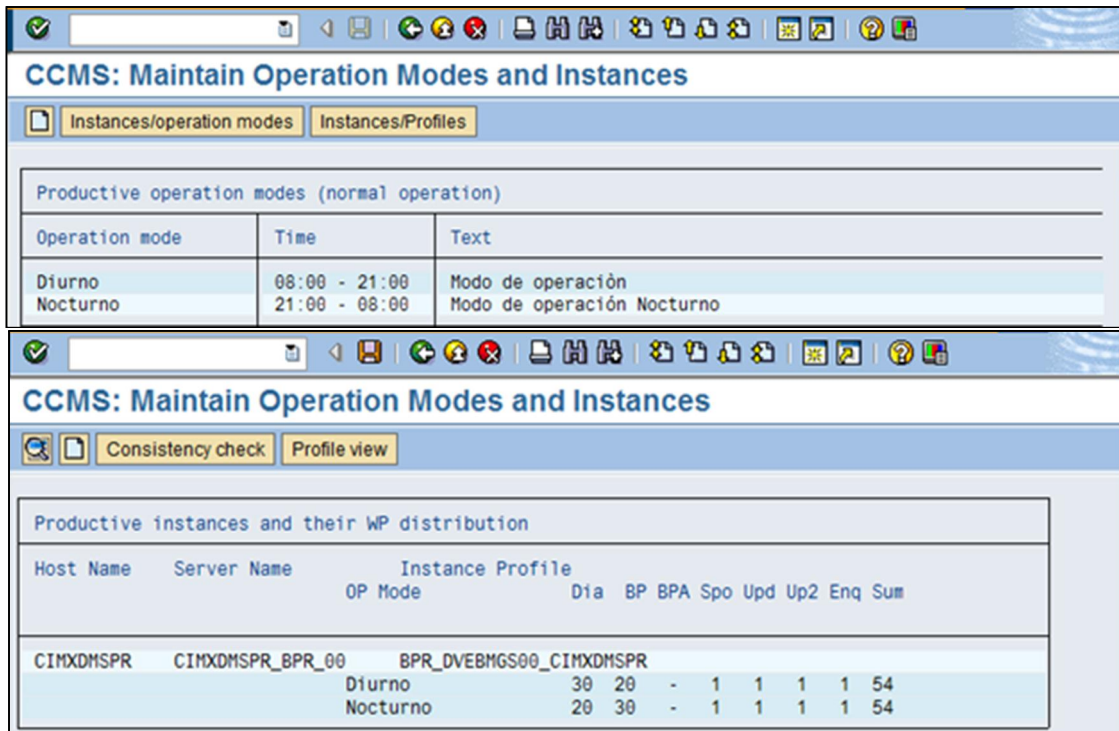


Fig. 4.6-3 Modos de operación.

Los servicios centrales de SAP son utilizados para la comunicación de la pila de Java [SHCS2011]. Estos servicios se componen básicamente de los procesos conocidos como “servicio de mensajes” y “servicio de bloqueos”, por medio de los cuales se establece la comunicación y sincronización de los aplicativos basados en soluciones Java, en la figura 4.6-4 se ilustra esta arquitectura.

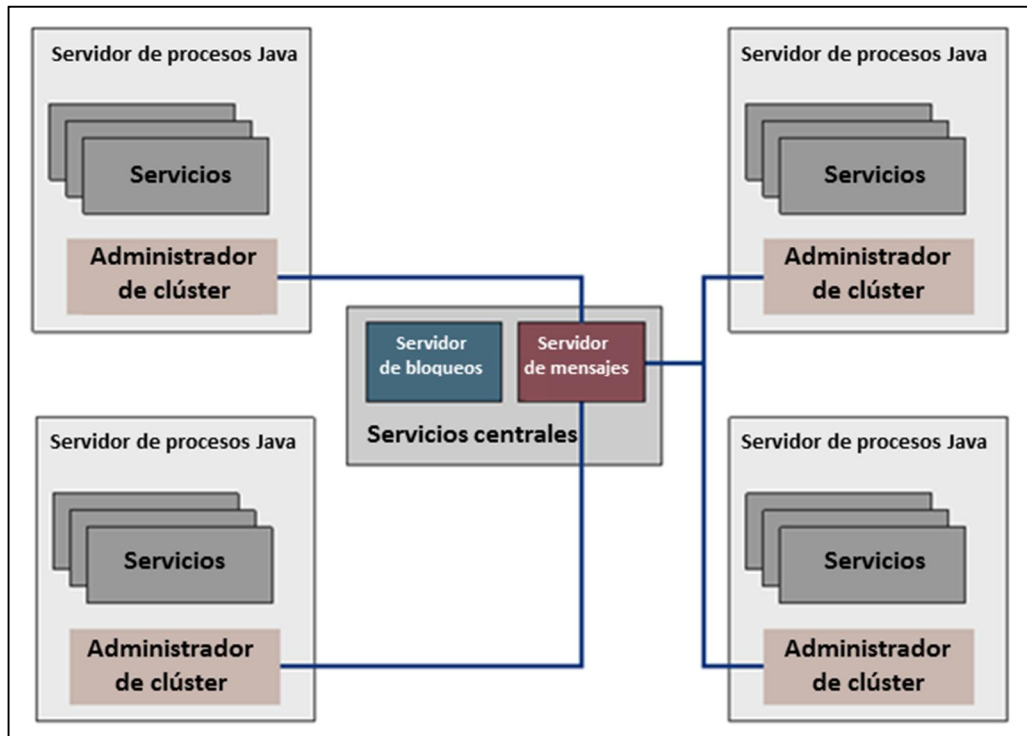


Fig. 4.6-4 Servicios centrales [SHCS2011].

El servicio de mensajes mejor conocido como “message service” es responsable de administrar los procesos despachadores de servicios o “dispatchers” en cada uno de los aplicativos, así como los “process server” o servidores asociados que se encuentran interconectados con cada despachador, estos componentes representan la estructura básica de comunicación para el procesamiento de

información, comunicación interna así como la atención de las peticiones provenientes de los usuarios.

El servidor de mensajes se encuentra en estrecha comunicación también con el Webdispatcher de SAP, el cual es un componente utilizado en instalaciones grandes y complejas en las que se requiere hacer uso de un balanceador de cargas, para distribuir el procesamiento de información entre todos los aplicativos, y ayudar con esto a realizar una mejor utilización de los recursos y mejorar los tiempos de respuesta hacia el usuario final.

Como se menciona originalmente, el servicio de bloqueos o mejor conocido como “Enqueue service” es otro de los componentes fundamentales de los servicios centrales de SAP en un ambiente Java, este componente tiene como principal tarea las siguientes funciones [SHJC2011]:

1. Administrar el bloqueo y liberación de objetos durante la actualización de información a la Base de datos por la ejecución de los procesos internos del sistema así como los procesos generados con la actividad de los usuarios
2. Es utilizado internamente para la sincronización con el clúster de java.

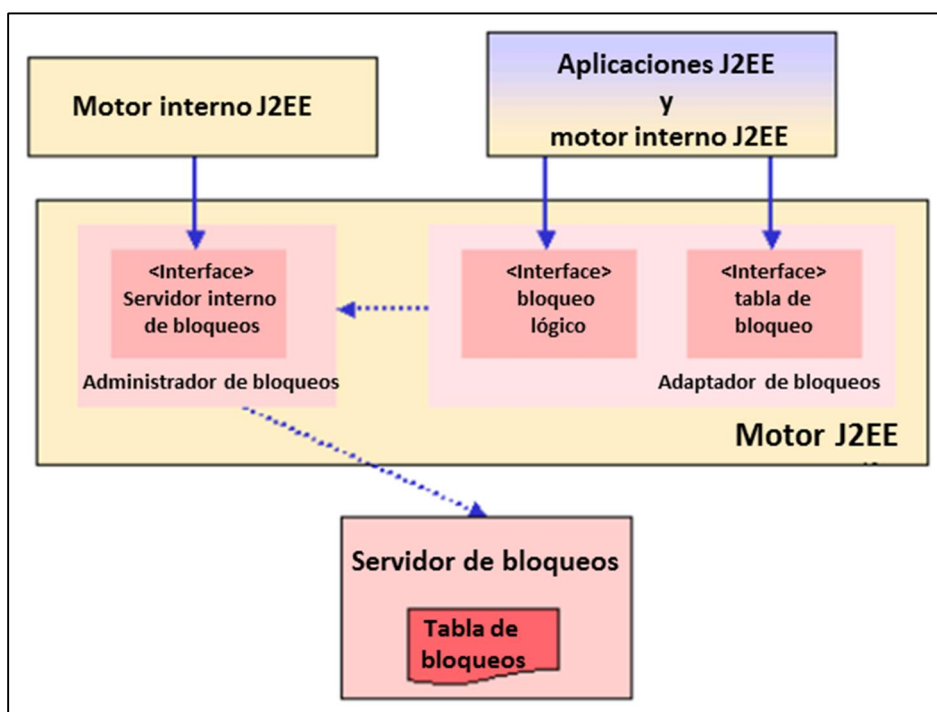


Fig. 4.6-6 Servicio de bloqueos [SHJC2011].

De esta manera el servicio de bloqueos o “enqueue service”, es el responsable de administrar la tabla de bloqueos alojada en la memoria principal, y encargarse con esto de recibir las peticiones asociadas para el bloqueo y liberación de objetos, este comportamiento se ve reflejado cuando por ejemplo varios usuarios están accediendo a una misma transacción en un mismo momento, e intentan actualizar un mismo registro, el servidor de bloqueos será responsable de evitar que se actualice este registro al mismo tiempo por diversos usuarios para evitar con esto temas de inconsistencia en el sistema.

Adicionalmente a los procesos ya señalados podemos incluir también el servicio conocido como SDM (Software Deployment Manager), el cual es un componente de una serie de servicios que permiten realizar la instalación y actualización de unidades de software en el sistema, para apoyar con la implementación y administración de cambios.

Como se muestra en la figura 4.6-7, en general podemos resumir que los principales componentes que integran una solución de Java en SAP se componen por los siguientes [SHJA2011]:

1. Instancia de Base de Datos

2. Servicio de administración de bloqueos como parte de los servicios centrales (Enqueue service)
3. Servicio de administración de mensajes, incluido también en el grupo de los servicios centrales (Message service)
4. Instancia central de Java
5. Despachador de Java (Java dispatcher)
6. Servidor de procesos de Java (Java server process)
7. SDM

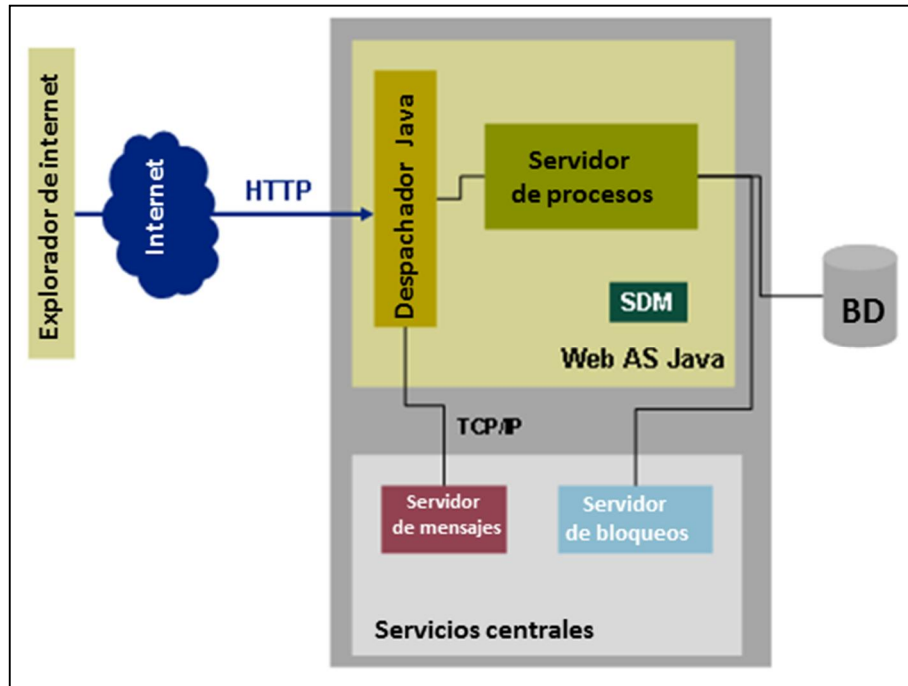


Fig. 4.6-7 Componentes de Java [SHJA2011].

En implementaciones grandes y complejas adicionalmente a la instalación de una instancia central de Java, se suele hacer uso de la instalación de servidores de aplicación de java, con el objeto de incrementar la escalabilidad y performance del sistema como se observa en la figura 4.6-8 [SHJA2011].

Es aquí cuando se requiere tomar en cuenta la adición de un balanceador de cargas ya sea de SAP, como es el caso del "Webdispatcher", o de algún proveedor de tercero que ofrezca este tipo de soluciones para ambientes web.

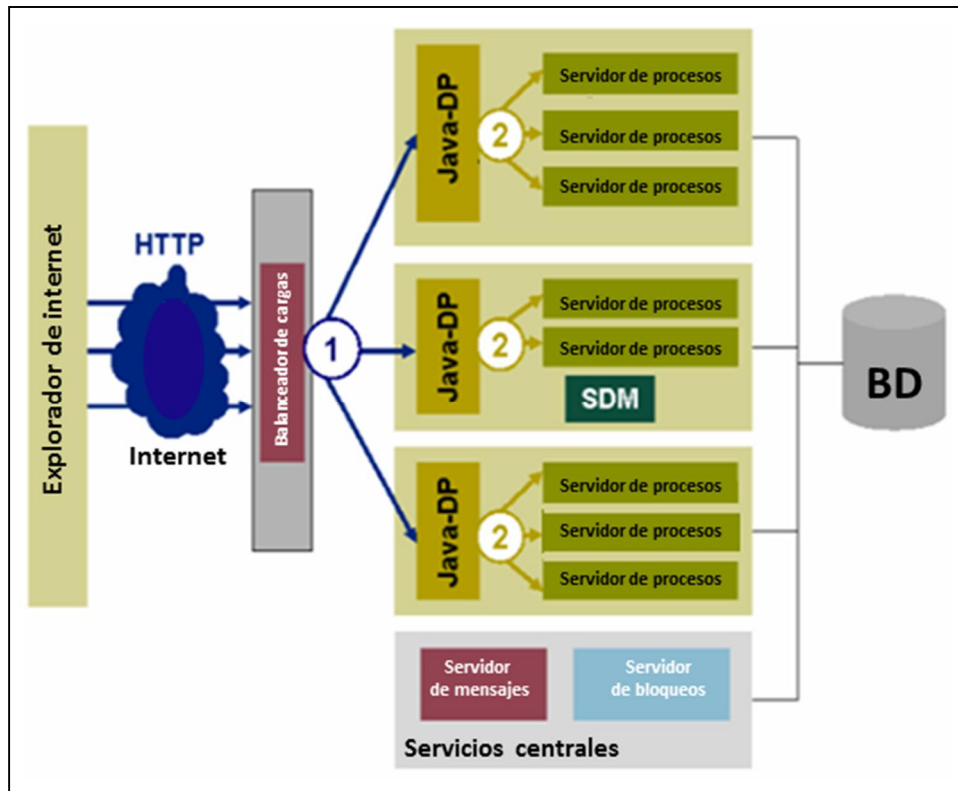


Fig. 4.6-8 Balanceo de cargas en ambientes java [SHJA2011]

#### 4.7. Actividades posteriores a la instalación del sistema

La experiencia nos ha demostrado que una vez que se ha completado la instalación inicial en la infraestructura de cómputo, es necesario considerar una serie de actividades de post instalación, con el objetivo de dejar el sistema listo para que el equipo inicie con las actividades de implementación de acuerdo a los requerimientos del negocio, a continuación listamos algunas de las principales actividades que es necesario tomar en cuenta.

##### Instalación de licencias.

Una de las actividades iniciales que hay que considerar, es realizar la instalación de las licencias a nivel aplicación, estas licencias se solicitan vía remota a SAP tomando como base algunos parámetros tales como el "SID" (System identifier) o identificador de la instancia, el tipo de producto, la plataforma tanto a nivel sistema operativo como la base de datos, así como también una llave de hardware que se encuentra mapeada con la arquitectura del servidor y que deberá ser generada una vez que la instalación inicial haya sido completada.

##### Actualización de componentes de software.

Generalmente, la recomendación es mantener el sistema con los últimos parches liberados por SAP tanto para los componentes de software, así como también para el núcleo mismo de la aplicación conocido también como "kernel", esto con el objetivo de mantener al día las últimas correcciones de software liberadas por el fabricante y mantener una solución más estable en la cual, los errores conocidos han sido ya identificados y solucionados.

SAP mantiene un portal con las últimas actualizaciones y parches liberados para su instalación en el "Marketplace", para poder ingresar a este sistema es necesario contar ya sea con un usuario de cliente, empleado o socio de SAP.

##### Mitigación de riesgos de seguridad.

Parte de las actividades requeridas por el negocio antes de liberar un ambiente a la operación, consiste en el cierre de vulnerabilidades de seguridad identificadas así como la configuración de las

reglas requeridas en el “firewall” o cortafuegos, esto se deberá realizar con el objetivo de evitar ataques ya sea del exterior o desde la propia intranet, así como mitigar todos los riesgos que se puedan presentar en el ambiente, de esta manera se asegurará la solución operando bajo una infraestructura segura y confiable para el manejo, administración y operación de la información del negocio.

### Instalación de la documentación.

SAP ofrece la posibilidad de mantener la documentación formal del sistema disponible en todo momento para los usuarios, con el objetivo de ofrecer la facilidad de realizar consultas en el momento que lo consideren necesario. Para realizar la instalación de esta documentación, es necesario realizar la descarga e instalación de las medias directamente en los aplicativos.

Si no se desea instalar la documentación directamente en el sistema, SAP ofrece el siguiente portal de ayuda, en el cual la documentación se encuentra disponible en todo momento para su acceso en caso de que se requiera realizar alguna consulta. La URL para acceder a este portal de ayuda es <http://help.sap.com>.

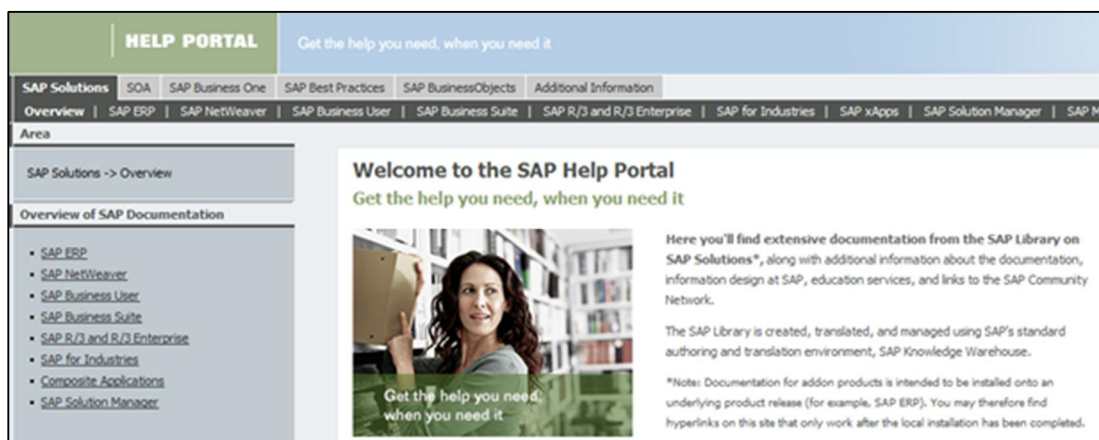


Fig. 4.7-1 Portal de ayuda en línea.

### Configuración de conexiones remotas al equipo de Soporte.

SAP ofrece a sus clientes una serie de servicios de soporte los cuales pueden ser ejecutados de manera remota o en sitio vía consultoría especializada, algunos de estos servicios incluyen la ejecución de reportes de “EarlyWatch” enfocados al monitoreo de rendimiento del sistema, ejecución de servicios de “Going live check” para la preparación de una salida en vivo, así como servicios de soporte remoto continuo en esquemas de 7x24.

Para este fin es necesario considerar la configuración de conexiones remotas que permitan la conexión del equipo de Soporte de SAP hacia el segmento del cliente, esta solución es implementada vía aplicativos conocidos como “SAProuters”.

### Políticas de respaldos

La experiencia profesional nos ha mostrado que con el objetivo de resguardar la información del negocio de manera periódica, es indispensable definir una estrategia de respaldos en la cual se asegure contar con la información disponible en todo momento, con el objetivo de poder recuperar el ambiente ante alguna falla de hardware en los equipos o ante algún problema mayor que requiera la ejecución de una recuperación de un ambiente en el mismo equipo, o en otro servidor físico.

Definir esquemas de respaldos, implica copiar el contenido lógico de los sistemas informáticos a un medio externo que cumpla básicamente con las siguientes características básicas:

1. Confiable: Algunos medios magnéticos tales como las cintas de respaldos, disquetes o discos duros tienen ciertas probabilidades de error o son sensibles ante algún daño físico, es indispensable asegurar el medio así como la estrategia que sea más confiable y permita

- almacenar la información de manera consistente de tal manera que en el momento que se requiera ejecutar alguna recuperación, se realice la copia con los menores inconvenientes.
2. Mantener los dispositivos de respaldos fuera de línea: Una vez que se realice la copia de información. Es necesario desconectar el dispositivo y resguardarlo en un lugar seguro, en el cual se cuenten con las condiciones ambientales adecuadas tales como humedad, temperatura, campos magnéticos entre otros.
  3. Recuperación rápida y eficiente: Es necesario mantener una estrategia de respaldos que considere la ejecución de recuperaciones en los ambientes de manera continua, con el objetivo de asegurar que la información que se está respaldando se recuperará de la mejor manera y en los tiempos adecuados.

La estrategia de respaldos deberá considerar periodos de retención en los cuales se definirá que tanto tiempo se mantendrá el medio de respaldo físicamente en el centro de cómputo, antes de ser depurado de manera definitiva.

#### 4.8. Configuración técnica del sistema

La configuración técnica de del sistema es una de las partes elementales que hay que realizar antes de liberar un sistema al proyecto, existe toda una serie de configuraciones que es necesario realizar como parte de la instalación del ambiente, pero adicionalmente es importante tomar en cuenta que el ciclo de vida del sistema, requerirá modificaciones continuas para dar atención a los nuevos requerimientos, así como también afinar el ambiente con los parámetros necesarios que permitan obtener los mejores tiempos de respuesta.

La recomendación es que todos los cambios que se requieran ejecutar, se realicen mediante un control de cambios como lo establece ITIL, con el objetivo de gestionar las autorizaciones que sean necesarias para realizar la modificación, y llevar un estricto control y documentación sobre todas los cambios realizados en el "landscape" [WIIT2012].

##### 4.8.1. Herramientas de configuración

SAP ha desarrollado una serie de herramientas conocidos como "wizards" las cuales permiten realizar y ejecutar de manera automática las configuraciones esenciales para un determinado escenario de negocio, en la figura 4.8.1-1 se ilustra un ejemplo de un "wizard". Con estas herramientas es posible omitir una serie de configuraciones manuales que conllevan a incrementar los tiempos asignados a las actividades en el proyecto [SHCW2012]:

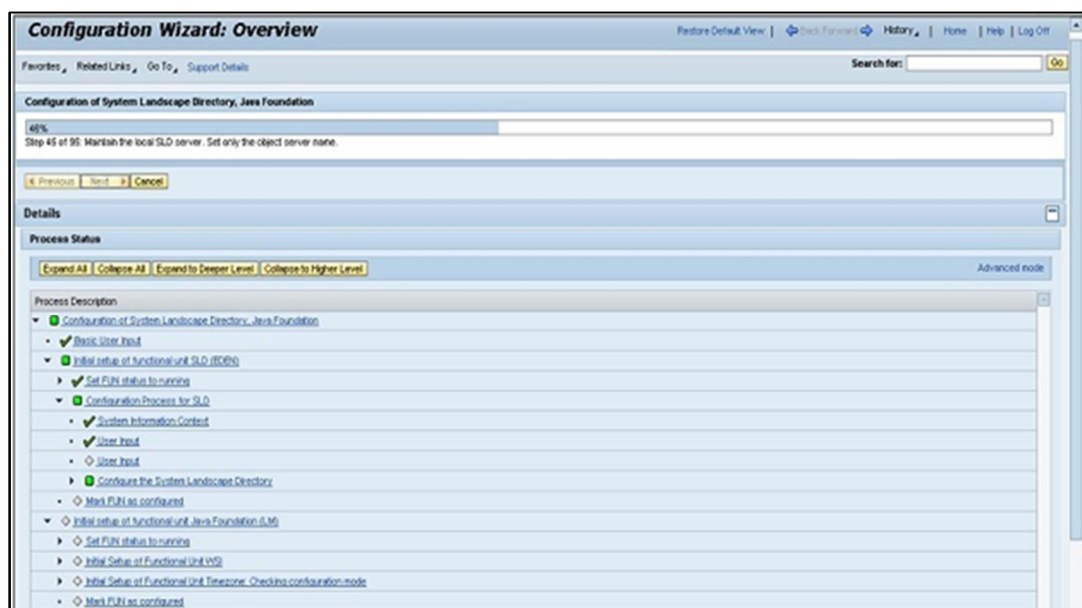


Fig. 4.8.1-1 Herramientas de configuración.

Las recomendaciones de SAP establecen hacer uso de estas herramientas, en lugar de gestionar la solicitud de recursos altamente especializados que ejecuten estas configuraciones de forma manual,

por tanto algunas de las principales ventajas que tenemos al hacer uso de los “Wizards” son las siguientes:

1. Existe la posibilidad de realizar las mismas tareas de configuración con menor tiempo y esfuerzo.
2. Los costos de consultoría asociados a la instalación de los ambientes se reducen de manera considerable, al hacer uso de recursos con menor especialización
3. La configuración e integración se realiza de manera local en base a los estándares planteados por SAP independientemente del recurso que ejecute las tareas.

Es importante señalar que el “wizard” deberá ser ejecutado una sola ocasión, y una vez que se haya completado ya la instalación inicial, así como la actualización de componentes de software y kernel al nivel recomendado, esto con el objetivo de contar con las últimas correcciones liberadas por SAP, así como evitar la corrupción de configuraciones ya generadas con anterioridad.

#### 4.9. Diseño del ambiente de la solución

El diseño de una solución basada en una plataforma SAP suele ser una tarea difícil y restadora para el equipo que emprenderá este proyecto.

Netweaver es una plataforma que contiene una multitud de soluciones y funcionalidades que requiere un conocimiento profundo por parte del diseñador, con el objeto de identificar ¿Qué componentes en específico será necesario implementar? ¿Cómo se implementará la solución? ¿Dónde iniciará el proyecto? entre otros aspectos.

Una de las actividades iniciales para diseñar el “landscape” de la solución, consiste en determinar cuáles son los escenarios y procesos de negocios en específico que se desean implementar, a partir de estas premisas y tomando como base la galería de soluciones que ofrece SAP para el ámbito empresarial, será posible identificar las unidades de software requeridas que deberán ser instaladas [AS2004].

Es importante señalar en este caso que el término “landscape”, hace referencia al conjunto de sistemas que conforman la solución SAP, llámese ambientes productivos y no productivos.

Durante esta etapa se determinará cuantas instancias se requerirán instalar, cuáles serán sus funciones principales, así como de qué manera serán distribuidas en el landscape.

Las instancias SAP son el bloque principal a partir del cual se realiza la construcción de una solución, estos ambientes son identificados por un identificador de sistema único para cada uno de los sistemas que se compone por 3 letras, cada instancia contará con uno o más “usage types” el cual determina el rol asignado para esa instancia y encapsulará la funcionalidad y capacidad asociada, el “usage type” se compone a su vez por los componentes de software específicos para esa solución.

Entre algunos de los principales “usage types” diseñados por SAP se encuentran los siguientes [SCHT2011]:

- Inteligencia de negocios (Business warehouse - BW)
- Inteligencia de negocios java (Business warehouse java – BW Java)
- Infraestructura de desarrollo (Development infrastructure – DI)
- Portal empresarial (Enterprise portal – EP)
- Integrador de procesos (Process Integration – PI)
- Servidor aplicativo ABAP (Application server ABAP – AS ABAP)
- Servidor aplicativo java (Application server Java – AS Java)
- Infraestructura móvil (Mobile infrastructure – MI)

Adicionalmente SAP cuenta con una serie de herramientas que residen de manera local en las computadoras personales de los usuarios y que son utilizadas para acceder a los ambientes instalados en el “landscape”, estas utilerías son conocidas como clientes o “front-ends”, entre algunas de las principales aplicaciones se encuentran las siguientes [SCHT2011]:



- Interfaz gráfica de SAP o también conocido como SAP GUI
- Explorador de negocios o business explorer
- Cliente de infraestructura móvil o mobile infrastructure client
- SAP Netweaver developer studio
- SAP Netweaver developer workplace
- Motor adaptador de java o J2SE adapter engine
- Adobe liveCycle designer

Adicionalmente a los componentes expuestos anteriormente, una implementación de SAP se compone por algunas instancias adicionales que proporcionan servicios compartidos a otras instancias del "landscape", y que son utilizados para actividades de administración y operación de la solución. Entre algunos de los principales componentes se encuentran los siguientes [SCHT2011]:

- SLD o system landscape directory
- SAP Solution Manager
- Solution Manager diagnostics para los análisis de causa raíz
- Alert monitor para el envío de notificaciones en el esquema de monitoreo
- SAP central process scheduling para la administración de procesos de fondo

En resumen, es necesario tomar en cuenta los siguientes puntos como base para realizar un diseño completo y robusto de la solución, que cumpla con todos los requerimientos técnicos planteados por el negocio:

1. Determinar los procesos de negocios que serán implementados.
2. Determinar cuáles unidades de software será necesario instalar en atención a los procesos mapeados.
3. Determinar los servicios compartidos requeridos para atender la operación y administración del "landscape" que será implementado.
4. Determinar cuántos sistemas serán requeridos y cuál será la función de cada uno de ellos.
5. En conjunto con los proveedores de hardware y de consultoría técnica, mapear los servidores y características físicas que será necesario adquirir para realizar la instalación de las instancias SAP, para este punto en particular se recomienda incluir un proyecto de dimensionamiento en el que en base a la carga esperada, se estime la capacidad de hardware que será necesario adquirir para soportar la operación de los ambientes.
6. Implementar el "landscape" de SAP Netweaver.

La experiencia nos ha demostrado que existe una multitud de factores que se deberán tomar en cuenta al momento de realizar el diseño del "landscape" de SAP, por lo cual esta tarea se deberá realizar en conjunto con el equipo de tecnología del cliente, los proveedores de consultoría externos, así como recursos asignados por parte del mismo fabricante SAP, con el objetivo de garantizar una implementación exitosa.

#### **4.9.1. Sistemas de transportes**

Con el objetivo de mantener una gestión, administración y control adecuado referente a los cambios y modificaciones que es necesario realizar a un aplicativo durante la implementación y operación diaria, SAP desarrollo la arquitectura conocida como sistema de transportes o "transport management system" por sus siglas en inglés [SCHT2011].

El sistema de transporte se accede vía la transacción STMS en una plataforma ABAP, en la figura 4.9.1-1 se muestra un ejemplo de implementación que incluye la conexión de una instancia de desarrollo, calidad y producción. El objetivo principal del TMS es controlar la propagación de cambios a través de todo el "landscape" a través de rutas predefinidas, por medio del cual se realiza el transporte tanto de los objetos que están siendo modificados desde el sistema origen hasta el sistema destino importando los objetos y cambios realizados, así como todo tipo de configuración manual que sea necesario realizar en SAP.



Fig. 4.9.1-1 Sistema de transportes.

Como en cualquier implementación de algún ERP, SAP recomienda realizar la instalación de varias instancias de una misma solución, con el objetivo de asignar funcionalidades puntuales a cada una de ellas, con esto se busca garantizar que ya sea la generación de nuevos desarrollos o nueva funcionalidad que sea necesario implementar, o por otra parte modificaciones a nivel técnico o funcional, sean correctamente probadas y validadas en un ambiente no productivo, antes de implementarlas en producción, con el objetivo de evitar posibles fallas que pudieran impactar la operación y estabilidad del sistema.

De esta manera, se recomienda contar con la instalación de la siguiente infraestructura para cada solución que será implementada [SCHT2011]:

1. Un ambiente conocido como “desarrollo” enfocado principalmente a las actividades de configuración manual, así como a las actividades de desarrollo de software, este ambiente deberá representar además al ambiente productivo, contando con la instalación de todos los componentes de software que se tienen en el ambiente productivo, así como toda la configuración asociada, “desarrollo” será el sistema sobre el cual se ejecutarán las pruebas unitarias iniciales.

El ambiente de desarrollo será utilizado también, como principal punto de análisis para la resolución y mantenimiento de los problemas que se vayan identificando en producción, a lo largo del ciclo de vida del sistema.

2. Se requerirá adicionalmente de un ambiente conocido como “calidad”, el cual estará enfocado a las siguientes tareas:
  - Ejecución de pruebas de configuración realizadas y transportas desde el ambiente de desarrollo.
  - Ejecución de pruebas de calidad de una manera integral para todos los desarrollos que se transportarán hacia el ambiente productivo.

Este sistema deberá ser de la misma manera una imagen del ambiente productivo, pero adicionalmente deberá contar con una copia de los datos que se tienen en el ambiente productivo para que la ejecución de pruebas que se requieran realizar, antes de aprobar algún desarrollo o configuración hacia productivo, se realicen de manera integral y satisfactoria.

En este sistema no se deberá realizar ningún desarrollo o cambio manual que no se tenga en el sistema de desarrollo, sino que se enfocará únicamente a la ejecución de pruebas de calidad, así como la replicación y rastreo de problemas identificados en el ambiente productivo.

3. Se deberá contar con un sistema productivo, estable y con una infraestructura robusta, que le permita soportar la operación productiva del negocio de una manera eficiente. Esta instancia estará enfocada a tareas meramente operativas y no se deberá realizar ninguna actividad asociada al desarrollo o depuración de programas.

En la figura 4.9.1-2 se muestra la arquitectura descrita que ilustra la recomendación de contar con un ambiente de desarrollo, otro de calidad así como una instancia productiva para cada tipo de solución SAP.

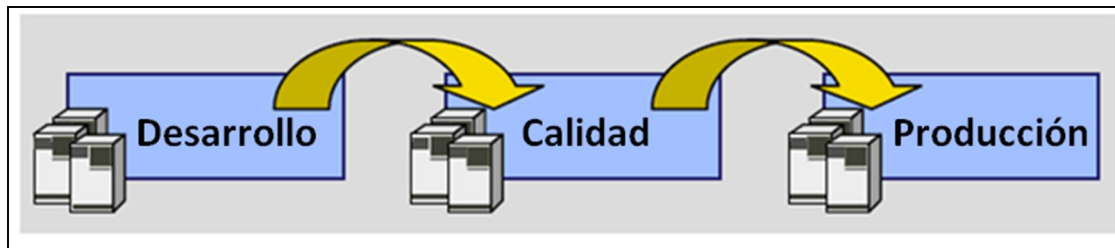


Fig. 4.9.1-2 Infraestructura de operación [SCHT2011].

De esta manera, se contará con un “landscape” compuesto por las siguientes tres instancias, con el objetivo de garantizar una administración de los cambios requeridos eficiente, durante el ciclo de vida de la solución:

- Sistema de desarrollo (DEV)
- Sistema de aseguramiento de la calidad (QAS)
- Sistema productivo (PRD)

En proyectos con empresas que cuentan con recursos económicos mayores destinados a la implementación de SAP, se suele incluir dos instancias adicionales conocidas como “sandbox” o caja de arena y un ambiente “pre-productivo”, en este caso el sistema de “sandbox” se enfocará a pruebas referentes a la instalación de paquetes de soporte, proyectos de actualización de versión entre otros, por su parte el ambiente “pre-productivo” se utilizará principalmente para las pruebas de volumen y de estrés sobre el sistema. En la siguiente figura se ilustra esta arquitectura en la cual se incluye un ambiente de “sandbox” o pruebas originales que no es incluido en el sistema de transportes.

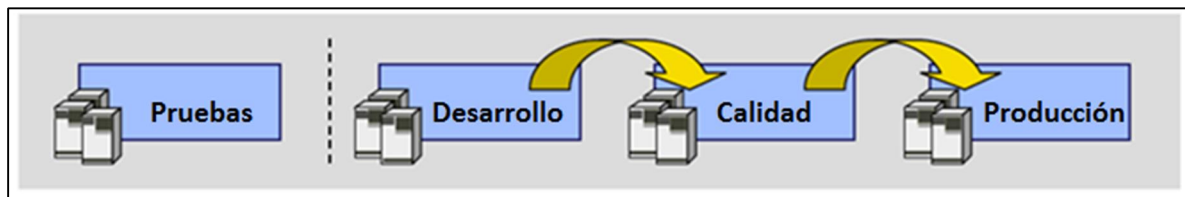


Fig. 4.9.1-3 Infraestructura de operación y pruebas [SCHT2011].

El cliente tiene la posibilidad de elegir el “landscape” que se acomode más a sus necesidades, tomando en cuenta el hecho de que el anexar instancias adicionales incrementará los costos del proyecto por la inversión en recursos destinados, al pago de licencias, operación, soporte y administración de la infraestructura asociada, entre otros.

Por tanto la recomendación es contar con tres instancias o al menos dos de ellas para mantener un control adecuado de todos los cambios que se requiera realizar en el “landscape” de SAP.

Como se observa en la figura 4.9.1-4, el TMS permite realizar esta administración de cambios centralizada, por medio de la generación de dominios de transporte, definición de rutas y asignación de un controlador del dominio de transportes.

El dominio de transportes es necesario crearlo para bajo este objeto, definir la inclusión de todos los sistemas que pertenecerán a dicho dominio y que accederán una ruta común entre ellos.

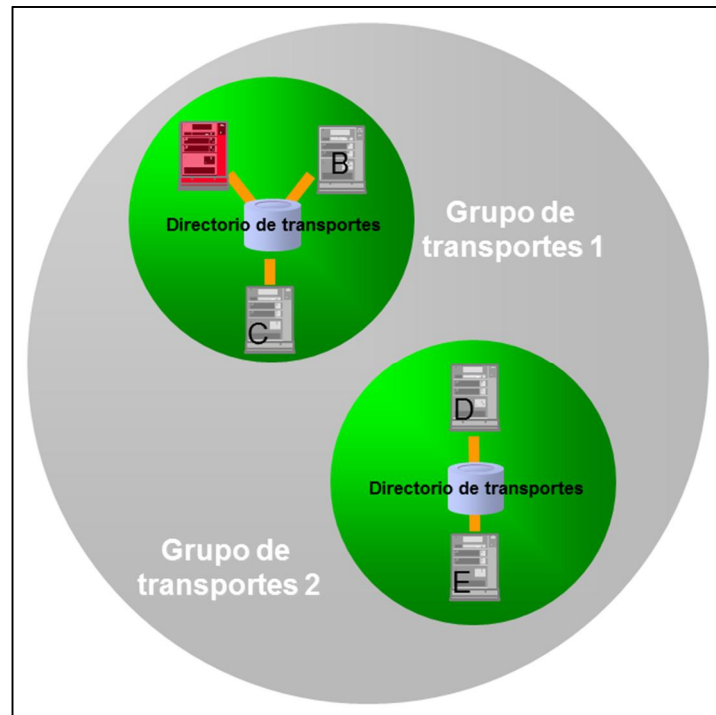


Fig. 4.9.1-4 Administración de cambios centralizada.

Por su parte la ruta de transporte establecerá de qué manera se comunicarán los sistemas entre sí, así como cuál será el flujo que seguirá la importación de los cambios dentro del “landscape”.

El controlador de dominio, será el sistema central desde el cual se generará la configuración central del TMS y se establecerá los mecanismos de autorización requeridos para la incorporación o eliminación de instancias en el dominio de transporte.

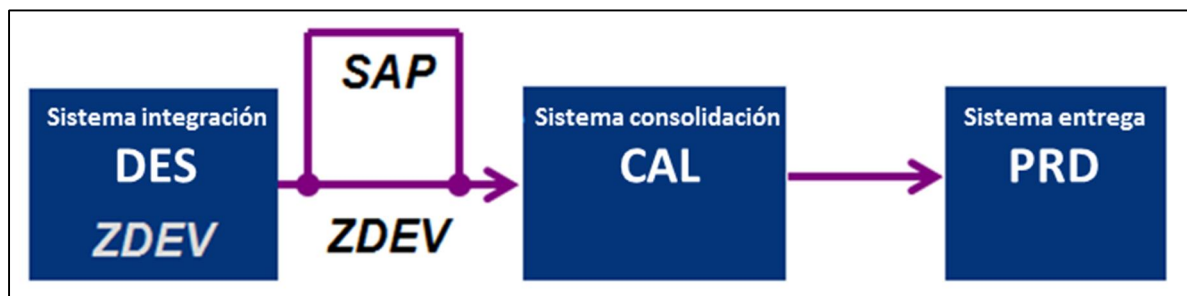


Fig. 4.9.1-5 Ruta de consolidación.

En una configuración estándar de tres instancias para una solución como la mostrada en la figura 4.9.1-5, se recomienda realizar la definición de una “ruta de consolidación” entre los ambientes de desarrollo (DEV) y calidad (QAS), por medio del cual se conecten estos sistemas y se permita la exportación de objetos y cambios desde desarrollo hacia la instancia de calidad.

De la misma manera se requerirá realizar la configuración de una “ruta de entrega”, enfocada a exportar los objetos desde el sistema de calidad una vez que han sido debidamente validados y probados exitosamente hacia el ambiente productivo en donde deberán ser ya importados para entrar en la operación diaria.

#### 4.9.2. SLD

El SLD o “system landscape directory” es un componente desarrollado por SAP que se utiliza para la obtención de información de versiones de componentes de todas las instancias que se encuentran instaladas en un “landscape” bajo la plataforma de Netweaver [AS2004].

Esta herramienta es un punto de conexión central entre todas las instancias, el cual entre otras cosas permite visualizar a nivel de detalle los componentes de software que están siendo utilizados, para apoyar en la administración del control de versiones, así como también es una herramienta que sirve de punto de comunicación entre algunas de las diversas soluciones SAP para el intercambio de información. En la figura 4.9.2-1 se muestra un diagrama que describe la arquitectura de operación de esta herramienta.

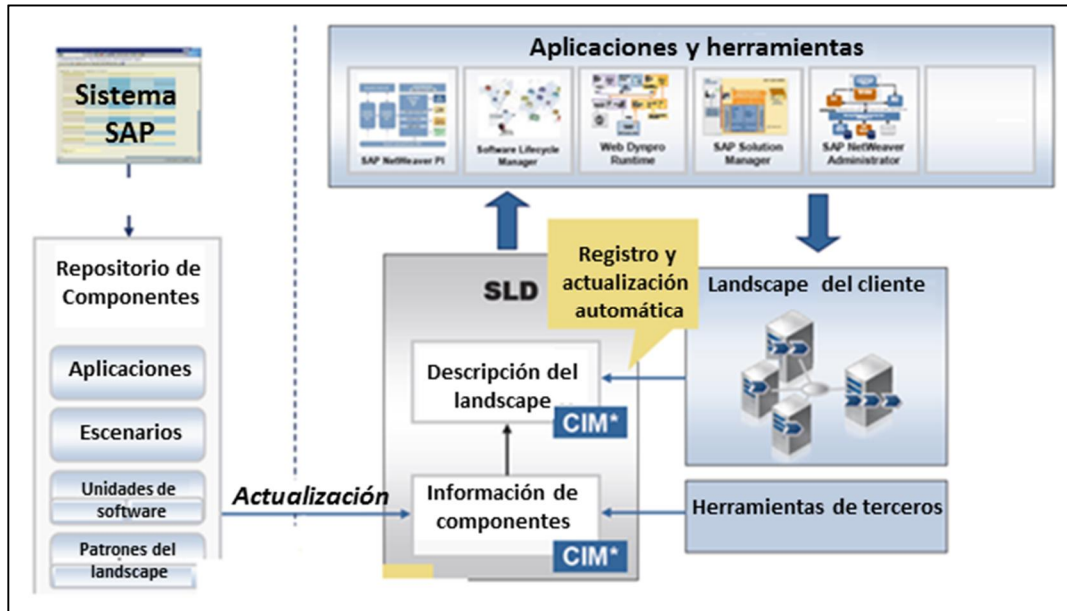


Fig. 4.9.2-1 SLD [AS2004].

El SLD se instala bajo una plataforma con java y permite almacenar la información referente a los componentes así como también la descripción del "landscape" para cada una de las instancias ya sean SAP o no-SAP como se muestra en la figura 4.24.

La obtención de esta información se realiza a partir de colectores de datos que deberán ser instalados en el sistema, así como la configuración correspondiente a las RFC que enviarán la información desde el sistema fuente hacia el SLD de manera periódica y automática.

Esta información es actualizada de manera automática, permitiendo de esta manera contar con un repositorio disponible en todo momento, en el cual se podrá visualizar a nivel de detalle la instancia, sus componentes, el nivel de versión así como el número de parche que se encuentra instalado como se muestra en la figura 4.9.2-2, esto suele ser muy útil en las etapas en las cuales inician los procesos de actualización de versiones y es necesario analizar la infraestructura completa actual, así como las posibles dependencias que se deberán analizar antes de iniciar con la ejecución de actividades de actualización, con el objeto de evitar posibles impactos que puedan repercutir en la operación del negocio.

La imagen muestra la interfaz del System Landscape Directory (SLD) con la siguiente información:

Name	Host	Version	Last Update
AB0 on db_host_00	db_host_00	48C	07/28/2005 15:45
AB1 on db_host_1	db_host_1	48D	07/29/2005 15:48
BWS on localhost	localhost	48D	08/08/2005 15:17
BWE on bwp_host	bwp_host	48D	09/05/2005 13:59
BWP on bwp_host	bwp_host	48D	08/23/2005 13:49

Fig. 4.9.2-2 Componentes del SLD.

## 5. Administración del sistema

Este capítulo aborda las tareas básicas en la administración de un sistema SAP ERP. Desde detener e iniciar el sistema de manera adecuada, revisar los registros o bitácoras al detener e iniciar el sistema para asegurarnos que ambos procesos ocurrieron sin errores; hasta el mantenimiento de versiones del sistema, aplicando correcciones o introduciendo nuevas funcionalidades al instalar los llamados "Support Packages" o parches de soporte. En este capítulo también revisamos la configuración del sistema, mediante la utilización de los diferentes perfiles: default, inicio y de instancia. Los cuales nos sirven entre otras cosas para configurar el número de procesos de trabajo y los tipos de procesos de trabajo que deseamos en nuestro sistema. Así mismo, se da a conocer el término mandante y la administración de los mismos dentro de nuestro servidor de aplicaciones SAP.

### 5.1. Iniciar y detener el sistema SAP ERP de forma adecuada

#### 5.1.1. Proceso de inicio de un sistema SAP ERP

El arranque del sistema SAP es un requisito básico para poder trabajar con el sistema. El reinicio de los sistemas SAP también es necesario después de algún mantenimiento en el hardware sobre el cual corre el sistema o después de modificar algún parámetro en los perfiles del sistema.

El inicio de un sistema SAP consta de varios pasos, los cuales son ejecutados por el usuario de sistema operativo "<sid>adm". A continuación se describen los pasos del arranque del sistema y la figura 5.1.1-1 ilustra dicho proceso.

Inicio de la base de datos:

- Un elemento fundamental para el sistema SAP es la base de datos. Por lo tanto, la base de datos es siempre el primer elemento en ser iniciado.

Inicio de la instancia central:

- El inicio de la instancia central comienza con el servidor de mensajes, luego el despachador y una vez que éste está funcionando, los procesos de trabajo son iniciados.

Inicio de las instancias de diálogo:

- Si existen instancias de diálogo, sólo se pueden iniciar, una vez que el servidor de mensajes y el servidor de encolamiento están activos. Al iniciarlas, arranca primero el despachador y luego los procesos de trabajo.

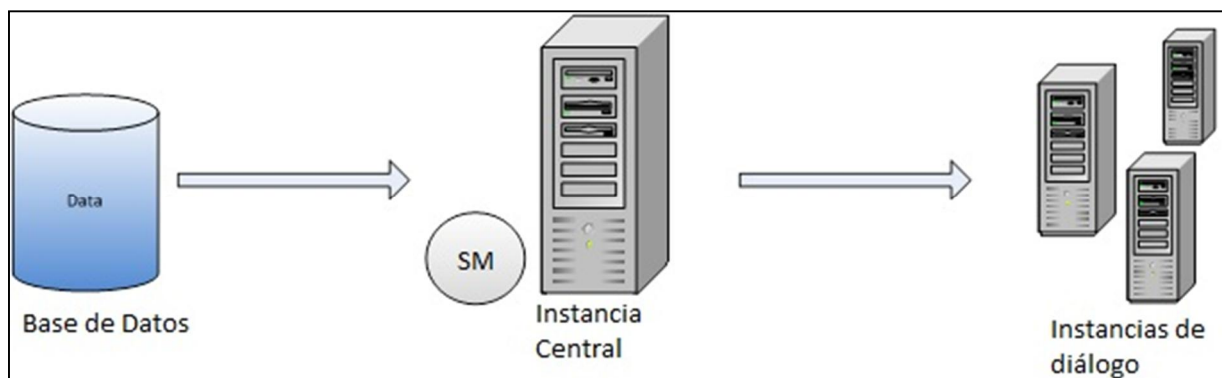


Fig. 5.1.1-1 Proceso de inicio de un sistema SAP

En sistemas operativos Windows, los sistemas SAP se pueden iniciar mediante la consola de administración Microsoft (o MMC, por sus siglas en inglés).

En esta consola se puede observar el estado de los elementos de la instancia como el servidor de mensajes, el despachador o de la instancia en general. El estado se indica, a través del siguiente código de colores:

- **Gris:** Detenido
- **Amarillo:** Está iniciando
- **Verde:** Activo

➤ **Rojo:** Detenido por errores

Las especificaciones de las instancias, tales como, el tipo y el número de procesos de trabajo, el tamaño de las áreas de memoria, y otras opciones son controladas en el arranque de las instancias por medio de perfiles.

Cuando las instancias son iniciadas, el perfil de inicio es leído para determinar qué procesos deben ser iniciados.

La configuración general de las instancias está guardada en el perfil default, mientras que el perfil de instancia, indica la configuración específica para cada una de ellas. Estos perfiles son leídos por el despachador quien se encarga de iniciar los procesos de trabajo y de crear la configuración de cada instancia.

Una vez que la instancia ha iniciado de manera exitosa, todos los procesos de trabajo se conectan a la base de datos.

### 5.1.2. Inicio del sistema: Bitácoras

El arranque de un sistema SAP es una fase especialmente importante, este proceso es registrado por el sistema operativo, el sistema SAP y la base de datos. Si un sistema SAP no inicia, usted puede encontrar el error en las bitácoras. El problema pudo haber ocurrido al iniciar la base de datos, y por lo tanto el sistema SAP ya no arrancó.

Las bitácoras correspondientes al arranque de un sistema SAP se almacenan en el sistema de archivos. Si ocurren problemas durante el inicio, estas bitácoras pueden proveer información útil como errores o descripción del problema. Estos archivos se almacenan en el directorio base de la instancia correspondiente. En la figura 5.1.2-1 se muestran los nombres de algunas bitácoras que registran los eventos del inicio del sistema.

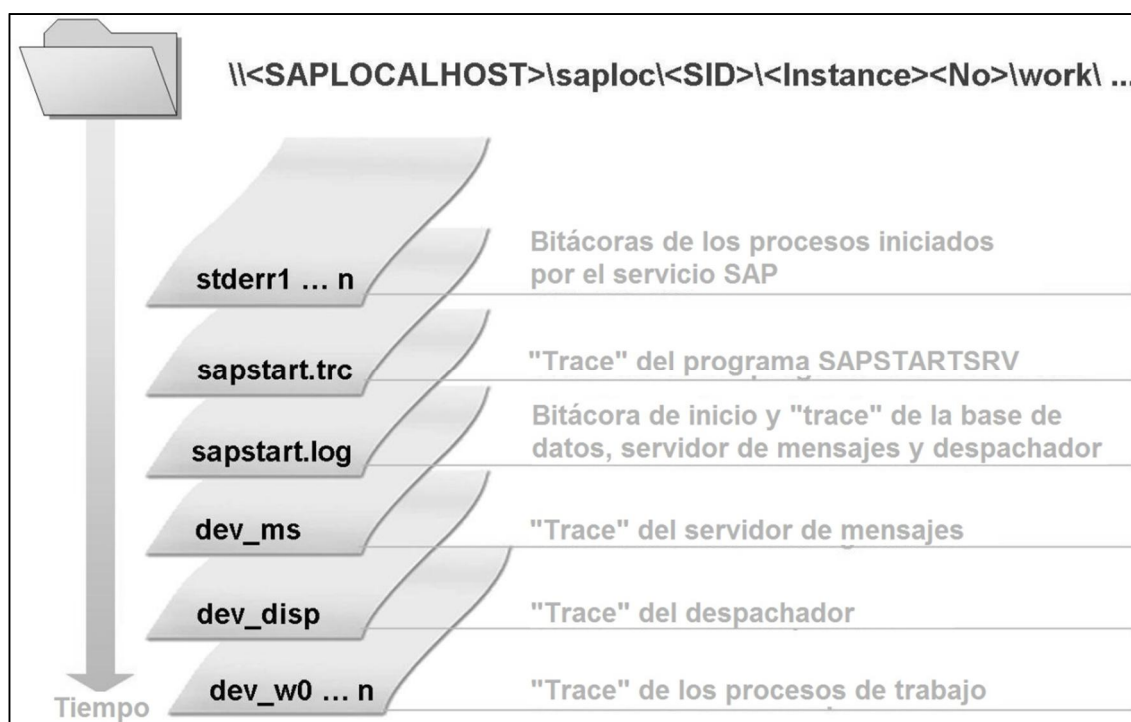


Fig. 5.1.2-1 Registro del proceso de inicio de un sistema SAP [TADM10\_12005]

Sin duda, es importante conocer la ubicación de estas bitácoras, porque la información que contienen es de gran ayuda cuando hay algún problema al iniciar los sistemas SAP. Hemos notado que es una buena práctica comenzar el análisis utilizando los "traces" de los procesos de trabajo. Sin embargo, si el problema ocurrió en una fase temprana del inicio del sistema, recomendamos revisar las bitácoras stderr.

### 5.1.3. Proceso de apagado de un sistema SAP

Detener un sistema SAP puede ser necesario por diversas razones, por ejemplo, reiniciar el sistema después de haber hecho cambios en los parámetros de la instancia, antes de instalar un parche al kernel o para llevar a cabo mejoras en el hardware sobre el cual corre el sistema.

Antes de detener el sistema, recomendamos llevar a cabo las siguientes tareas:

- Revisar si hay usuarios en el sistema.
- Revisar si hay procesos de fondo activos. Los procesos de fondo que serán detenidos al detener el sistema necesitan recalendarizarse. Aquellos procesos de fondo que estaban calendarizados mientras el sistema está detenido, correrán automáticamente al iniciar el sistema.
- Revisar si hay procesos de actualización que serán terminados al detener el sistema. A estos procesos de actualización se les hace “roll back” y se les asigna el estado de “init”. En el momento de reinicio de la instancia son aplicados.

Antes de detener el sistema, es recomendable mandar un mensaje a través del sistema a los usuarios para que puedan guardar los cambios en los que estaban trabajando.

El proceso de apagado de un sistema SAP se lleva a cabo en el orden contrario al de inicio. Primero se detienen las instancias de diálogo y posteriormente la instancia central.

Durante el proceso, los procesos de trabajo son los primeros en detenerse, luego el despachador y por último el servidor de mensajes. Si es necesario, también se puede detener la base de datos.

Antes de detener la base de datos, también recomendamos que se cercioren que no hay ningún respaldo en progreso. El detener la base de datos mientras hay un respaldo corriendo puede ocasionarnos problemas al momento de iniciarla nuevamente y el respaldo, de cualquier forma, no será útil. Por lo que sugerimos cancelar el respaldo antes de detener la base de datos.

## 5.2. Configuración del sistema

### 5.2.1. Configuración de los parámetros del sistema

La configuración de las instancias de un sistema SAP, se lleva a cabo mediante los parámetros del sistema. Los valores por omisión para estos parámetros están definidos en el código de programa del kernel.

Los valores por omisión pueden ser modificados usando los archivos de los perfiles, los cuales son leídos cuando una instancia es iniciada. Como los perfiles sólo son leídos al momento que arranca el sistema, es necesario reiniciar la instancia después de haber modificado los parámetros.

Los cambios de forma dinámica, es decir, cuando el sistema está corriendo, sólo es posible para un pequeño número de parámetros. El orden en que los parámetros del sistema son asignados se aprecia en la figura 5.2.1-1.



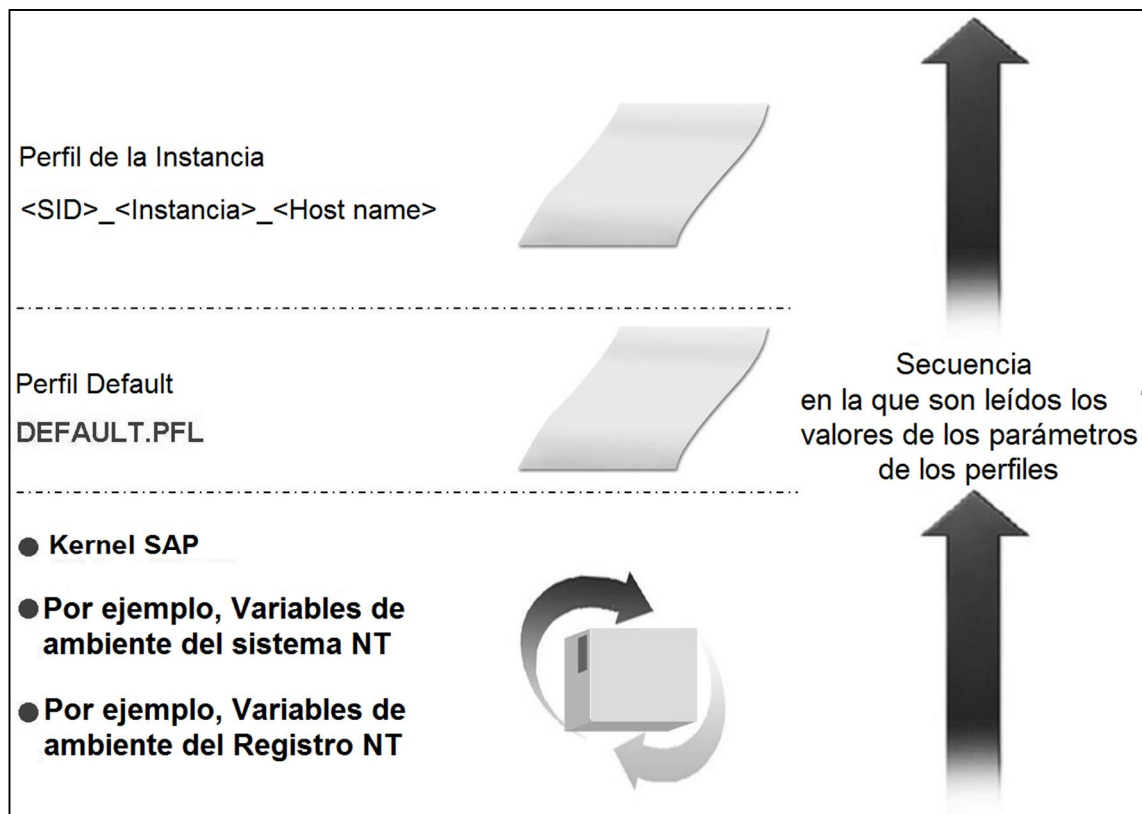


Fig. 5.2.1-1 Asignación de los parámetros del sistema [TADM10\_12005]

Los archivos de los perfiles son creados automáticamente al momento de la instalación. Una vez que la instalación es finalizada, los perfiles se almacenan a nivel sistema operativo en el directorio: \usr\sap\<SID>\SYS\profile.

En principio, estos archivos pueden ser modificados utilizando herramientas del sistema operativo (editores de texto, tales como vi o el bloc de notas). Si se hace de esta manera, el usuario debe tener cuidado de introducir los nombres de los parámetros sin errores. Parámetros configurados de manera errónea pueden ocasionar que el sistema no levante. Además, una vez hechos los cambios, se deben de importar los perfiles a la base de datos mediante el menú "Utilidades -> Importar perfiles -> De servidores activos". En muchas ocasiones, nos ha tocado que los perfiles han sido modificados mediante editores de texto y se ingresan parámetros duplicados con diferente valor y también hemos tenido experiencias, en las que no los perfiles no son importados en la base de datos. Para evitar problemas, es más conveniente y seguro cambiar los parámetros de los perfiles utilizando las herramientas de SAP (transacción RZ10).

El sistema SAP tiene tres perfiles de sistema:

- Perfil de inicio
- Perfil default
- Perfil de instancia

En el perfil de inicio específico de cada instancia (START<instancia><número instancia>\_<Host name>), se indica qué procesos serán iniciados en cada instancia. Como por ejemplo, el servidor de mensajes y el despachador.

Sólo existe un perfil default (DEFAULT.PFL) por cada sistema SAP, y éste puede ser leído por todas las instancias que forman ese sistema. Este perfil incluye configuración que afecta a todas las instancias, tales como el nombre del sistema, el nombre del servidor de la base de datos, el nombre del servidor de encolamiento, o incluso el mandante por omisión.

El perfil de la instancia (<SID>\_<Instancia><número instancia>\_<Host name>) define los parámetros que aplicarán sólo para esta instancia, tales como el número y tipo de procesos de trabajo, la configuración y tamaño de la memoria que usará la instancia. Este perfil es específico de cada instancia.

### 5.2.2. Ajustar los parámetros del sistema

Si es necesario hacer cambios a los parámetros, los cambios pueden realizarse usando editores del sistema operativo. Sin embargo, este procedimiento presenta ciertos riesgos. Por lo tanto, un sistema SAP cuenta con herramientas para administrar y darle mantenimiento a los perfiles.

Ventajas de la administración de perfiles desde el sistema SAP:

- Administración y mantenimiento centralizados de todas las instancias
- Los cambios en los perfiles son revisados consistentemente
- Administración de varias versiones de un mismo perfil
- Comparación entre el perfil que está actualmente activo con aquel que se encuentra guardado en la base de datos
- Activación inmediata de los parámetros seleccionados

Después de la instalación del sistema, los perfiles sólo existen a nivel sistema operativo. Para hacer uso de la administración de perfiles del sistema SAP, los perfiles deben ser importados dentro de la base de datos. Una vez importados, los cambios a los perfiles pueden ser realizados desde el sistema SAP, teniendo las ventajas mencionadas anteriormente. Los cambios son guardados en la base de datos y escritos a nivel archivo. Los cambios únicamente toman efecto cuando son leídos por el sistema, esto es, cuando se reinicia el sistema.

### 5.3. Mantenimiento de versiones

Existen dos formas de dar mantenimiento a un sistema SAP:

- Al aplicar notas de SAP con la transacción SNOTE.
- Al aplicar “support packages” de SAP con la transacción SPAM.

Al implementar una nota de SAP un error en un sólo programa o transacción es removido. Cuando se aplica un “support package”, todos los errores conocidos al momento de crear el “support package” son corregidos. Además, las notas y los “support packages” no son sólo correcciones de errores, sino que también pueden traer nuevas funciones al sistema.

#### 5.3.1. Importar “support packages”

Un “support package” consta de una cantidad de objetos corregidos o de nuevos objetos. Los “support packages” son necesarios para corregir errores en varios componentes. Esto se logra al remplazar los objetos erróneos con versiones corregidas de dichos objetos.

Además de los “support packages” existen otros elementos que pueden dar mayor funcionalidad al sistema: soluciones industriales y “Plug-Ins”. Ambos, son conocidos como “add-ons”.

Las soluciones industriales proveen al sistema de funcionalidades específicas de ciertas industrias, como la de calzado, aeronáutica, farmacéutica, etc. Los “plug-ins” añaden las interfaces necesarias para habilitar la comunicación entre diferentes sistemas.

La transacción SPAM permite importar los “support packages”. Esto aplica para los “support packages” del sistema estándar y para todos los “add-ons”.

También es importante recordar que la actualización para la SPAM/SAINT sólo se puede realizar utilizando la transacción SPAM.

La transacción SAINT permite instalar o subir de versión un “add-on”.

Al trabajar para diferentes clientes, hemos tenido la oportunidad de importar “support packages” en diversas ocasiones. De esa práctica, hemos aprendido que los siguientes requisitos y ajustes se deben tomar en cuenta al momento de importar “support packages” con la transacción SPAM:

- Los “support packages” requeridos deben estar disponibles.
- Debe de haber suficiente espacio en disco.
- La importación debe realizarse durante lapso de baja operación en el sistema.
- La actualización más reciente de la SPAM/SAINT es requerida.

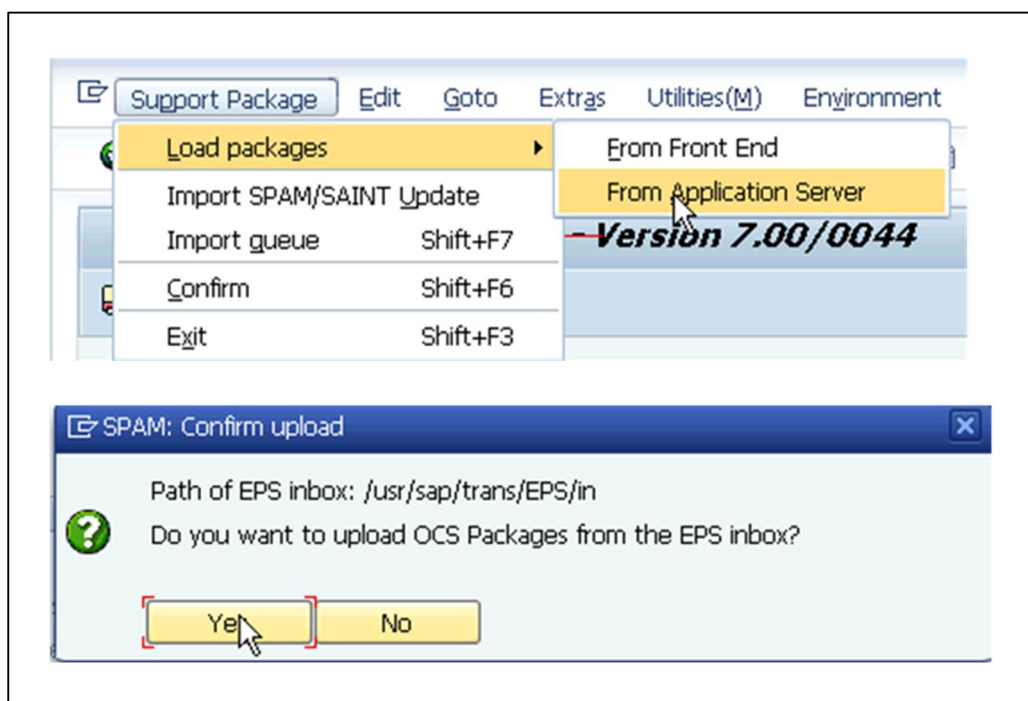
- El sistema de transportes debe estar configurado.
- No existen “support packages” abortados en el sistema.
- La importación se debe realizar desde el mandante 000.
- Hacer un respaldo del sistema antes de iniciar la actividad.
- Revisar la nota correspondiente a la versión del sistema de problemas conocidos.

El primer paso para aplicar “support packages” a un sistema es cargarlos en el mismo. Los “support packages” se pueden conseguir a través del “Marketplace” o a través de discos ópticos que se piden en el mismo portal de soporte.

Para cargar los “support packages” en el sistema se llevan a cabo los siguientes pasos:

1. Descargar los “support packages” desde el “Marketplace” o copiarlos desde el CD y descomprimirlos en el servidor de aplicaciones SAP.
2. Cargarlos en el sistema por medio del menú “Support Package -> Cargar Paquetes -> Desde servidor de aplicaciones”

Este proceso se ilustra a continuación:



**Fig. 5.3.1-1 SPAM: Cargar “support packages”**

Si los archivos comprimidos están en la computadora de escritorio y son más pequeños que alrededor de 10 MB, los archivos se pueden transferir al servidor de aplicaciones y descomprimirlos directamente desde el administrador de “Support Packages” (SPAM) con los siguientes pasos:

1. Llamar a la transacción SPAM.
2. Seleccionar “Support Package -> Cargar Paquetes -> Desde front-end”
3. Para descomprimirlos, se escoge “Descomprimirlos”.

A pesar de que existe esta opción, no es usada frecuentemente en la práctica, ya que, la mayoría de las veces, son bastantes “support packages” los que se importan y este método es más lento, que el mencionado previamente.

Una vez que se han cargado los “support packages”, el siguiente paso es crear la cola de aquellos que se importarán en el sistema. Para ello, se seleccionan los componentes a los cuales se les quieren aplicar las nuevas versiones.

El administrador de “Support Packages” (SPAM) se asegura que sólo los “support packages” que coinciden con el sistema sean desplegados en la cola.

Las siguientes reglas aplican cuando se crea la cola:

- Los “support packages” de cada componente seleccionado son colocados de acuerdo a su secuencia.
- Si algún “support package” en la cola está relacionado con “support packages” de otro componente, los “support packages” extra son posicionados en la cola hasta que todas las dependencias queden cubiertas.

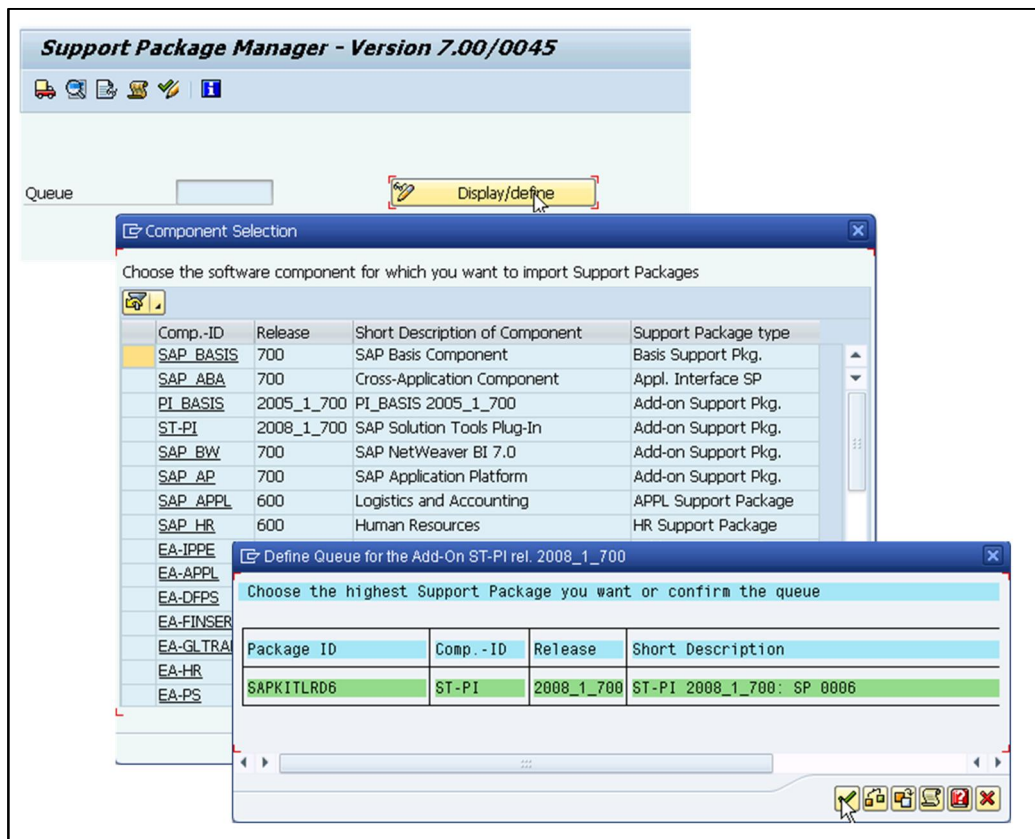


Fig. 5.3.1-2 SPAM: definir la cola

Cuando se ha definido la cola de “support packages” (Fig. 5.3.1-2) a ser importada, el sistema está listo para iniciar el proceso de importación. Para comenzar se selecciona “Support Package -> Importar Cola”. La duración de este proceso depende del número de “support packages” que se vayan a importar.

Después de que la cola ha sido importada de forma satisfactoria, es recomendable revisar las bitácoras de dicho proceso para asegurarse de que todo el proceso se ha llevado a cabo sin errores.

El último paso de este proceso consiste en confirmar la importación exitosa de la cola.

### 5.3.2. Actualizar las herramientas

Las transacciones SPAM y SAINT son las herramientas que nos permiten dar mantenimiento al sistema. La transacción SPAM nos ayuda a importar “support packages”. Esto aplica para los “support packages” del sistema estándar y para todos los “Add-Ons”. La transacción SAINT permite instalar y subir de versión un Add-On.

SAP recomienda que siempre se importe la versión más reciente de la herramienta SPAM/SAINT antes de importar otros “support packages”. Sólo es posible importar una actualización de SPAM/SAINT con la transacción SPAM. Para importar una actualización de SPAM/SAINT, es necesario que no haya una implementación de “support packages” abortada en el sistema. Si ésta existe, una caja de diálogo informa de esto. Se tienen entonces dos opciones:

- Se importa por completo la cola de “support packages” que está pendiente y luego se importa la actualización de SPAM/SAINT.
- Se resetea el estado de la cola, se importa primero la actualización de SPAM/SAINT y luego la cola que estaba pendiente. Se puede resetear el estado de la cola utilizando el menú “Extras -> Resetear Estado -> Cola”.

Para importar la actualización más reciente de SPAM/SAINT, se escoge “Support Package -> Importar actualización SPAM/SAINT”. Las actualizaciones de SPAM/SAINT son confirmadas automáticamente después de que la importación ha sido satisfactoria [TADM10\_12005].

### 5.3.3. Importar notas de SAP

Las notas de SAP proveen información de cualquier tipo para clientes y socios de negocios de SAP. Pero también, frecuentemente, las notas de SAP contienen soluciones para problemas conocidos con un sistema SAP.

En esta sección cuando nos referimos a una nota, hablamos de aquellas que dan solución a un error contenido en el código fuente de un programa ABAP. Estas notas describen como quitar esos errores en el sistema SAP al proveer el código fuente correcto. Éstas incluyen una descripción de los siguientes elementos [TADM10\_12005]:

- El síntoma del error
- La causa del error
- Descripción de cómo corregir el código fuente
- Versión del sistema y estado del “support package” para la cual es válida la nota
- Ligas al “support package” que corrigen el problema

Aplicar este tipo de notas en el sistema es una actividad cotidiana para un administrador de sistemas SAP. Aprovechando una de las tantas ocasiones en las que hemos realizado esta actividad, nos dimos a la tarea de tomar pantallas de cada una de las etapas y las usamos en esta sección para ilustrar nuestros comentarios.

Antes de implementar una nota, es necesario leerla cuidadosamente. La nota puede contener prerequisites, interacciones, y referencias a actividades después de implementarla, tal como hacer un cambio a una tabla.

Los sistemas SAP cuentan con una herramienta llamada asistente de notas SAP. Esta herramienta se inicia utilizando la transacción SNOTE.

La pantalla principal se conoce como área de trabajo. Todos los usuarios ven las notas inconsistentes y nuevas en su área de trabajo. Pero sólo cada usuario ve las notas que están la categoría “En Proceso”. Estas notas son las que procesa cada usuario. Desde esta pantalla se pueden realizar todas las acciones, como bajar, implementar y mostrar las notas. La figura 5.3.3-1 muestra esta pantalla.

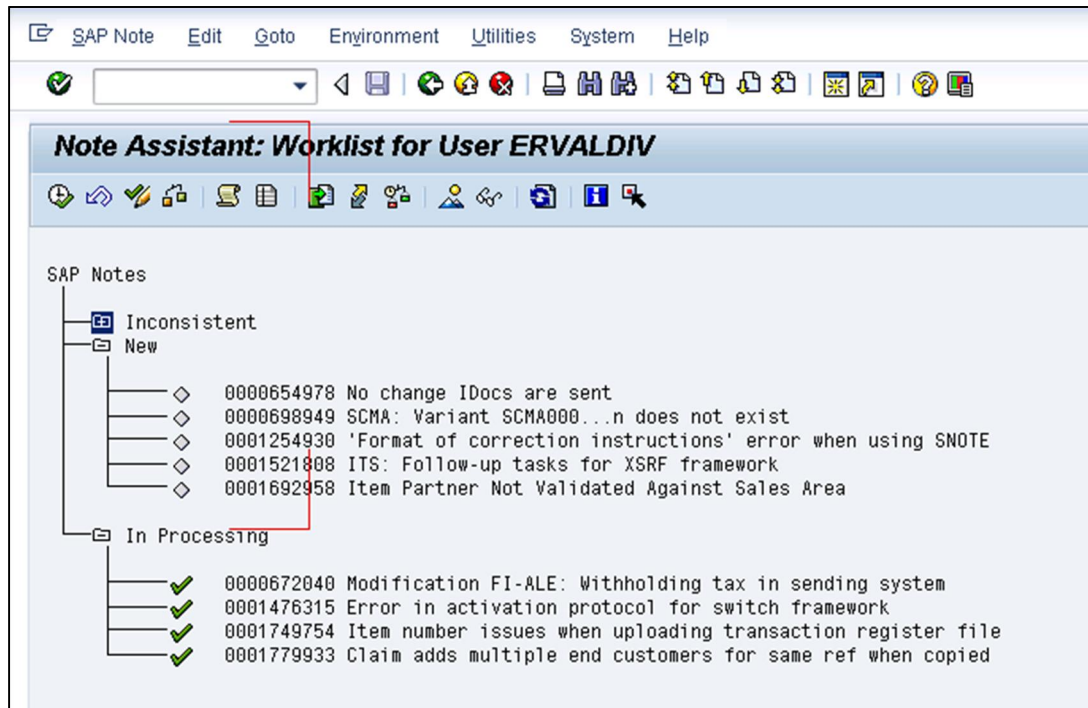


Fig. 5.3.3-1 Área de trabajo

Para corregir un error usando una nota, se debe conocer el número de la nota que corrige el error.

Para bajar una nota directamente con el asistente de notas, se necesita una conexión a SAPNet – R/3. Se selecciona descargar nota de SAP y se ingresa el número de la nota. En la figura 5.3.3-2 se aprecia este proceso.

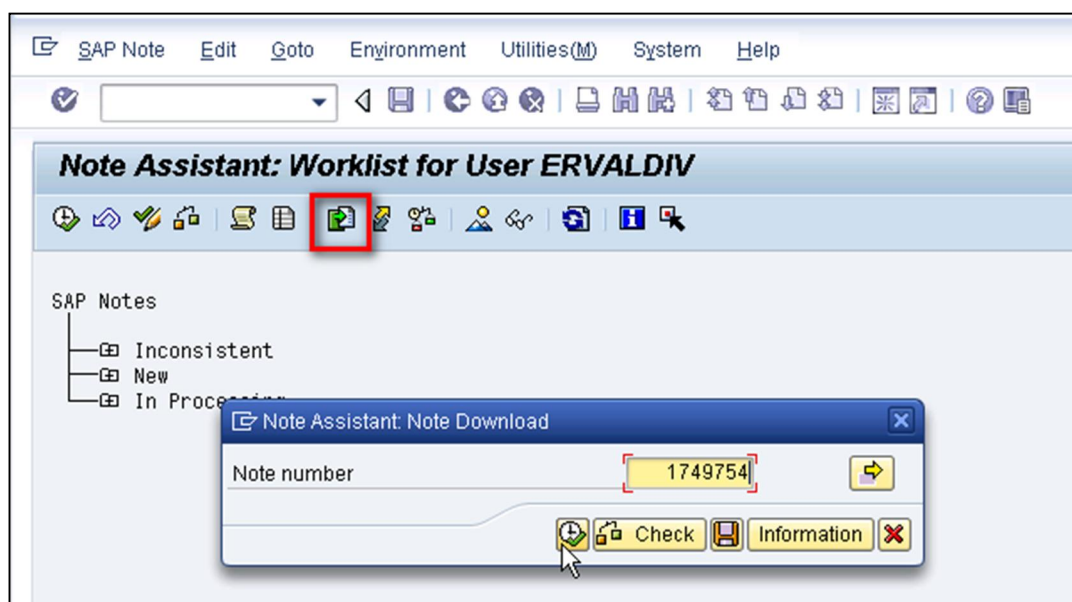


Fig. 5.3.3-2 Descargar desde SAP R/3

El asistente de notas se puede utilizar también para clasificar una nota o para cambiar el estado en el que se encuentra.

Si se desea que cierto usuario en particular se haga cargo de una nota, se ingresa el nombre de usuario de esa persona y se pone “En proceso” (figura 5.3.3-3). La nota aparecerá en su área de trabajo.

Si la nota no corresponde a la situación actual del sistema, se puede poner como “No relevante”. Después de implementar una nota, se puede clasificar como “Completada”.

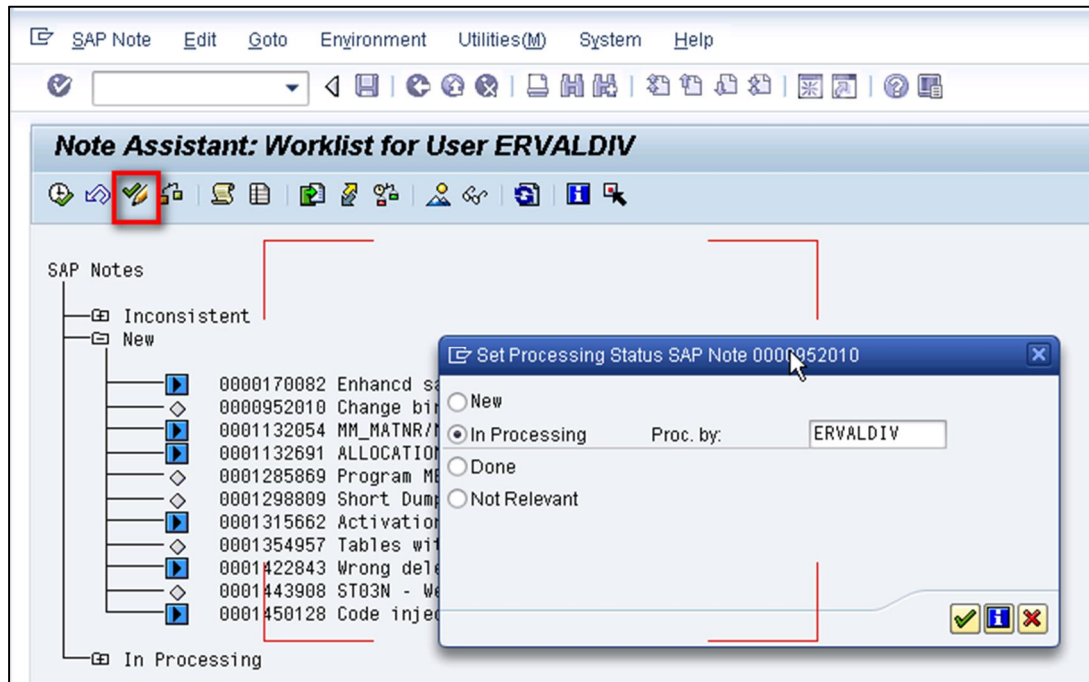


Fig. 5.3.3-3 Asignando el estado de proceso de una nota

Para implementar una nota, se selecciona la nota respectiva y se selecciona la opción Implementar nota.

Salta una nueva ventana en la que se debe confirmar que la nota ha sido leída porque ésta puede contener instrucciones de actividades que se deben realizar ya sea, antes o después de implementarla.

Al confirmar que se ha leído la nota surge una nueva ventana que solicita agregar los cambios a una petición de cambios. Estas pantallas se ilustran en la figura 5.3.3-4.

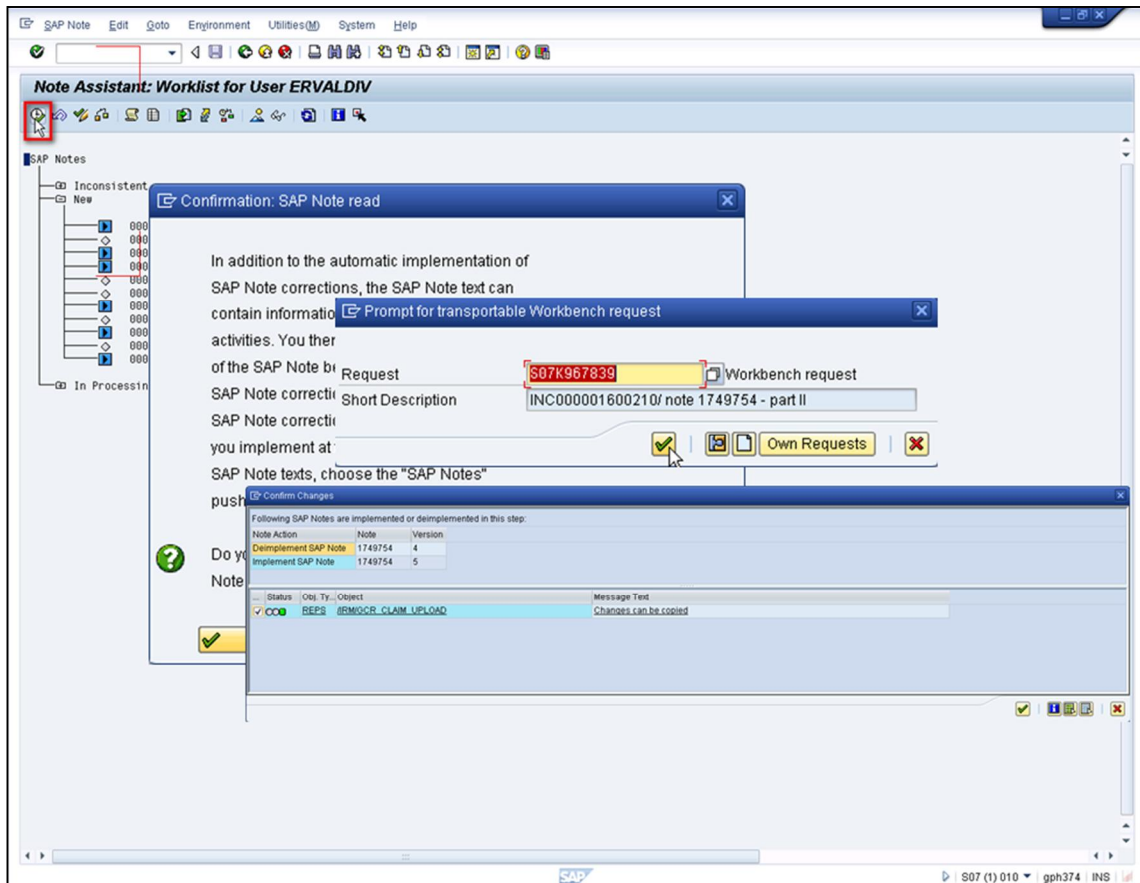


Fig. 5.3.3-4 Implementar nota

En casos excepcionales, tal vez sea necesario quitar una nota que ya se ha implementado. Esto se puede llevar a cabo con la opción "Cancelar implementación de nota".

Si se hace esto, los cambios que habían sido introducidos por la nota son reseteados y el estado de la nota se cambia a aquel que tenía antes de la implementación.

## 5.4. Administración de mandantes del sistema

### 5.4.1. Estructura de datos de un sistema SAP

Los sistemas SAP tienen una estructura de datos específica. Además, de los ajustes de negocio (personalización) que son únicamente relevantes para ciertos mandantes del sistema SAP, un sistema SAP también contiene siempre objetos y ajustes de aplicación a través de todo el sistema y personalización a través de todo el sistema y el repositorio. Un esquema de dicha estructura se muestra en la figura 5.4.1-1.



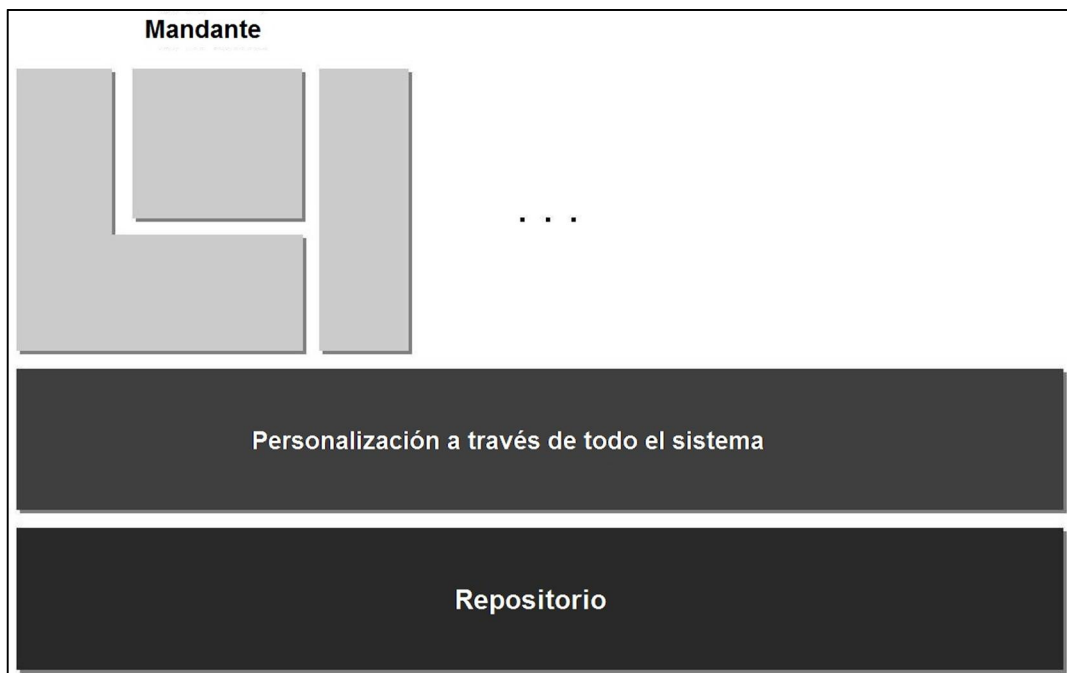


Fig. 5.4.1-1. Estructura de datos de un sistema SAP [TADM10\_12005]

El repositorio es el lugar en donde se guardan de forma centralizada todos los objetos de desarrollo de ABAP, y es a través de todo el sistema. Contiene todos los objetos de diccionario (tablas, dominios), y también contiene todos los programas ABAP, menús y pantallas [TADM10\_12005].

#### 5.4.2. Mandante

En un sistema SAP, un mandante es una unidad autónoma en términos de negocio, organización y datos. Un mandante se caracteriza por el hecho de que tiene su propio entorno de datos de negocios, sus propios datos maestros y transaccionales, y sus propios datos de usuarios. Este tipo de información contenida en un mandante se le conoce como **datos específicos de mandante**. La figura 5.4.2-1 representa los datos que contiene un mandante.

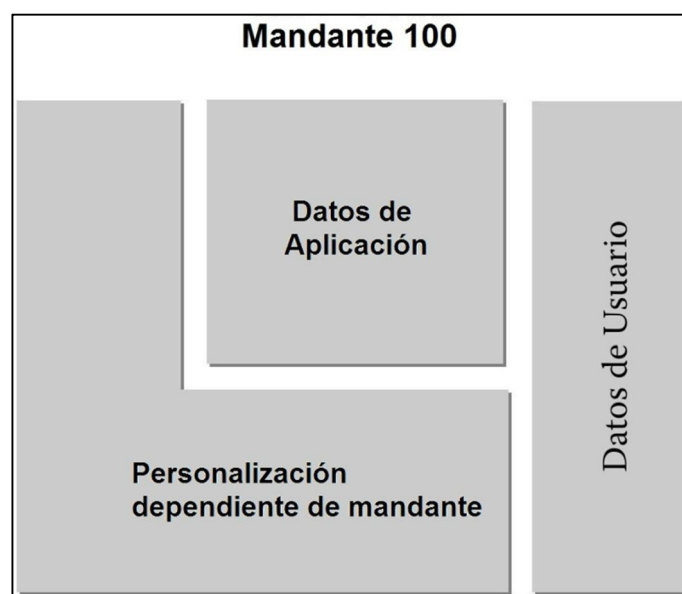


Fig. 5.4.2-1 Mandante [TADM10\_12005]

Los datos dependientes de mandante es aquella información que sólo es válida en ese mandante. Esto incluye, por ejemplo, registros maestros de material, órdenes de compra y facturas.

Los datos de usuarios y autorizaciones son también dependientes de mandante.

En un sistema SAP los mandantes toman diferentes roles. Los siguientes, son los roles estándar en un sistema SAP.

Mandante CUST, desarrollo y personalización, es el mandante central de personalización en donde una adaptación completa del sistema SAP a las necesidades específicas del cliente toma lugar. Todos los cambios realizados en este mandante son guardados para que puedan ser transferidos a otros mandantes.

Mandante QTST, aseguramiento de calidad, se utiliza para probar y verificar los nuevos ajustes de personalización en la aplicación.

Mandante PROD, es el mandante en donde se llevan a cabo las actividades de producción. Los cambios de personalización importados en este mandante tienen que ser probados cuidadosamente en el mandante QTST primero.

Se recomienda que cada uno de estos mandantes exista en un sistema SAP por separado [TADM10\_12005]. Un resumen de los roles principales de los mandantes se presenta en la figura 5.4.2-2.

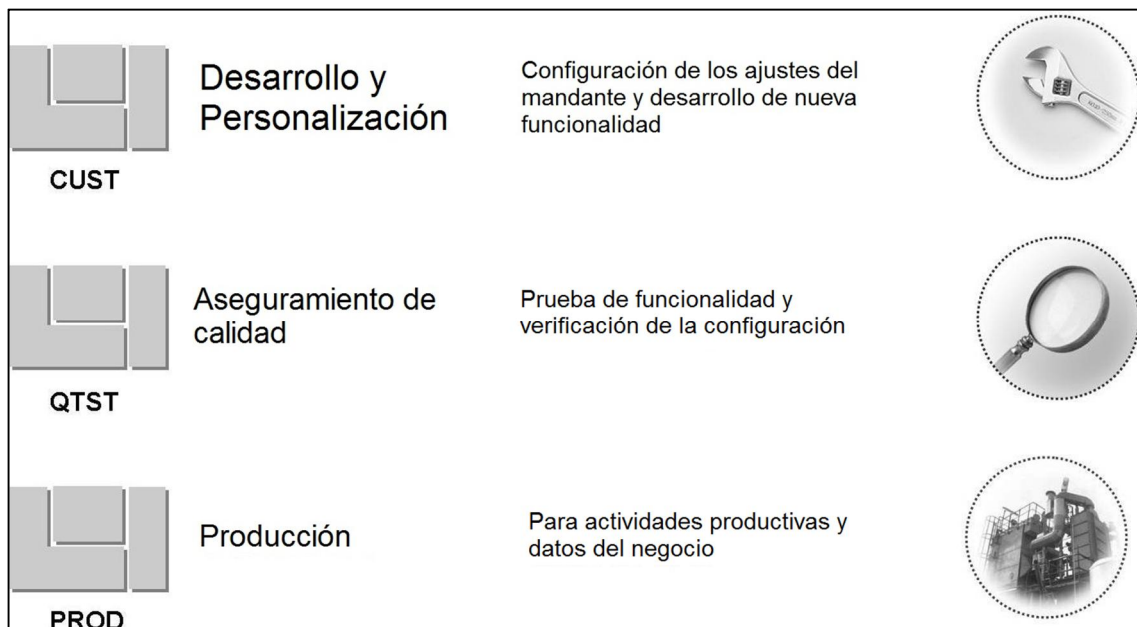


Fig. 5.4.2-2 Roles estándar de los mandantes [TADM10\_12005]

Mandantes adicionales dentro del conjunto de sistemas pueden incluir [TADM10\_12005]:

1. Un mandante “sandbox” para experimentar con las transacciones y los ajustes.
2. Un mandante de prueba para verificar los cambios a la personalización utilizando una muestra de datos.
3. Un ambiente para entrenamiento del usuario final.
4. Un mandante que satisfaga requerimientos específicos del cliente.

En el ejercicio laboral, nos hemos encontrado con muchos clientes que no llevan a cabo estas recomendaciones. Entre las razones más comunes están la falta de conocimiento del sistema, tanto de los clientes como de los consultores que implementaron la solución; y la falta de recursos económicos para montar tres diferentes instancias (desarrollo, calidad y producción). Este material pretende ayudar a mitigar la falta de conocimiento de los consultores. Por otro lado, si el problema es económico, se recomienda al menos tener dos instancias (desarrollo y producción).

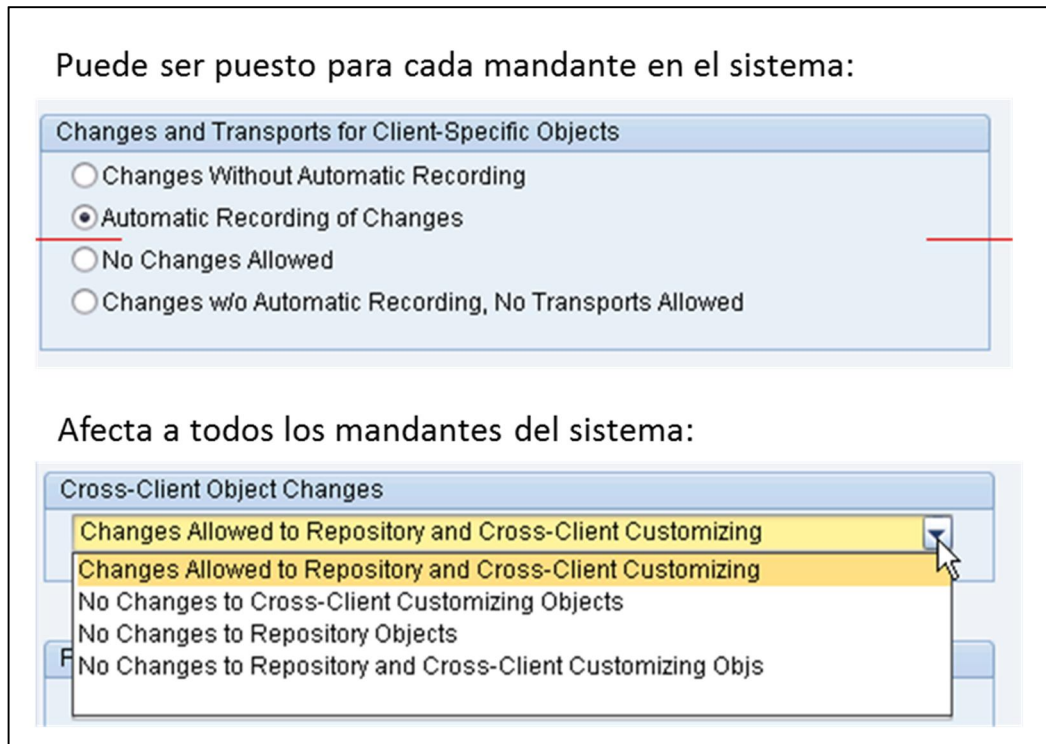
### 5.4.3. Opciones de cambios en mandante

Las opciones de cambios en los mandantes se encuentran en la tabla maestra de mandantes T000, y pueden ser ajustados usando la transacción SCC4.

Los dos ajustes que deben ser mantenidos para implementar un control en dónde los cambios son hechos e imponer que los cambios sean guardados en solicitudes de cambio son:

- Cambios y transportes para objetos específicos de mandante
- Cambios a objetos independientes de mandante

Estas opciones se muestran en la figura 5.4.3-1.



**Fig. 5.4.3-1 Opciones de cambios a mandantes**

A continuación se da una descripción de cada una de las opciones [TADM12\_22005]:

- Cambios y transportes para objetos específicos de mandante:
  - Cambios sin registro automático: Permite realizar cambios a las tablas de personalización pero no incluye los cambios en una petición de cambios. Los cambios pueden ser incluidos manualmente en una petición de cambios para ser transportados.
  - Registro automático de cambios: Permite realizar cambios a las tablas de personalización y automáticamente solicita incluir los cambios en una petición de cambios.
  - No se permiten cambios: Previene a todos los usuarios de hacer cambios a las tablas de personalización.
  - Cambios sin registro automático, no se permiten transportes: Permite hacer cambios a las tablas de personalización específicas de mandante únicamente, pero no permite que los cambios sean transportados, ya sea, manualmente o automáticamente.
- Cambios a objetos independientes de mandante:
  - Cambios a Repositorio y personalización de mandantes permitidos: Permite cambios a las tablas de personalización que afectan a todos los mandantes y al repositorio de SAP y automáticamente solicita incluir los cambios en una petición de cambios.
  - No cambios a objetos de personalización de mandantes: Sólo permite cambios al repositorio de SAP y automáticamente solicita incluir los cambios en una solicitud de cambios.
  - No cambios a objetos de repositorio: Sólo permite cambios a las tablas de personalización que afectan a todos los mandantes y automáticamente solicita incluir los cambios en una solicitud de cambios.

- No cambios a objetos de repositorio ni a objetos de personalización de mandantes: Previene cualquier cambio que afecte a todos los mandantes, ya sean de personalización o de desarrollo.

### 5.5. Programación de trabajos en fondo

Los trabajos de fondo deben esencialmente separar a aquellas tareas que corren por un periodo largo de tiempo o aquellas tareas periódicas de las que requieren interacción por parte del usuario y no requieren un tiempo prolongado de procesamiento. Una ilustración de dicha diferenciación se observa en la figura 5.5-1.

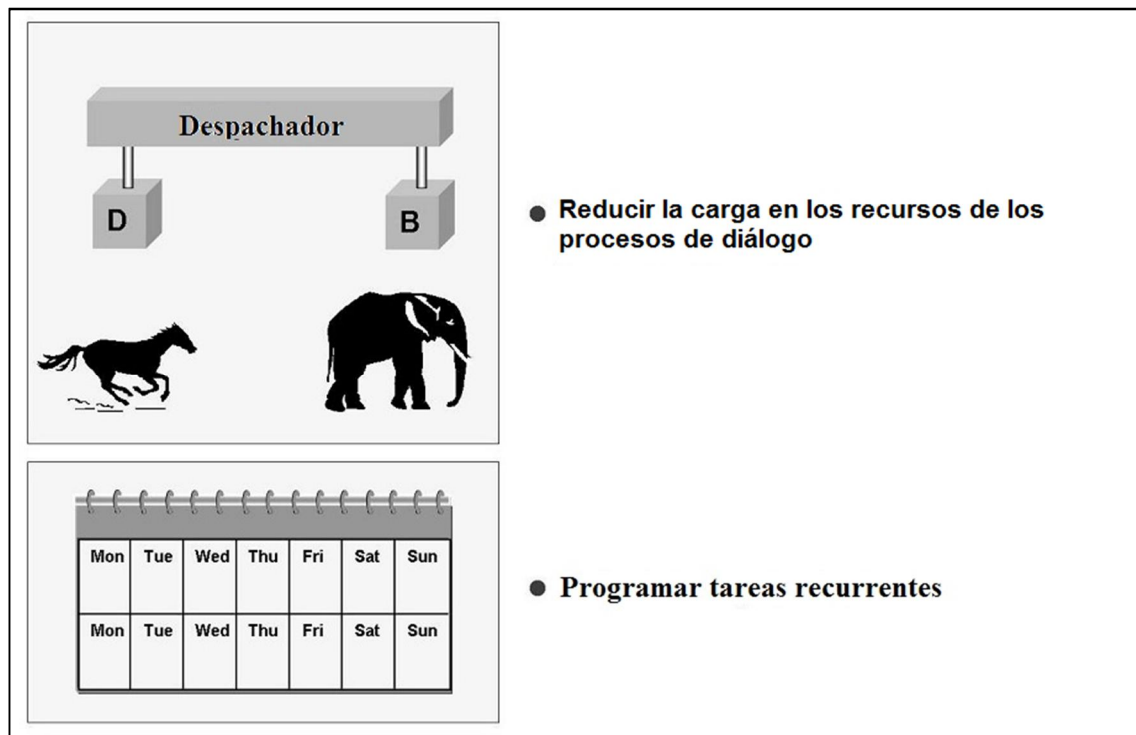


Fig. 5.5-1 Procesos de fondo [TADM10\_12005]

Un proceso de fondo consiste en uno o más pasos. Un paso puede ser:

- Un programa ABAP
- Un comando externo
- Un programa externo

Cualquier programa ABAP puede ser calendarizado como un paso dentro de un proceso de fondo. Si el programa ABAP tiene una o más pantallas de selección, se debe de crear una variante con los valores correspondientes. Una variante permite correr un programa ABAP como proceso en segundo plano aunque el programa requiera que se introduzca información como parte del proceso. Los valores guardados en la variante son utilizados durante la ejecución del programa.

Un comando externo es el llamado de un script predefinido, un comando, o un programa fuera del sistema SAP. Con el uso de la función comandos externos, se pueden enmascarar llamadas al sistema operativo y guardarlas en el sistema SAP bajo un nombre determinado. La ejecución de un comando externo puede ser protegida usando el concepto de autorización de SAP, de esa forma ciertos comandos externos pueden ser ejecutados únicamente por determinados usuarios del sistema SAP.

Un programa externo es un comando en cualquier sistema operativo.

Cada trabajo es procesado sin interrupciones por un solo proceso de trabajo de fondo.

Los procesos de fondo pueden ser calendarizados con diferentes prioridades:

- Clase A (prioridad más alta)
- Clase B (prioridad media)

- Clase C (prioridad normal)

Un proceso de fondo puede ser calendarizado para que inicie:

- En una fecha y tiempo en específico.
- Al ocurrir un evento en particular definido en el sistema SAP (esto incluye procesos que inician al terminar otros procesos o cuando ocurre un cambio en el modo de operación del sistema).

### 5.5.1. Calendarización y Monitoreo de Procesos de Fondo

Para crear y calendarizar procesos de fondo se utiliza la transacción SM36. En esta transacción también existe un asistente para facilitar el proceso.

Para poder crear un nuevo proceso de fondo las siguientes especificaciones son necesarias:

- Especificaciones generales tales como nombre del proceso de fondo, prioridad (C por omisión) y (opcionalmente) servidor objetivo.
- Definición de uno o más pasos
- Definición de las condiciones de inicio (controlado por tiempo o evento)

La transacción SM37 se utiliza para monitorear los procesos de fondo.

En la pantalla inicial de dicha transacción se pueden seleccionar los procesos de fondo utilizando varios criterios. De esta forma, se pueden desplegar todos los trabajos en segundo plano que contienen un paso en particular, que se encuentran en cierto estado, o que reaccionan a cierto evento.

Desde la vista general de los procesos de fondo que se nos presenta como resultado de la selección, es posible ir a otras vistas específicas de los trabajos de fondo:

- Los detalles del proceso de fondo contienen entre otra información, la definición, la duración de ejecución y el tiempo de inicio.
- Todos los mensajes enviados como salida por un proceso en segundo plano se guardan en un log. Este log puede ser desplegado para obtener información de por qué un programa terminó con errores o para realizar una investigación detallada acerca del procesamiento de este trabajo de fondo.

Un proceso de fondo puede tener los siguientes estados:

- **Calendarizado:** Los pasos que conforman este proceso han sido definidos; sin embargo, la condición de inicio aún debe ser definida.
- **Liberado:** El trabajo de segundo plano ha sido completamente definido, incluyendo la condición de inicio.
- **Listo:** La condición de inicio se ha cumplido. El planificador de procesos de fondo ha colocado el proceso en la cola de espera de un proceso de trabajo libre.
- **Activo:** El proceso de fondo se está ejecutando y no puede ser borrado o modificado.
- **Finalizado:** Todos los pasos del proceso fueron completados satisfactoriamente.
- **Cancelado:** El proceso fue terminado. Esto puede pasar por dos causas:
  - Un administrador terminó el proceso deliberadamente.
  - Un paso del proceso terminó debido a errores.

## 6. Administración del servidor de aplicaciones Java

El lenguaje de programación Java fue presentado por primera vez por la compañía Sun Microsystems Inc. en 1995. Java es un lenguaje de programación orientado a objetos e independiente de plataforma. Este concepto permite el desarrollo de un gran número de diferentes tipos de aplicaciones.

SAP toma en cuenta la creciente importancia de Java para crear aplicaciones de negocio desde que la especificación J2EE fue introducida, así como el gran mercado de desarrolladores de Java. SAP comenzó una evaluación fundamental de Java desde 1998 y, con el servidor de aplicaciones web, ahora ofrece una infraestructura completamente compatible con la especificación J2EE.

La arquitectura de un servidor de aplicaciones en concordancia con la especificación J2EE es similar al ambiente de ejecución ABAP usado por las soluciones SAP, ya que ambos son independientes de plataforma, ejecutados en una arquitectura multinivel, y separan la lógica de presentación de aquella de la lógica de negocios [TADM10\_22005].

En este capítulo se presentan las herramientas de configuración y administración del servidor de aplicaciones web SAP Java y sus áreas de aplicación son discutidas. Además, se revisa el ajuste de parámetros utilizando dichas herramientas.

La administración de usuarios en el ambiente Java también es abordada a lo largo de estas páginas. Y se estudiará la instalación de correcciones, tanto para todo el servidor de aplicaciones, como para alguna aplicación Java en específico.

### 6.1. Herramientas de administración y configuración del servidor de aplicaciones Java

En esta primera parte se revisarán las herramientas que existen para la administración y configuración del servidor de aplicaciones Java y sus características. Ciertas tareas de administración sólo se pueden llevar a cabo con una u otra de las herramientas disponibles.

#### 6.1.1. Config Tool

La herramienta "Config Tool" se utiliza para dar mantenimiento a los ajustes del servidor de aplicaciones Java en la base de datos. Por lo tanto, es necesario que la base de datos esté funcionando para poder utilizar el "Config Tool". Los parámetros de la máquina virtual de Java del servidor de aplicaciones Java sólo pueden ser ajustados en el "Config Tool". Una pequeña síntesis de las características de esta herramienta, se observa en la figura 6.1.1-1.

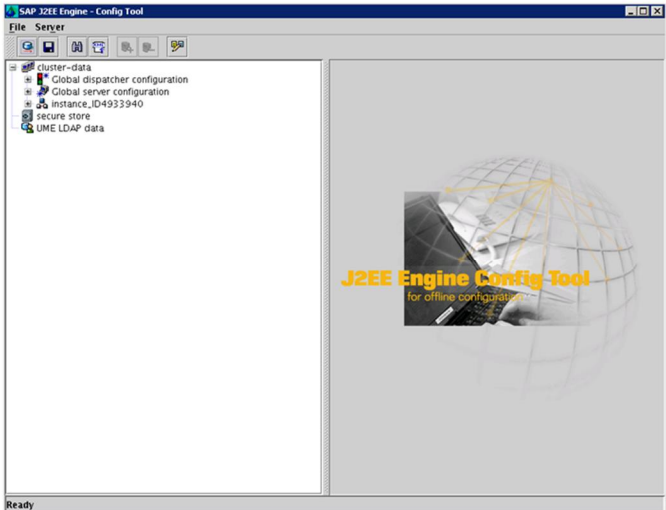
<p><b>Requisitos</b></p> <ul style="list-style-type: none"> <li>- La base de datos debe estar corriendo</li> </ul> <p><b>Uso</b></p> <ul style="list-style-type: none"> <li>- Configuración de la máquina virtual de Java</li> <li>- Configuración de servicios y controladores</li> </ul> <p><b>Características especiales</b></p> <ul style="list-style-type: none"> <li>- Reiniciar el servidor de aplicaciones Java para que los cambios tengan efecto</li> <li>- No se requiere usuario ni contraseña</li> <li>- Sólo se puede ejecutar de manera local en la instancia central</li> </ul>	
---	--

Fig. 6.1.1-1 Config Tool

Las configuraciones hechas con la herramienta “Config Tool” únicamente toman efecto cuando el servidor de aplicaciones Java es iniciado, esto quiere decir, que cada que se hacen cambios con el “Config Tool” es necesario reiniciar el servidor de aplicaciones Java.

El “Config Tool” es iniciado desde el servidor en donde reside la base de datos del servidor de aplicaciones Java. La autenticación se realiza con el usuario que se firmó al sistema operativo.

### 6.1.2. Visual Administrator

Con el “Visual Administrator”, es necesario firmarse a una instancia del servidor de aplicaciones Java. Por lo tanto, es un requerimiento que la instancia esté en funcionamiento. El “Visual Administrator” puede ser instalado en un servidor en donde no haya una instancia de Java. De esa forma, puede ser utilizado para una administración remota. Los cambios realizados en el “Visual Administrator” toman efecto mientras el sistema está corriendo, es decir, no se requiere reiniciar el sistema para que los cambios tomen efecto. El “Visual Administrator” se puede utilizar para detener e iniciar servicios. Las características principales de esta herramienta se ilustran en la figura 6.1.2-1.

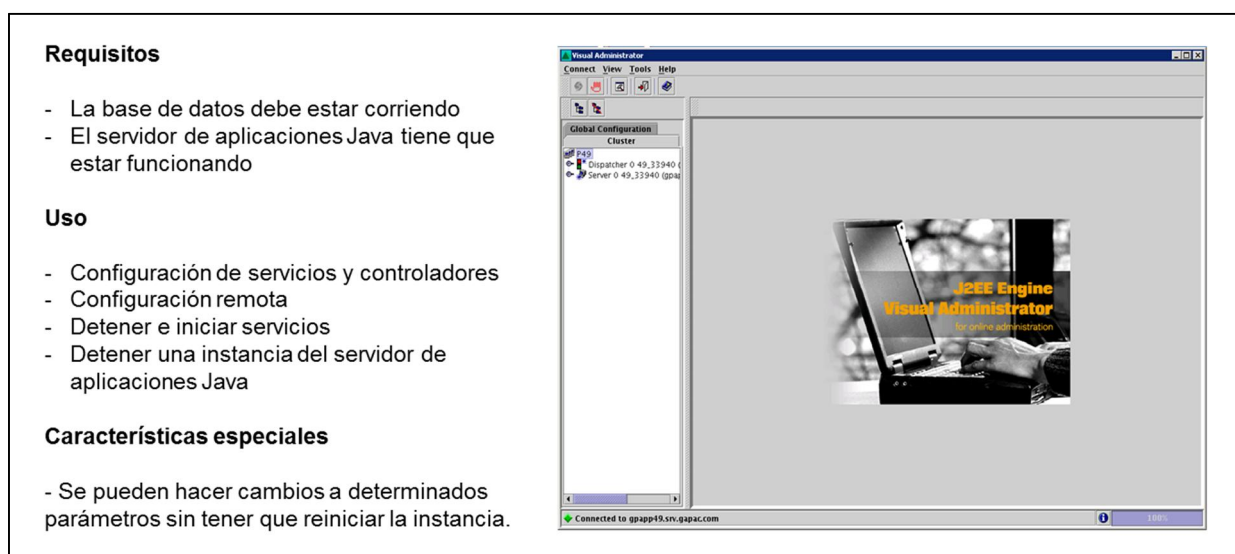


Fig. 6.1.2-1 Visual Administrator

Si es necesario, el “Visual Administrator” también puede ser utilizado para apagar la instancia Java del servidor de aplicaciones deteniendo el despachador Java.

### 6.1.3. Configuración del servidor de aplicaciones SAP Java con la herramienta Config Tool

La ventana de la izquierda del “Config Tool” muestra la estructura del clúster en ajustes globales y en ajustes para los elementos del clúster de forma individual. Los elementos del clúster están siempre asignados a una instancia con un número de instancia. Este número de instancia también está contenido en el nombre del despachador o los procesos servidor. Los parámetros del elemento seleccionado son desplegados y configurados en la ventana de la derecha del “Config Tool”. La figura 6.1.3-1 muestra estos detalles.

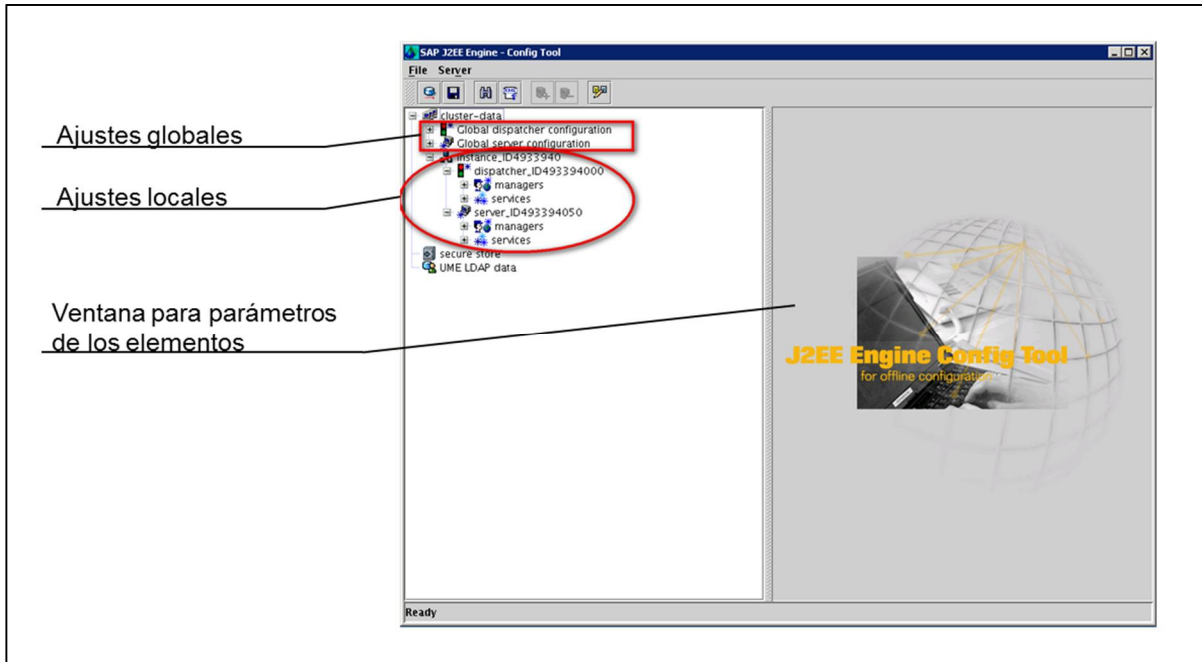


Fig. 6.1.3-1 Secciones del Config Tool

### 6.1.3.1. Ajustes globales en el Config Tool

Los ajustes globales de parámetros se pueden encontrar bajo los nodos de despachador o de servidor que no están asignados a una instancia. Si, por ejemplo, seleccionamos el servicio http debajo del nodo global del despachador, aparecerá una pantalla similar a la que se muestra en el figura 6.1.3.1-1 "Ajustes de parámetros globales". Para cambiar un parámetro, se selecciona el parámetro. El campo "Value" se muestra, listo para ingresar un valor y con un valor por omisión como propuesta. Al ingresar un valor, éste se debe confirmar por medio del botón "Set". El nuevo valor aparece ahora bajo "Custom Value". Para que estos ajustes tengan efecto, es necesario guardar los valores en la base de datos por medio del botón "Apply changes" y reiniciar la instancia del servidor de aplicaciones web Java.

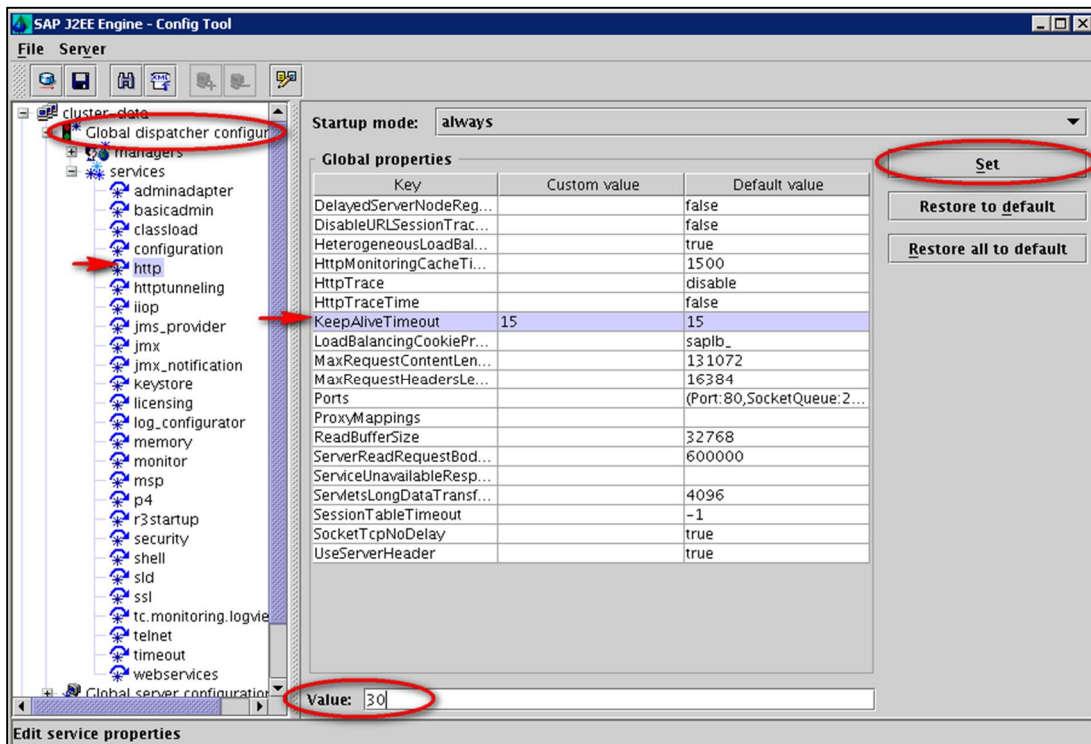


Fig. 6.1.3.1-1 Ajustes de parámetros globales



### 6.1.3.2. Ajustes locales en el Config Tool

La figura 6.1.3.2-1 nos muestra nuevamente el servicio http. En esta figura se muestra el valor configurado de manera global para el parámetro “*KeepAliveTimeout*”. Es posible sobreponer un valor local para el nodo del despachador seleccionado. Para hacer esto, se selecciona el parámetro de la sección “Global properties”. El campo “*Value*” aparece, listo para introducir un valor y asignarlo con el valor de las propiedades globales. Se edita el valor y se confirma al presionar el botón “*Set*”. El parámetro ahora aparece bajo la sección “Local properties” con el nuevo valor bajo el campo “*Custom value*”.

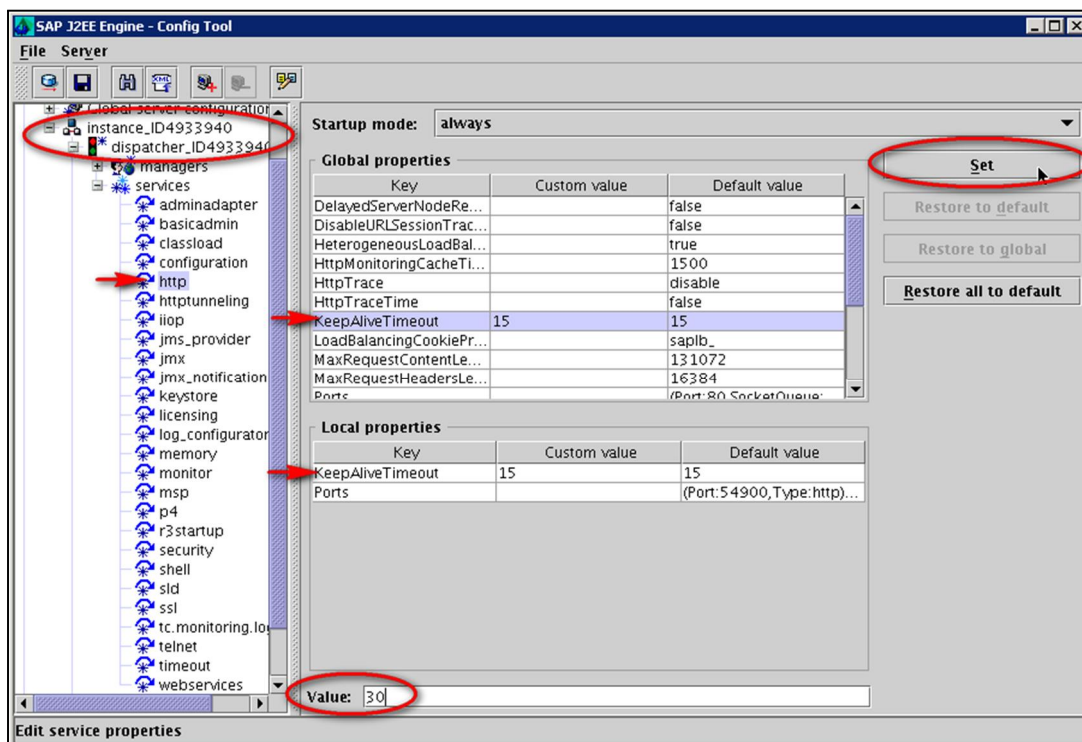


Fig. 6.1.3.2-1 Ajustes locales personalizados

Para que estos valores tomen efecto, se requiere guardar los cambios en la base de datos con el botón “*Apply changes*” y reiniciar la instancia del servidor de aplicaciones Java.

Es importante recordar que los ajustes locales anulan a los valores globales y que los valores personalizados tienen preferencia ante los valores por omisión.

### 6.1.3.3. Reseteando valores

Se pueden resetear valores modificados a los valores de los ajustes globales o a los valores por omisión. Con el botón “*Restore to Default*”, la entrada bajo “*Custom value*” es borrada para los ajustes locales y el parámetro seleccionado regresará al valor por omisión. Si se escoge el botón “*Restore to global*”, la entrada completa es borrada de los ajustes locales. En este caso, los ajustes globales son usados para este parámetro.

### 6.1.4. Configuración del servidor de aplicaciones SAP Java con la herramienta Visual Administrator

El “Visual Administrator” tiene dos diferentes vistas: la vista para una configuración global, y la vista para la configuración local, que también se conoce como configuración de clúster. En la configuración de clúster todos los nodos activos del clúster se muestran y pueden ser ajustados con el “Visual Administrator”. En el caso de la configuración global, el mantenimiento de los nodos del despachador y servidores se realiza de forma independiente utilizando la pestaña correspondiente. Dicha diferencia se puede observar en la figura 6.1.4-1.

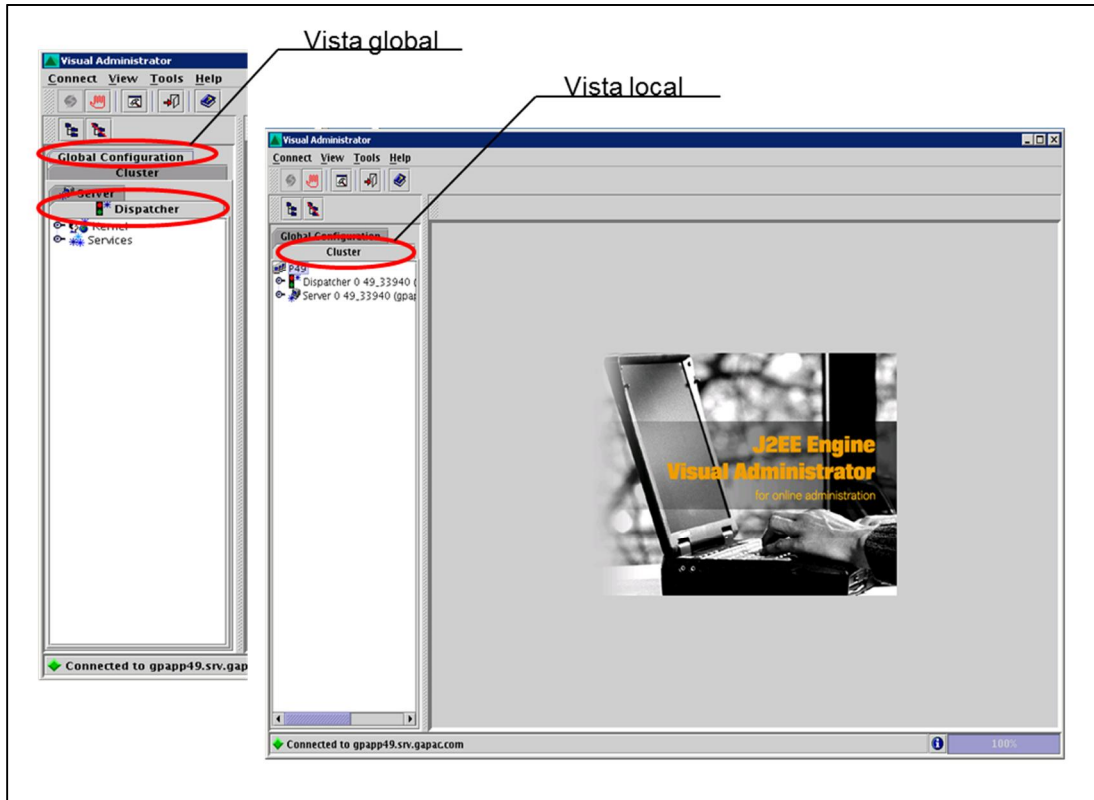


Fig. 6.1.4-1 Configuración global vs local

En la figura 6.1.4-2 se muestran los íconos más importantes para la administración de los parámetros utilizando el “Visual Administrator”.

- “Update” (Actualizar)
- “Save Properties” (Guardar propiedades)
- “Reset Properties” (Reseteo propiedades)
- “Reset Default Properties” (Reseteo al valor por omisión)
- “Show Details” (Mostrar detalles)

Cuando se modifica un valor se transfiere a la vista utilizando el botón “*Update*”. Para activar los valores y guardarlos en la base de datos es con el botón “*Save Properties*”. Mientras los valores modificados no hayan sido guardados, se pueden deshacer los cambios con “*Reset Properties*”. Se pueden desplegar los valores por omisión con “*Show Details*”. El botón “*Reset Default Properties*” resetea el valor del parámetro al valor especificado bajo “*Default Value*”.

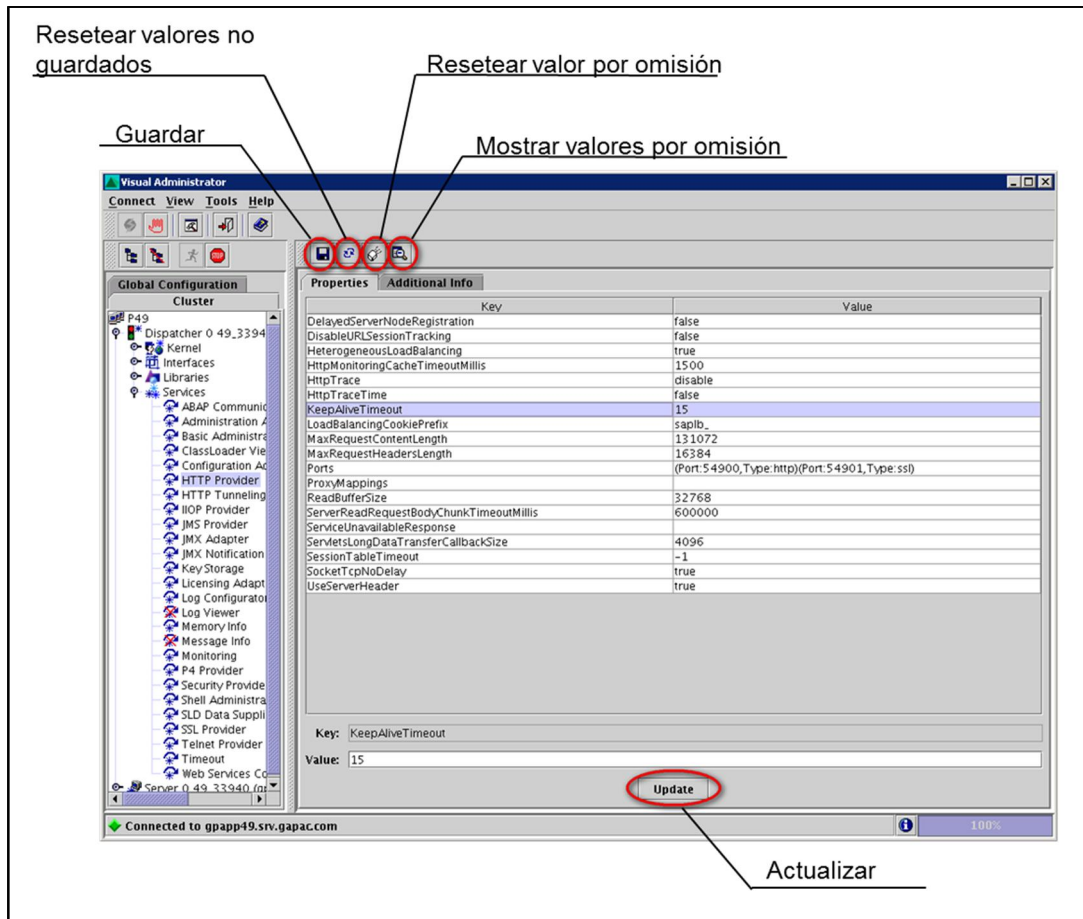


Fig. 6.1.4-2 Valores locales e íconos

### 6.1.4.1. Visual Administrator: Configuración global

Al hacer ajustes globales, también es posible transferir los valores modificados de un parámetro a los ajustes locales. Esto lo revisaremos con más detalle en la sección Configuración global con mantenimiento local. A continuación, se considera el mantenimiento de parámetros globales sin mantenimiento local.

#### 6.1.4.1.1. Configuración global sin mantenimiento local

Los ajustes globales se realizan en la pestaña "Global Configuration". Una vez más, tomaremos como ejemplo el mantenimiento del parámetro "KeepAlive Timeout". Para ello, seleccionamos "Global Configuration → Dispatcher → HTTP Provider". Al seleccionar el parámetro se puede ajustar el valor bajo el campo "Value". El valor ajustado se transfiere a la vista al dar clic en el botón "Update". El mantenimiento del parámetro se finaliza al dar clic en el botón "Save Properties".

Una nueva ventana aparece, la cual ofrece transferir los ajustes a los nodos del clúster. En este caso, no deseamos esta opción, por lo tanto seleccionamos "No". Estos pasos se ilustran en las figuras 6.1.4.1.1-1 y 6.1.4.1.1-2.

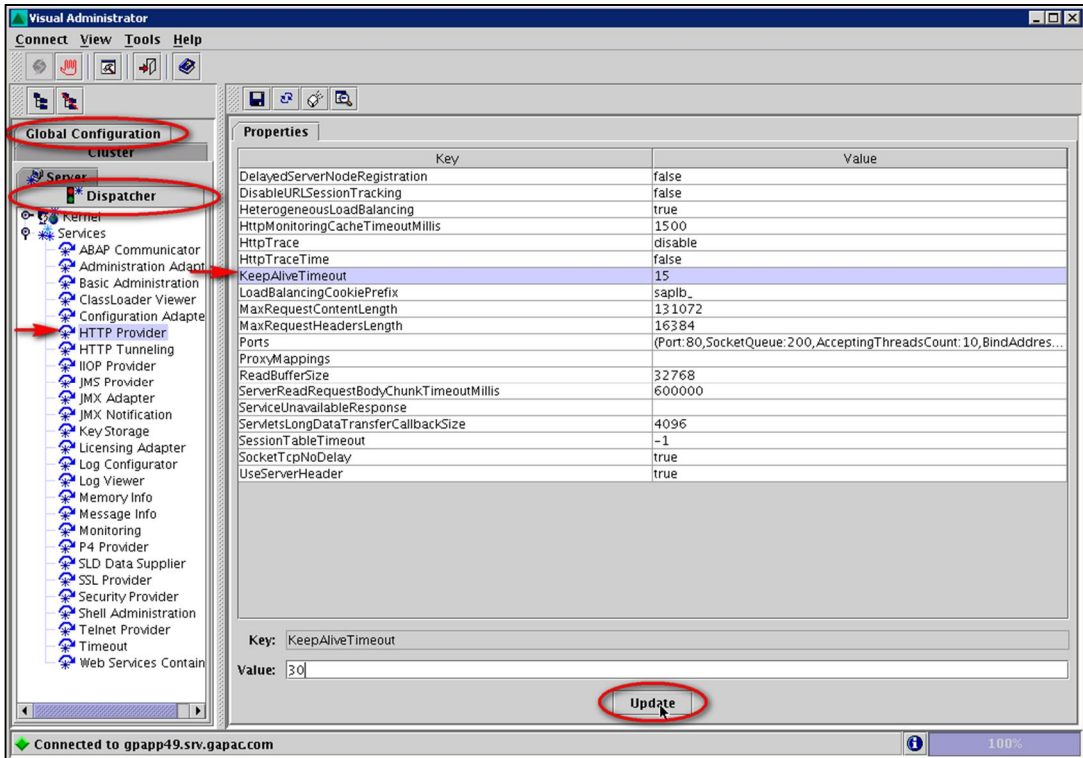


Figura 6.1.4.1.1-1 Ajustes globales 1a

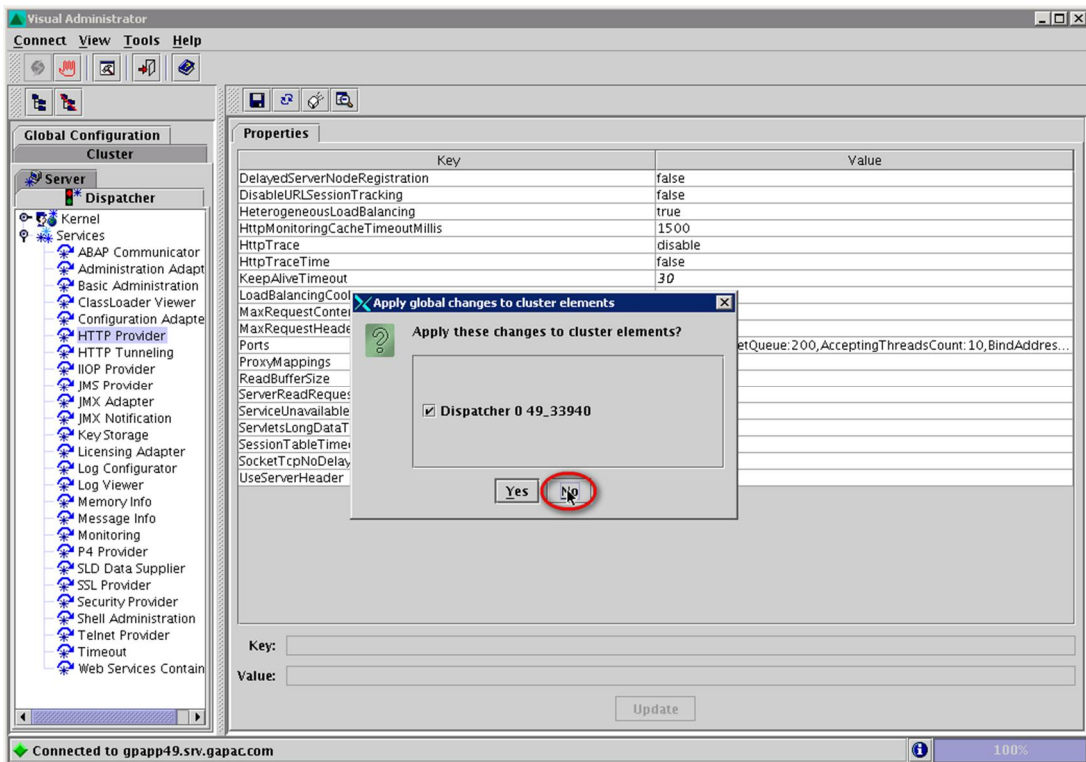


Figura 6.1.4.1.1-2 Ajustes globales 1b

### 6.1.4.1.2. Configuración global con mantenimiento local

Para transferir las modificaciones globales a valores locales, se procede de forma similar a la sección anterior. En esta ocasión, se decide transferir los ajustes a los nodos del clúster al seleccionar "Yes" en lugar de "No". De esta forma, el nuevo valor del parámetro seleccionado se verá reflejado tanto en los ajustes globales como en los ajustes locales.

### 6.1.4.2. Visual Administrator: Configuración de clúster (local)

Los ajustes de clúster son modificados en la pestaña "Cluster". Aquí se pueden seleccionar todos los nodos activos. Como ejemplo, ajustaremos nuevamente el parámetro "KeepAlive Timeout" del servicio "HTTP Provider" del nodo despachador. El procedimiento es muy similar a los vistos en las secciones anteriores. Los cambios realizados toman efecto inmediatamente, a menos que salte una ventana indicando lo contrario. Algunos de los mensajes que pueden aparecer indicando lo contrario, puede ser, por ejemplo, que el servicio tenga que ser reiniciado primero, etc. El procedimiento se ilustra en las figuras 6.1.4.2-1 y 6.1.4.2-2.

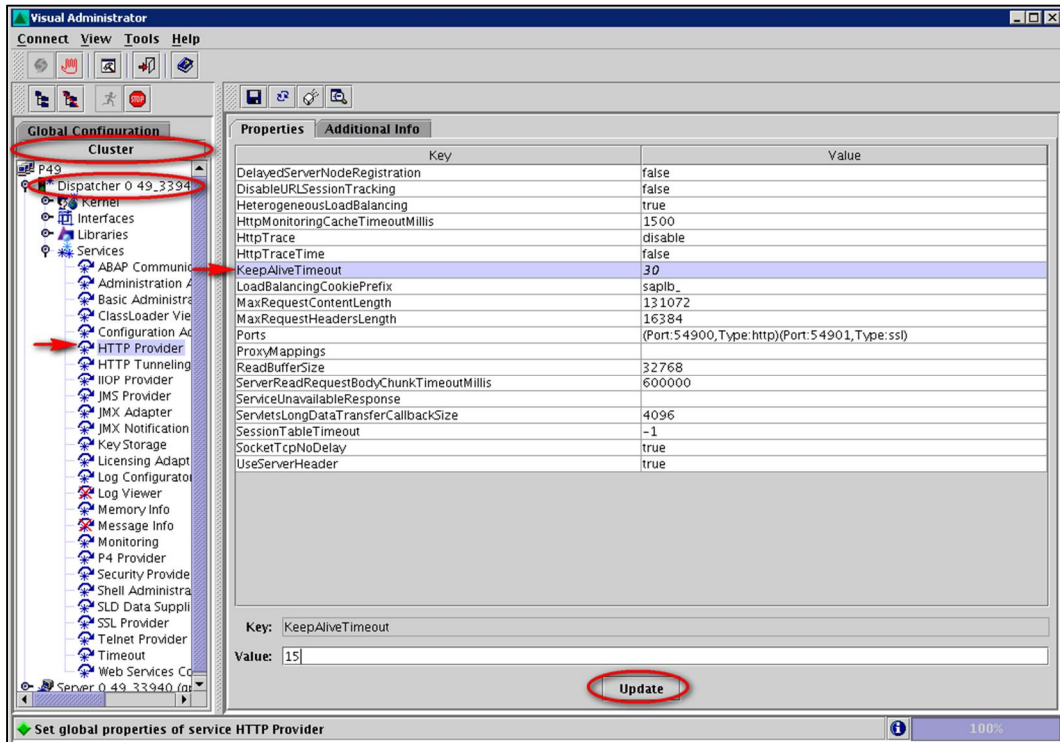


Fig. 6.1.4.2-1 Visual Administrator: Ajustes locales 1a

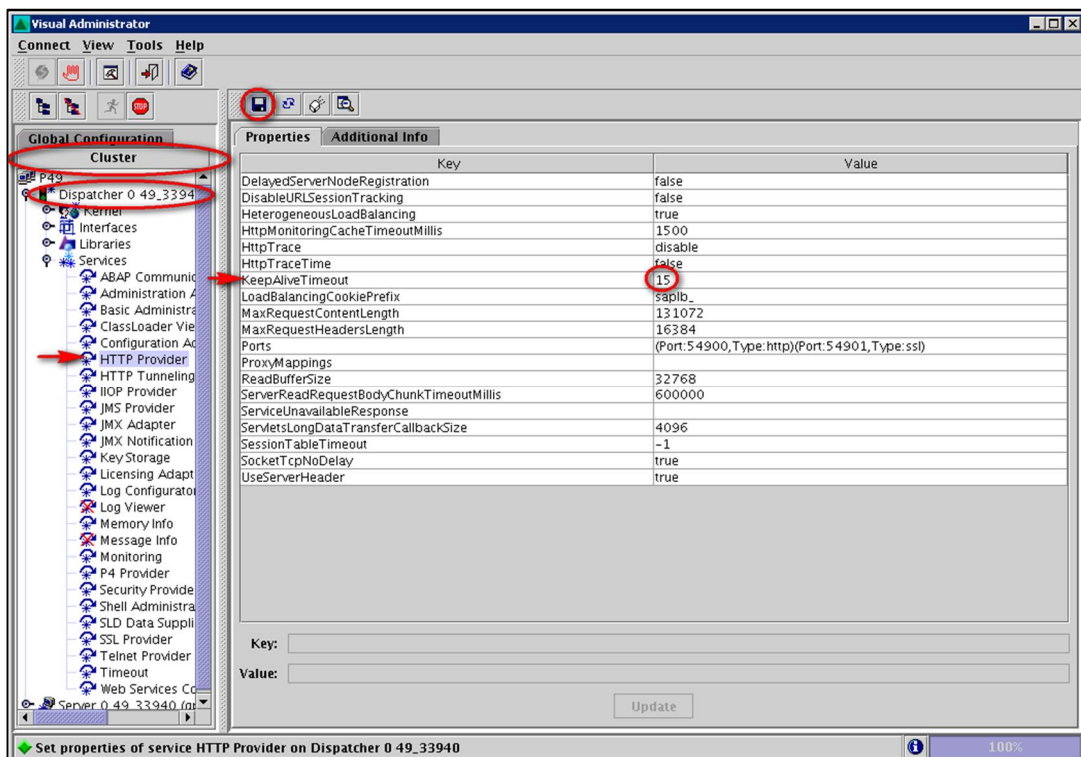


Fig. 6.1.4.2-2 Visual Administrator: Ajustes locales 1b

## 6.2. Administración de versiones

Para cualquier tipo de software existen correcciones que requieren ser instaladas como parte del ciclo de vida del software. En esta sección haremos una revisión de la actualización de los componentes del servidor de aplicaciones web Java y de la actualización de aplicaciones que corren en el mismo. Para ello, también revisaremos las herramientas que se utilizan para dichas tareas.

### 6.2.1. Instalar correcciones para el servidor de aplicaciones Java

SAP proporciona correcciones y actualizaciones a los componentes del sistema en paquetes de soporte llamados “Support Packages” en inglés. Los paquetes de soporte se pueden obtener por medio del “SAP Service Marketplace”. Es posible descargar un paquete de soporte para un componente en específico, o se puede descargar un grupo de paquetes para todos los componentes. Este grupo de paquetes se conoce como pila de paquetes de soporte o “Support Package Stack” en inglés, y es una combinación compatible de paquete de los diferentes componentes de software. Este agrupamiento de “Support Packages” ha sido probado por SAP y por lo tanto, SAP recomienda que se apliquen dichos parches en conjunto.

Cada “Support Package Stack” cuenta con su documentación en forma de un documento PDF, y se puede descargar del “SAP Service Marketplace”.

El procedimiento general para la instalación de “Support Packages” en el servidor de aplicaciones Java consiste en los siguientes pasos:

1. Leer la documentación correspondiente al “Support Package Stack” a ser instalado.
2. Descargar los archivos del “SAP Service Marketplace”.
3. Generar el archivo XML del “Support Package Stack”.
4. Instalar los “Support Packages”.

Durante las siguientes páginas, se describe más a detalle este procedimiento, haciendo uso de pantallas capturadas durante el ejercicio de esta actividad para uno de los clientes con quién trabajamos en el pasado.

Para descargar el “Support Package Stack”, primero es necesario seleccionar el tipo de sistema que se quiere actualizar. La figura 6.2.1-1 ilustra este paso.

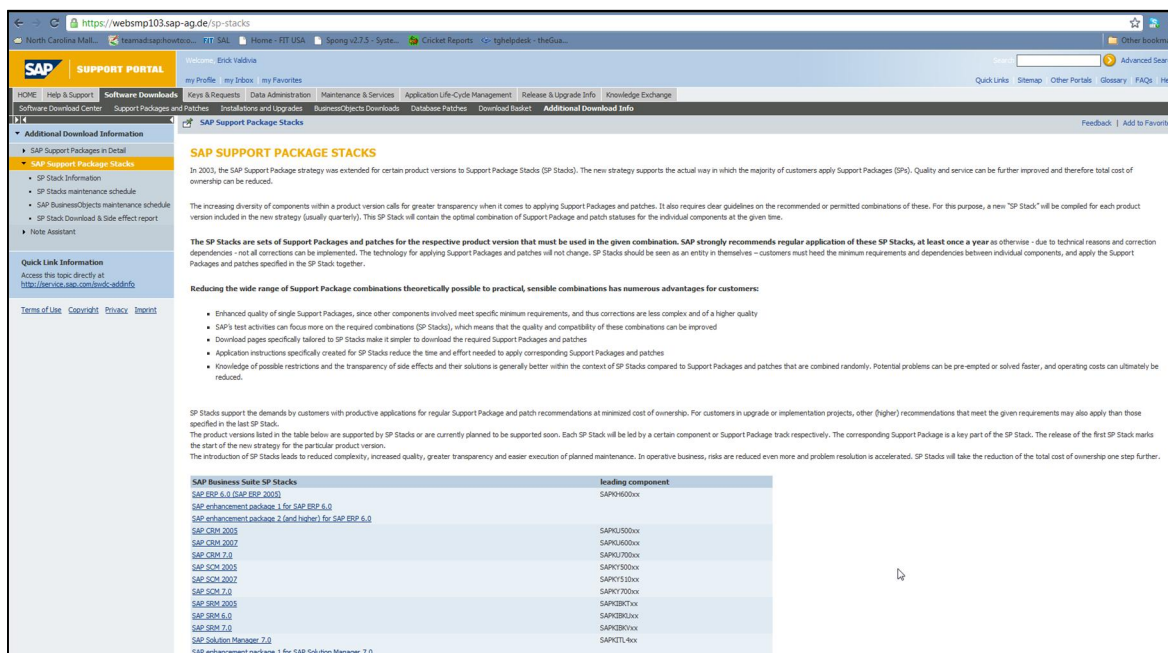


Fig. 6.2.1-1 Selección de tipo de sistema

La figura 6.2.1-2 muestra la selección del número de “Support Package Stack” que se desea descargar.

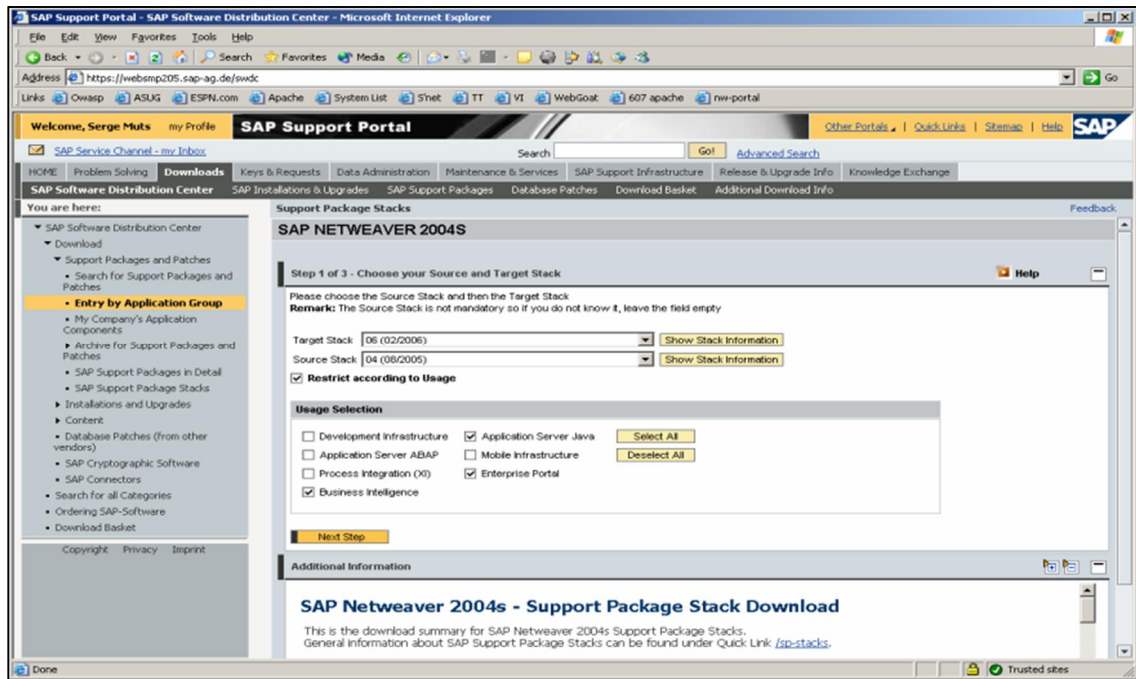


Fig. 6.2.1-2 Selección del número de Support Package Stack

A cada “Support Package Stack” le corresponde un nivel de kernel. Sin embargo, SAP recomienda instalar el último nivel de kernel disponible en el “Service Marketplace”. Por este motivo, al momento de descargar el “Support Package Stack”, uno puede optar por no descargar el kernel correspondiente y descargarlo de forma independiente. Dicha elección se aprecia en la figura 6.2.1-3.

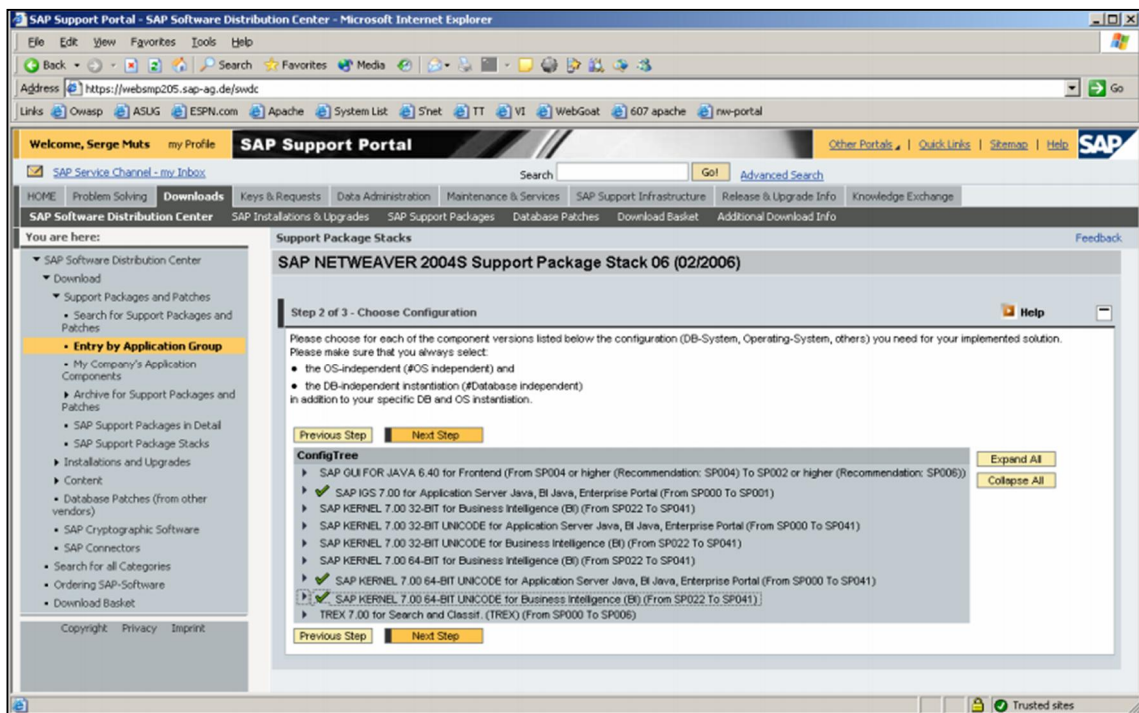


Fig. 6.2.1-3 Descargar archivos de kernel

Al final de nuestra selección, se mostrarán todos los archivos que serán descargados como parte del “Support Package Stack”. En caso de querer eliminar cualquiera de los archivos, basta con remover la marca del recuadro. Esto se puede llevar a cabo en la pantalla que ilustra la figura 6.2.1-4.

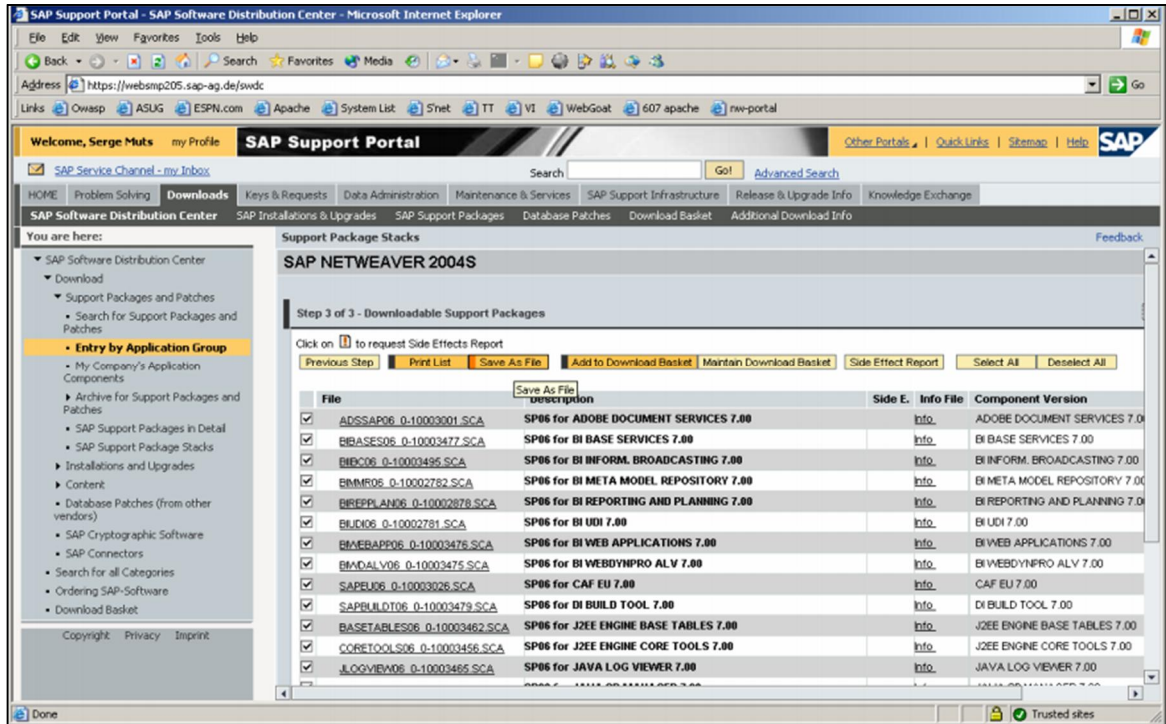


Fig. 6.2.1-4 Selección de archivos

Una vez que se ha tomado una decisión acerca de los archivos que se descargarán, es necesario crear un archivo XML que contendrá la información de los componentes a los que se les aplicarán "Support Packages". La herramienta JSPM leerá el contenido de este archivo y aplicará los "Support Packages" en el orden adecuado. Para crear dicho archivo, se da clic en la liga que indica guardar lista como archivo XML, como se muestra en la figura 6.2.1-5.

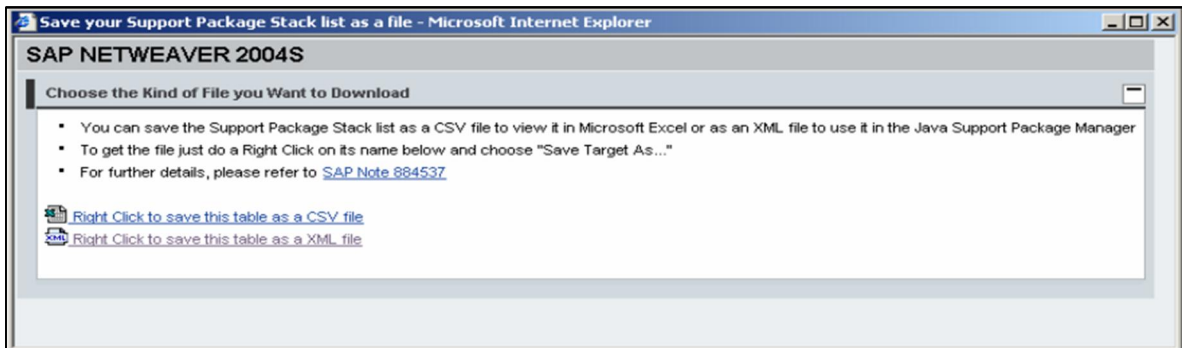


Fig. 6.2.1-5 Creación de archivo XML

Ahora que los archivos han sido descargados y que hemos creado el archivo XML con la información de los "Support Packages" a ser aplicados, se puede proceder con la instalación de los parches. Cabe mencionar que dicha actividad tiene que ser programada en una ventana de mantenimiento, ya que el servidor de aplicaciones Java será reiniciado varias veces por el JSPM durante la implementación de los parches.

Para aplicar los parches se utiliza el JSPM. Al iniciar el JSPM, nos solicitará la contraseña del administrador de distribución de software (SDM, por sus siglas en inglés), como se aprecia en la figura 6.2.1-6. El SDM es otra herramienta específica del servidor de aplicaciones Java y tiene como propósito la instalación de nuevas aplicaciones, la desinstalación de aplicaciones o la reinstalación de aplicaciones.



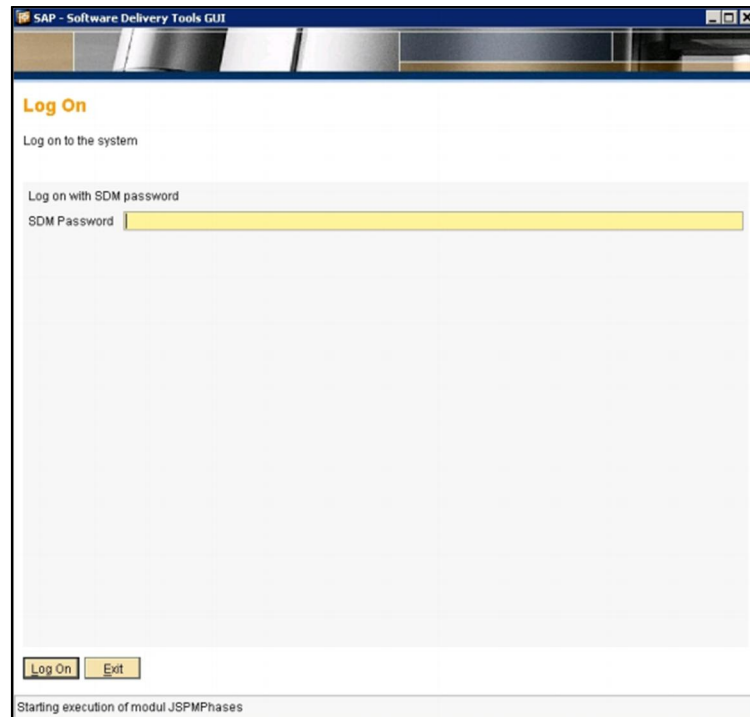


Fig. 6.2.1-6 Autenticación en el SDM

Al introducir la contraseña correcta, el JSPM nos mostrará una nueva pantalla con diferentes opciones (figura 6.2.1-7). Dependiendo de la actividad que se desee realizar será la opción que se elegirá. En el caso de la instalación de “Support Packages” se puede optar por dos opciones: “Support Package Stack” o “Single Support Package”. Al seleccionar “Support Package Stack”, el JSPM buscará el archivo XML con la definición del “stack” y en caso de encontrarlo, instalará los “Support Packages” a todos los componentes en el orden adecuado. La opción “Single Support Package”, es útil para instalar correcciones a un componente del sistema de forma individual.

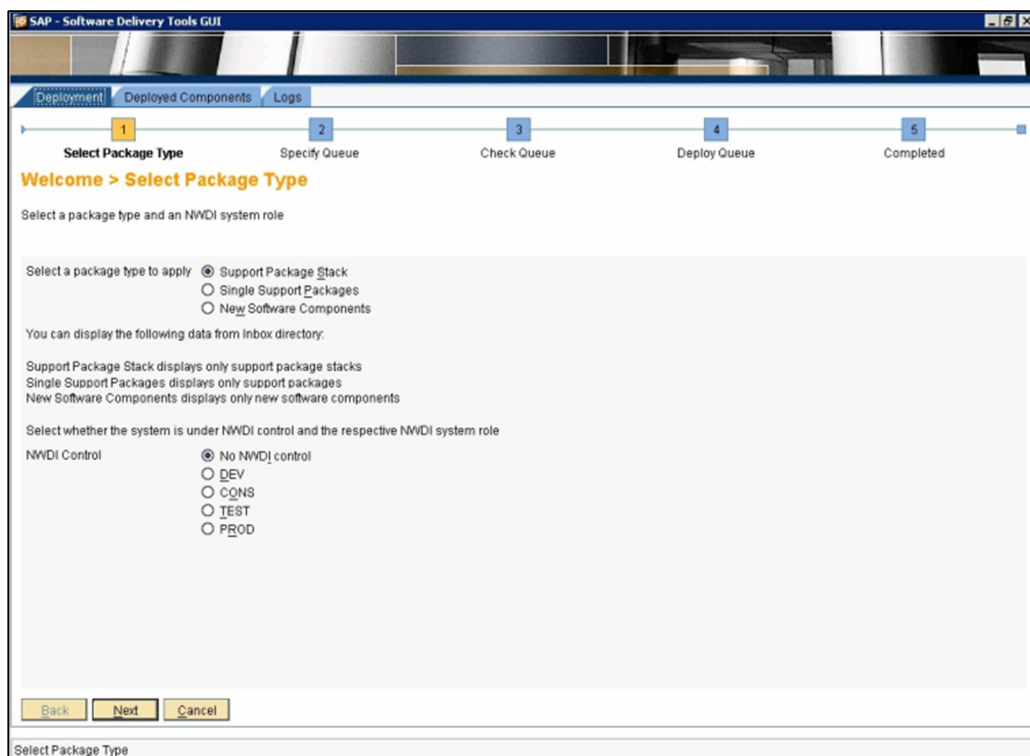


Fig. 6.2.1-7 Selección tipo de instalación

En caso de que el JSPM encuentre algún archivo XML con la descripción de algún “stack”, lo mostrará en la siguiente pantalla, tal cual se ilustra en la figura 6.2.1-8.

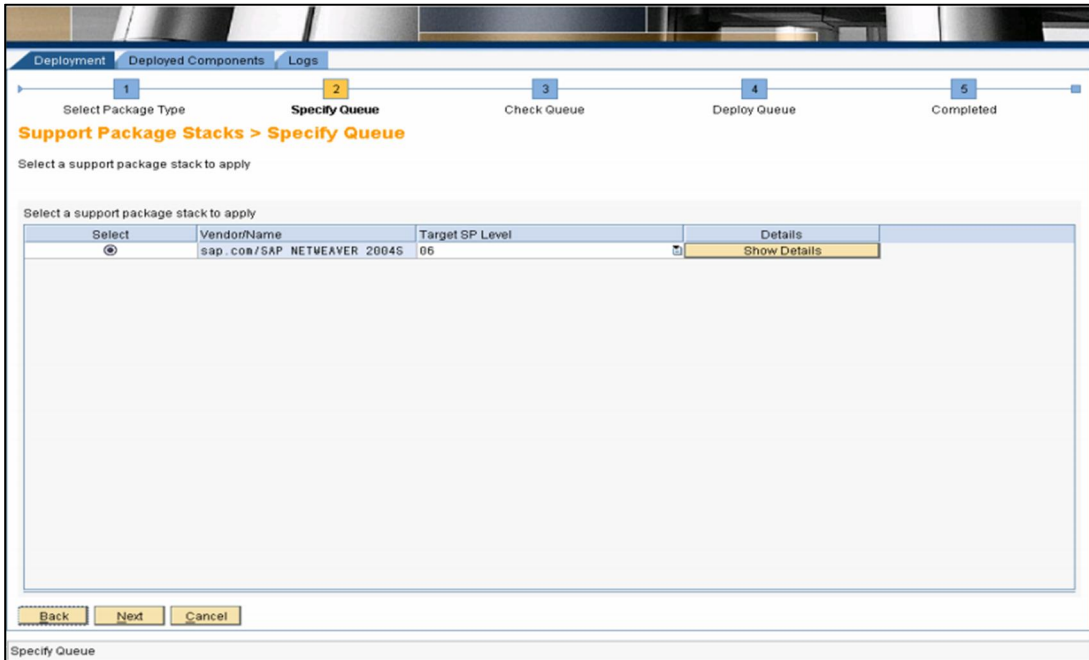


Fig. 6.2.1-8 Archivo XML

La figura 6.2.1-9 nos muestra la siguiente pantalla, en la cual, el JSPM lista todos los componentes del sistema y la versión actual; si existe un "Support Package" para algún componente, también muestra la nueva versión y la condición de dicho parche.

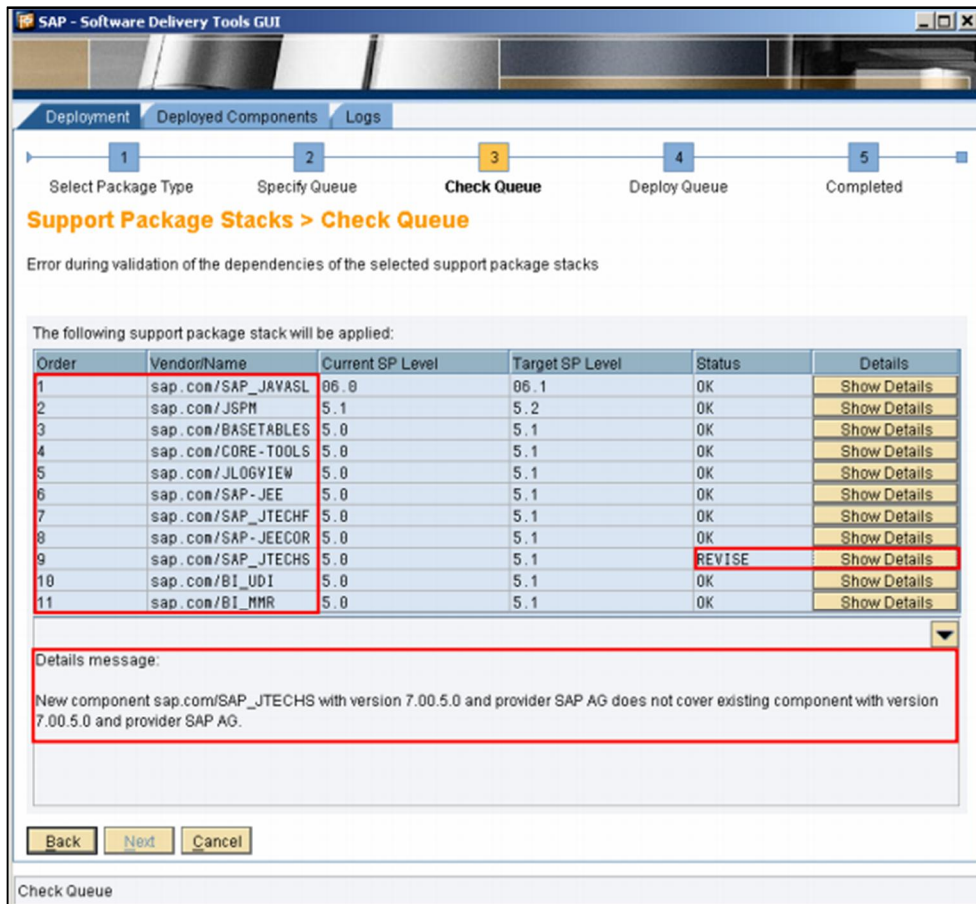


Fig. 6.2.1-9 Lista de componentes y versiones

El proceso de implementación de los parches comienza al dar clic en el botón “Next”. El JSPM se encarga de aplicar los parches en el orden adecuado a todos los componentes, reinicia el sistema cuando así es requerido y al final muestra el estado de la implementación de cada parche.

Una vez concluida la operación se verifica en el sistema, que los componentes muestren la nueva versión.

### **6.2.2. Instalar correcciones para una aplicación Java**

La información necesaria para actualizar una aplicación en específico se puede encontrar en las notas de SAP en el “Service Marketplace”.

Los archivos requeridos para la actualización se obtienen en el “Service Marketplace”.

La distribución de los archivos de la nueva versión se hace mediante la herramienta SDM (Software Deployment Manager). Dependiendo de la aplicación a ser actualizada, puede ser que los archivos tengan que ser distribuidos fuera de línea, es decir, el servidor de aplicaciones Java tiene que ser detenido, o también, puede ser que se puedan distribuir en línea, es decir, el servidor de aplicaciones Java tiene que ser iniciado.

#### **6.2.2.1. Distribución fuera de línea del SDM**

Para una distribución fuera de línea con el SDM, se necesita detener el servidor de aplicaciones Java. Ya que al detener el servidor de aplicaciones, el SDM también se detendrá, es necesario iniciarlo manualmente en modo “standalone”. Existe un script apropiado para cada sistema operativo.

##### **Microsoft Windows**

En el directorio <Unidad>:\usr\sap<SID>\<instancia\_central>\SDM\program

Ejecutar los siguientes comandos sucesivamente:

```
sdm.bat jstartup "mode=standalone"
```

```
StartServer.bat
```

##### **Unix**

En el directorio /usr/sap/<SID>/<instancia\_central>/SDM/program

Ejecutar los siguientes comandos sucesivamente:

```
sdm.sh jstartup mode=standalone
```

```
StartServer.sh
```

Después de haber iniciado el SDM, se necesita iniciar el SDM GUI de la siguiente forma:

##### **Microsoft Windows**

Iniciar RemoteGui.bat en el directorio: <Unidad>:\usr\sap<SID>\<instancia\_central>\SDM\program

##### **Unix**

Iniciar RemoteGui.sh en el directorio: /usr/sap/<SID>/<instancia\_central>/SDM/program

A continuación, se ilustra el proceso de actualización de una aplicación de java, utilizando pantallas tomadas durante la actividad realizada para uno de los muchos clientes con los cuales hemos trabajado. El primer paso, consiste en firmarse al servidor SDM seleccionando “Connect to SDM Server”, y, posteriormente, introduciendo la información correspondiente y eligiendo “Login”, como se ilustra en la figura 6.2.2.1-1.

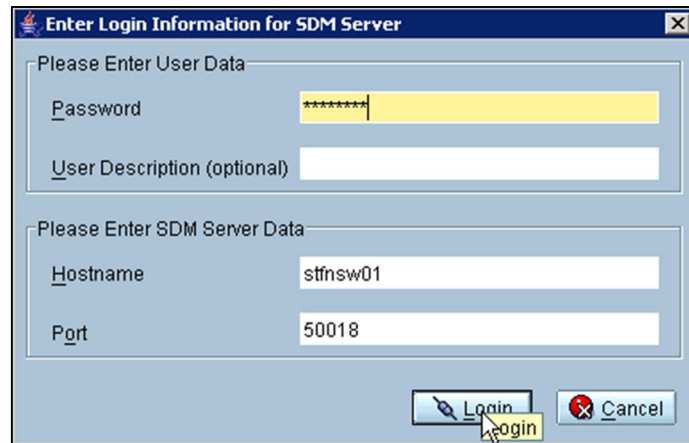


Fig. 6.2.2.1-1 Firmándose al servidor SDM

El puerto por omisión del SDM es 5<número de instancia>18, en este caso, 50018 si el número de instancia es 00.

En la pestaña “Deployment”, uno puede seleccionar los archivos que se desean distribuir con “Add SCA/SDA to Deployment List (Local File Browser)”. En la figura 6.2.2.1-2 se observa dicha acción.

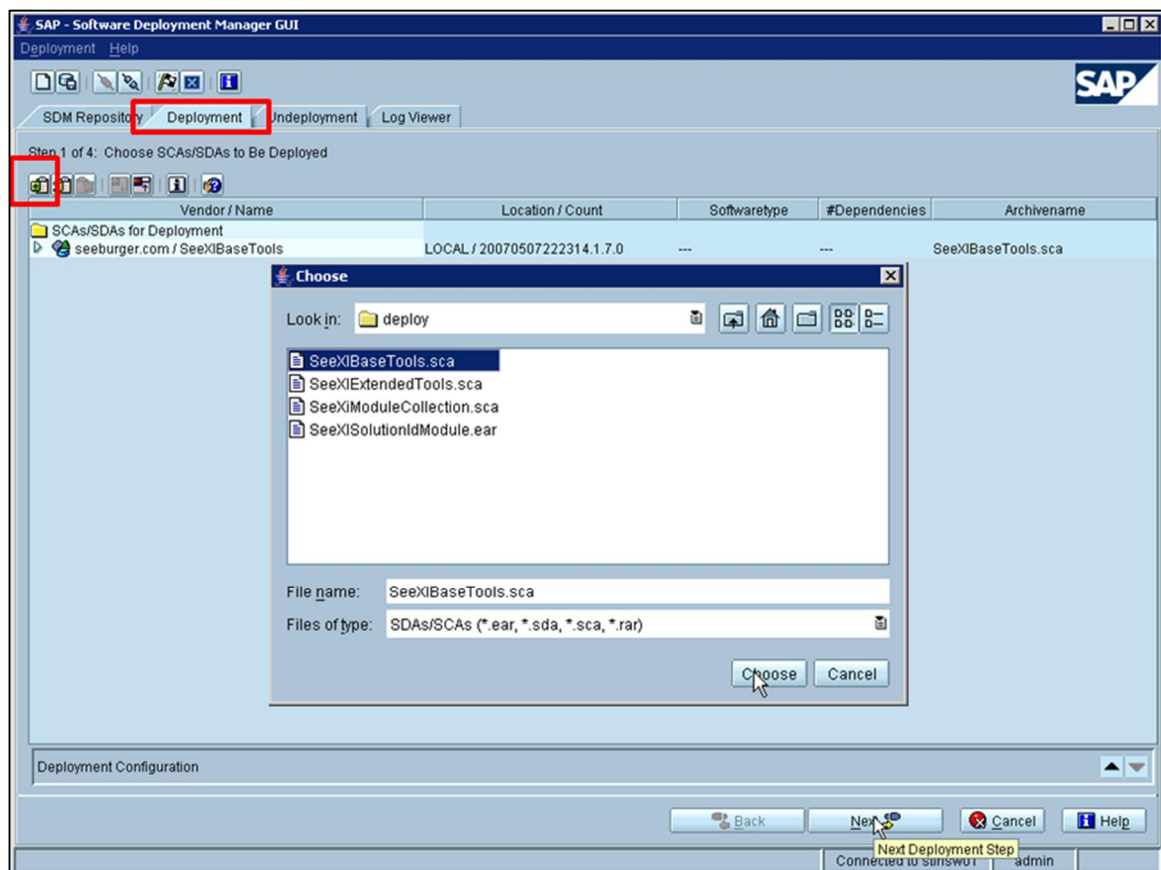


Fig. 6.2.2.1-2 Seleccionando los archivos

Una vez seleccionados los archivos, se confirma la selección con “Choose”.

Después se presiona “Next” varias veces y luego “Start Deployment”. La distribución comienza. Al final se confirma la distribución seleccionando “Confirm” (figura 6.2.2.1-3), y se puede cerrar el SDM GUI. Para esto, se escoge “Exit”.

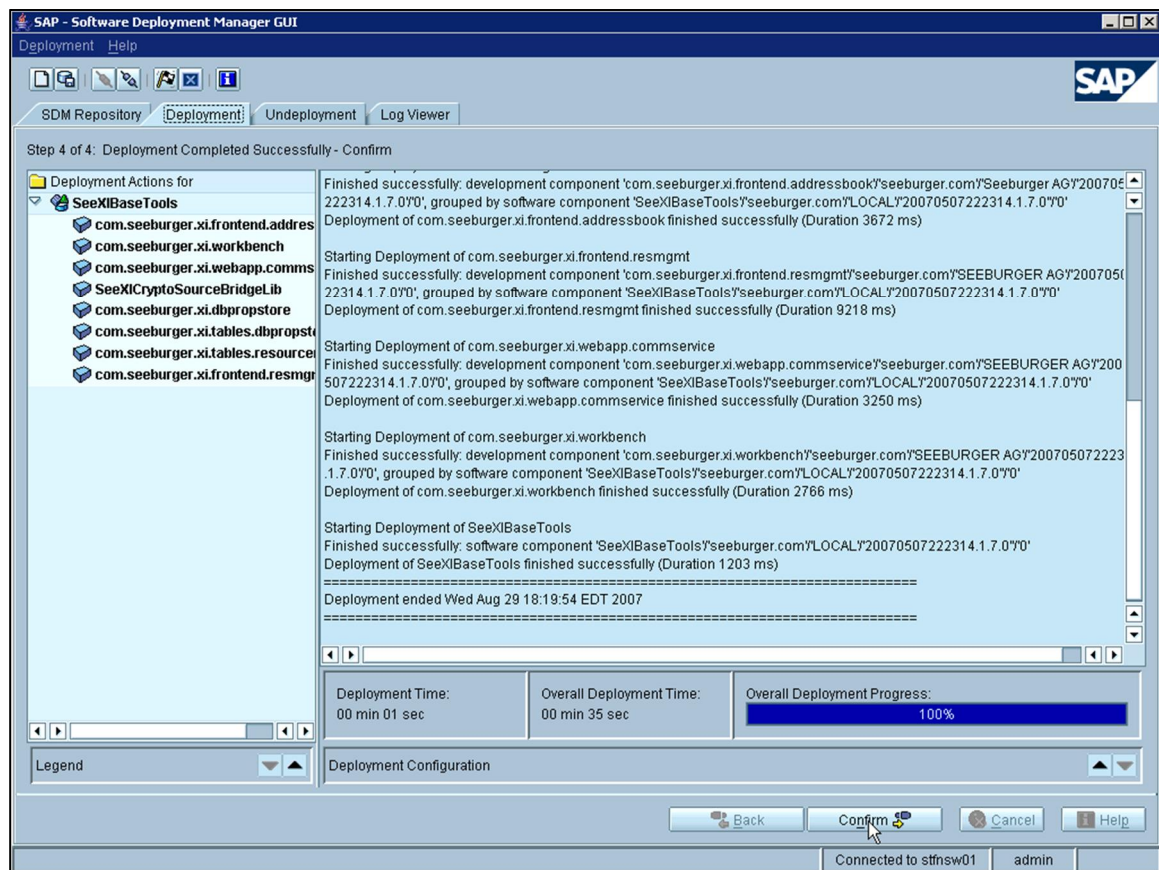


Fig. 6.2.2.1-3 Confirmando la distribución

Por último, se requiere detener el servidor SDM e iniciarlo nuevamente en modo integrado. Los pasos para completar la primera acción, son los siguientes:

- Cambiarse al directorio <Unidad>:\usr\sap\<SID>\<instancia\_central>\SDM\program
- Detener el servidor SDM con el comando StopServer.bat en Microsoft Windows, o StopServer.sh en Unix.

Después, se tiene que establecer el modo de inicio del servidor SDM en integrado, con el comando correspondiente para que el servidor SDM inicie también cuando el servidor de aplicaciones Java arranque:

```
sdm.bat jstartup "mode=integrated" (Microsoft Windows)
sdm.sh jstartup mode=integrated (Unix)
```

Finalmente, iniciar el servidor de aplicaciones Java.

### 6.2.2.2. Distribución en línea del SDM

Para llevar a cabo la distribución de una aplicación en línea, el servidor de aplicaciones Java, y por lo tanto, el SDM, deben ser iniciados.

Los pasos necesarios para llevar a cabo la distribución en línea incluyen el inicio del "SDM GUI" y la distribución de los archivos necesarios. El procedimiento es muy similar al de la sección anterior, distribución fuera de línea del SDM.

## **7. Evaluación técnica para un buen rendimiento y escalabilidad para los sistemas SAP en base a la plataforma ABAP**

Los sistemas empresariales deben de cumplir con estándares de calidad ya que son facilitadores para las empresas en la realización de sus procesos de negocio. En lo que refiere a los problemas de rendimiento, estos puede causar problemas de tipo: financieros (reportes de resultados de empresas que cotizan en bolsas de valores), logísticos (distribución en tiempo de productos o servicios), administrativos (cálculo de nómina para fin de año) o productivos (los empleados dedican mucho tiempo a recibir la respuesta del sistema).

El rendimiento es la forma en cómo se comporta un sistema con determinado tipo de carga de información. Este se mide siempre como tiempo de respuesta, capacidad de procesamiento y disponibilidad. La meta de tener un buen rendimiento es disminuir el ratio de costo/beneficio y aumentar las operaciones de manera escalable.

El rendimiento se ve afectado en general por los recursos disponibles en el sistema y que tan bien se usan y comparten esos recursos.

Los problemas descritos anteriormente causa mucha frustración dentro de la empresa y además posiblemente pérdidas monetarias. Es por eso que en base a nuestra experiencia consideramos que los administradores e ingenieros en computación deben de trabajar en conjunto en un plan de optimización continua y evitar en lo posible estas eventualidades o en caso de que ocurra poderlas resolver rápidamente.

En lo que refiere al plan de optimización continua, está altamente ligado con monitoreo de las variables de estado del sistema y la detección temprana de tendencias.

Los sistemas SAP en general son ambientes complejos y con altos niveles de integración con sistemas como SRM, CRM, BW, WM de SAP o alguna otra compañía. La evaluación aislada de un componente de software no es eficiente a menos que un análisis previo de causa raíz señale a ese componente de software como el responsable de los problemas de lentitud.

El análisis de causa raíz requiere herramientas que nos proporcionen los tiempos de respuesta del procesos de negocio por componente de software, para poder atacar las áreas con mayor margen de mejora.

El servidor de aplicación ABAP de SAP cuenta con varias herramientas especialmente diseñadas para ayudar a esta tarea de encontrar la causa raíz de los problemas de rendimiento y que además se integran con Willy Introscope de Computer Associates para el análisis de componentes con el servidor de aplicación de JAVA.

El resto del capítulo se cubrirán el uso de estas herramientas con algunos ejemplos, sin embargo cabe mencionar que como toda herramienta es necesario tener experiencia para poder utilizarla de la mejor manera.

A lo largo de este capítulo 7 se explicaran más a detalle las herramientas y los métodos que se pueden utilizar con base a la experiencia que hemos adquirido en nuestra vida laboral, además de discutir a fondo todo el proceso de optimización.

### **7.1. Plan para la optimización continua del rendimiento en el sistema**

El proceso de afinación de un sistema es un proceso continuo e iterativo, en el cual el monitoreo es una parte esencial. En general el equipo de ingenieros busca cambios en la configuración o actualizaciones en el software y basado en el monitoreo se puede apreciar si los cambios son positivos. Dependiendo de la aplicación se pueden tener diferentes requerimientos de rendimiento, pero se puede utilizar el siguiente proceso como guía general que se ha elaborado en base a experiencia:

1. Definir los objetivos de rendimiento.
2. Establecer los indicadores que reflejen directamente las mejoras a los usuarios finales.
3. Desarrollar y ejecutar un plan de mejora continua.

4. Continuamente analizar los resultados de monitoreo, para analizar cuales recursos necesitan mejora. Mantener una historia del sistema (historia médica).
5. Hacer un solo ajuste a la vez a medida de lo posible. Esto ayudara a ver reflejado los efectos de la afinación del sistema, sin embargo se puede llegar al punto de que no se pueda obtener mejores resultados con las mejoras realizadas en la base de datos y en SAP. Entonces tal vez sea necesario hacer una actualización de hardware.

La afinación de un sistema en general requiere no solo esfuerzo por parte de los ingenieros, sino que existe un reclamo de recursos, como por ejemplo aumentar el buffer de datos en una base de datos ayudará a evitar problemas por lecturas en disco, pero puede causar problemas en la disponibilidad de esa memoria para otros sistemas.

Dentro de este plan no se deben de olvidar los siguientes directrices [IBM\_PTG2011]:

- Los primeros esfuerzos producen mayores resultados. Existe una ley en economía de los rendimientos decrecientes, que refiere a una reducción progresiva en el margen de utilidad cuando la cantidad de un solo factor de producción aumenta y los demás permanecen constantes. Es por eso que una instalación limpia con parámetros adecuados, configuración de discos y "tablespaces" adecuada para evitar cuellos de botella en I/O y planeación del número de servidores de aplicación o poder de procesamiento es una parte muy importante de la fase de implementación del proyecto. Los cambios en etapas posteriores en general producen resultados cada vez menos visibles.
- Afinar una vez identificada la limitante. Si no se tiene bien delimitado el problema, los cambios que se hagan en la mayoría de los casos puede que no reflejen los resultados esperados. Afinar en donde solo donde se puede obtener el mayor resultado en el tiempo de respuesta.
- Hacer un solo cambio a la vez y tener un plan emergente para revertir el cambio, no siempre los cambios tienen el efecto positivo esperado y en ocasiones puede empeorar la situación.
- Medir y reconfigurar por capas. Por la misma razón que en el punto uno, solo se recomienda afinar una capa a la vez. Se recomienda separa el sistema en las siguientes capas:
  - Hardware
  - Sistema operativo
  - Servidor de aplicación
  - Manejador de base de datos
  - Sentencias de SQL
  - Programas de aplicación

El realizar la afinación de sistemas complejos genera una gran inversión de tiempo y recursos. Es por eso que es muy importante entender muy bien el problema antes de tomar medidas correctivas, ya que en ocasiones el proceso de negocio puede estar causando este problema sin posibilidad de mejora técnica por ejemplo facturar todos los meses del año el último día del año, cuando se puede hacer cada fin de mes o cuarto. Otro punto que se debe considerar en el proceso desafinación es la posibilidad de adquirir más o mejores recursos como medida adicional.

En los sistemas ABAP el monitor de carga en la transacción ST03 es la mejor herramienta para poder detectar aquellas transacciones, reportes e interfaces que se ejecutan. Este monitor nos da información de cuáles son los procesos, usuarios y sistemas que más consumen recursos con información importante que nos indica cómo se distribuye el tiempo de respuesta.

Una limitante que hasta el momento tiene esta herramienta, es que la cantidad de información histórica de información. Lo máximo es un par de meses lo que disminuye la capacidad de estudiar tendencias de un lapso de tiempo mayor, sin embargo los administradores pueden exportar esa información a archivos para su procesamiento posterior.

Es por eso que se recomienda a los administradores seguir dentro de una base de conocimiento o un cubo en un sistema de inteligencia de negocios generar y conservar esa información.

En la gráfica 7.1-1 podemos observar los tiempos de respuesta conservados para uno de nuestros clientes desde que inició sus operaciones productivas y que nos ha ayudado a establecer métricas

objetivas sobre el rendimiento generalizado de su sistema y sobre transacciones individuales evitando métricas subjetivas de usuarios.

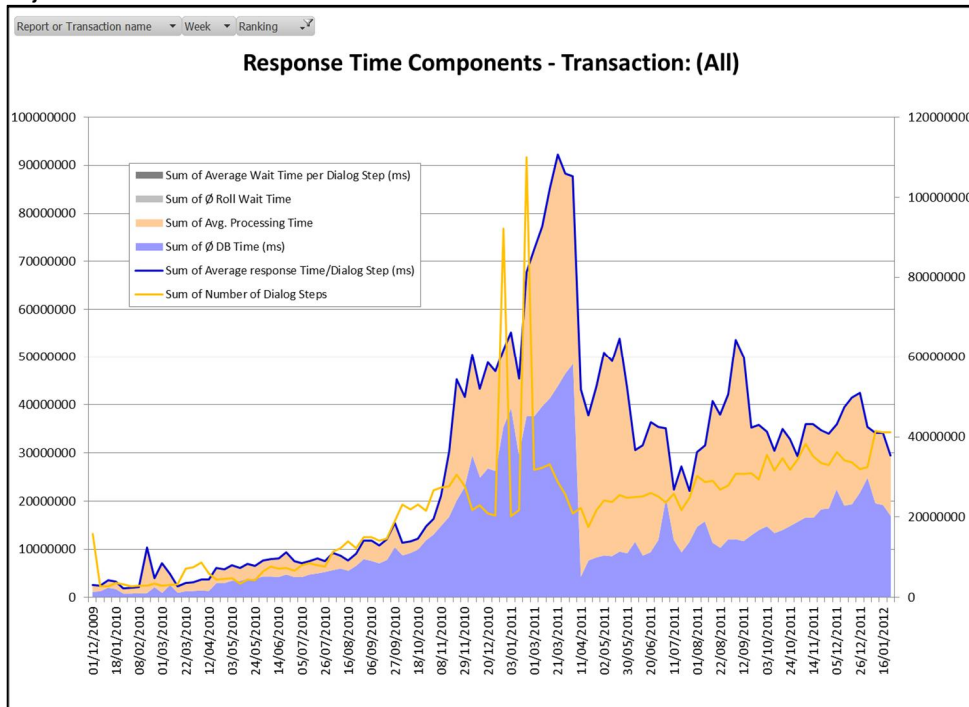


Fig. 7.1-1 Tiempo de respuesta para todas las transacciones de un ERP desde 2009.

## 7.2. Análisis del rendimiento

Para poder cumplir con las expectativas de los usuarios de un sistema SAP, es necesario tener un plan de optimización y monitoreo continuo. Este plan o estrategia tiene como fin mantener una solución, por lo cual el monitoreo debe cubrir muchos componentes individuales que en conjunto dan una solución al proceso de negocio.

En la figura 7.2-1 se conjunta los distintos elementos que se deben de considerar para asegurar el proceso de negocio cuenta con el soporte necesario para evitar problemas a su ejecución dentro del software. La figura 7.2-1 muestra la estructura general de una estrategia de monitoreo y optimización.



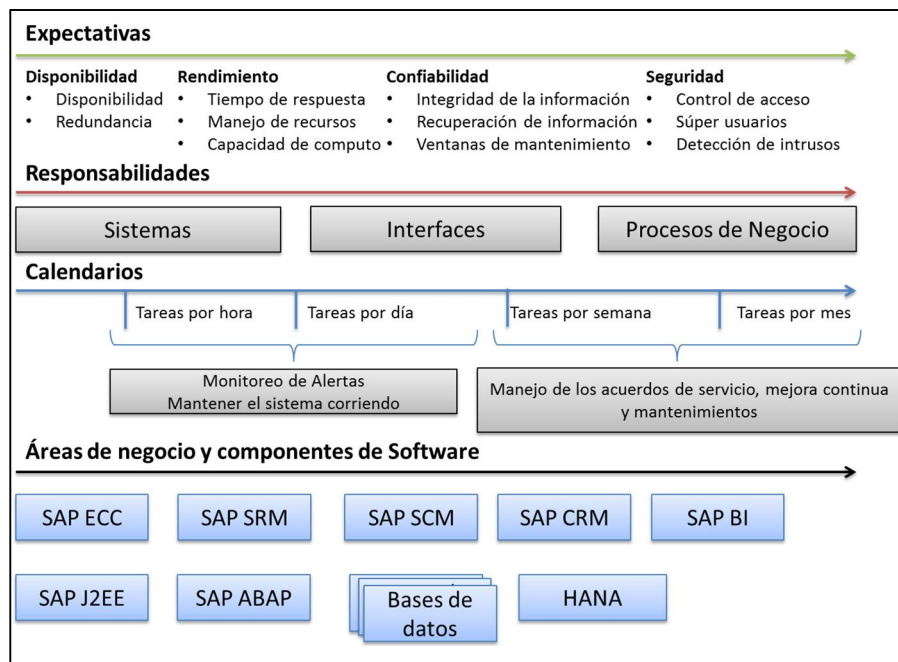


Fig. 7.2-1 Monitoreo de Soluciones

Entre los principales lineamientos y motivantes de estos análisis tienen que estar documentadas las expectativas de los usuarios, estas son:

1. La disponibilidad del servicio cuando se necesita
2. El tiempo de respuesta debe ser razonable
3. La aplicación debe ejecutarse correctamente (es decir que los productos se envíen correctamente de acuerdo a la orden de internet)
4. La aplicación es segura (nadie está manipulando tu información).

Dentro de estas expectativas de los usuarios, la que nos focalizaremos más en este trabajo está relacionado con el tema de rendimiento o tiempo de respuesta razonable y es por eso que teniendo como base el monitoreo adecuado, podemos asegurar en forma preventiva que se cumplan nuestras metas de servicio en tiempos de respuesta y capacidad de cómputo.

El tiempo de respuesta siempre se asocia con el tiempo de diálogo experimentado por los usuarios y es una de las principales causas de disgusto cuando no es adecuado para la tarea que realiza el usuario que trabaja con el sistema.

Por otro lado la capacidad de cómputo se asocia más a la capacidad de los trabajos de fondo de cumplir con los requerimientos de procesamiento en ventanas de ejecución predeterminadas por el negocio.

Como regla general siempre se deben separar los análisis para estos dos tipos de tareas y la carga balanceada de recursos es parte de las tareas de monitoreo. La prioridad para cualquiera de las tareas antes mencionadas está relacionada directamente con el tipo de negocio, por ejemplo en los sistemas bancarios el cierre de día es un proceso indispensable dentro del proceso por lo cual los trabajos de fondo son de muy alta criticidad.

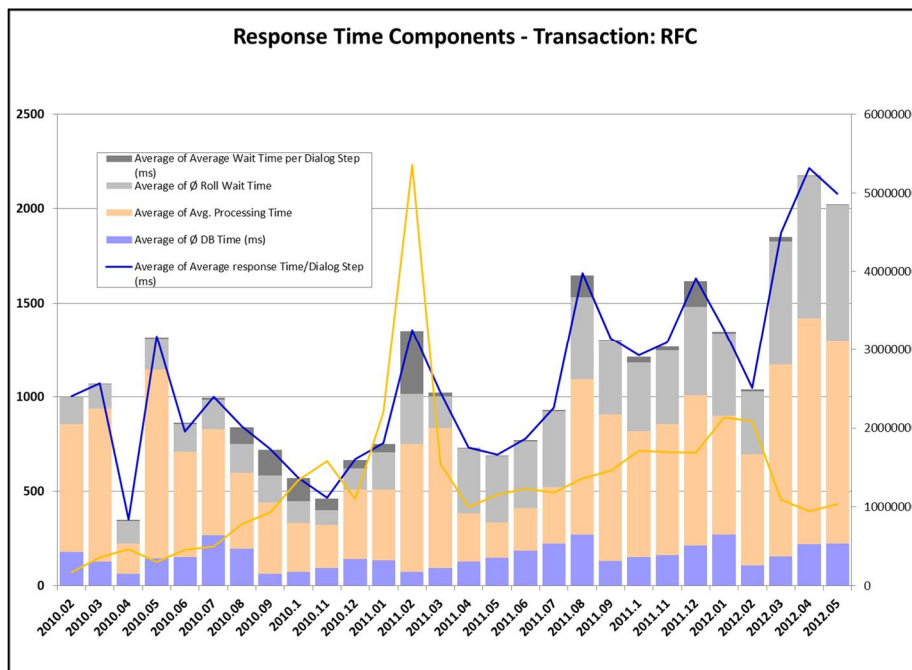
## Rendimiento

Usualmente los proveedores de infraestructura garantizan un tiempo promedio menor o igual a un segundo. Este promedio sin embargo puede no reflejar completamente los requerimientos del cliente y al ser un valor generalizado para cientos de transacciones provoca una sensación falsa de buen rendimiento.

Para analizar mejor un sistema en base a experiencia recomendamos hacer lo siguiente:

1. A partir de las definiciones de negocio, escoger de 10-30 transacciones críticas de las cuales su rendimiento se tiene que mantener bajo la lupa. Para poder hacer esa selección se puede seguir el criterio de daño económico, imagen de la compañía o si son ampliamente utilizadas en la compañía o por altos directivos.
2. Mide el tiempo de respuesta de estas transacciones en el sistema productivo por periodos de tiempo prolongados para guardar referencias de mejoras o pérdidas de rendimiento. Al mismo tiempo platica con los usuarios que utilizan dicha funcionalidad y familiarízate con la forma de ejecución y con su experiencia.
3. Presenta estos acuerdos de servicio en juntas semanales con los actores indicados
4. Agrega estas transacciones al CCMS (Central Computed Monitoring System)
5. Acuerda que si los valores de referencia se sobrepasan un análisis se llevara a cabo con las personas involucradas y que un plan de acción se generará para solucionar el problema con responsables y fechas compromiso.

En la figura 7.3-2 mostramos una gráfica generada conforme a la metodología antes descrita y que se presentan puntualmente el cliente en juntas semanales con sus distintos proveedores, la gráficas se generan con información obtenida de la transacción ST03 que se detallará más adelante y reflejan el comportamiento de por ejemplo la tarea RFC en el sistema a través del tiempo.



**Fig. 7.3-2 Rendimiento de la tarea de RFC en el tiempo**

Con base en esta información el análisis continuo y la optimización pueden comenzar. Las medidas de optimización se pueden categorizar en técnicas y funcionales.

Las técnicas pertenecen a los componentes generales del sistema que ayudan a su buen desempeño como son parámetros de SAP, base de datos o sistema operativo; configuraciones de discos; redes; o problemas de CPU y memoria.

Las funcionales se ejecutan a nivel de los programas, en donde se debe verificar que el uso de recursos sea óptimo y no cause problemas generalizados.

En base en esto podemos clasificar el análisis de un sistema en varios frentes. Por ejemplo si se tiene un problema de rendimiento es preciso conocer si es general o solo puntual a una transacción. En los

siguientes capítulos explicaremos más a detalle este análisis y las herramientas necesarias para ejecutarlo.

### 7.2.1. Problemas de desempeño generalizados

Existen dos formas de detectar problemas generalizados en los sistemas, uno es mediante una insatisfacción de uso por el rendimiento del sistema (método reactivo) y el segundo que es mediante el correcto monitoreo, obtención de datos y de tendencias (método preventivo).

Método reactivo.

Los usuarios son los primeros en reportar problemas generalizados de rendimiento, para poder confirmar estas observaciones, podemos hacer las siguientes verificaciones en el caso de los usuarios o tareas de dialogo:

- ¿Es el tiempo del despachador mucho mayor a 50 milisegundos? Este problema puede ser causado por programas muy lentos o que están bloqueando el proceso por mucho tiempo. Es posible también que existan pocos procesos de dialogo para la carga actual del sistema.
- ¿Es el tiempo de base de datos en dialogo mucho mayor a 40% del tiempo de respuesta? La base de datos al ser un recurso centralizado afecta a la mayoría de las transacciones. Si la mayoría presenta este síntoma es necesario investigar el rendimiento de la base de datos.
- ¿Es el tiempo de procesamiento mayor a dos veces el tiempo de CPU? Si el tiempo de procesamiento es mucho mayor al tiempo de CPU puede ser un indicador de problemas en la capacidad de CPU. En procesadores nuevos como los POWER7 de IBM esta métrica no es representativa como nos ha demostrado la experiencia.
- ¿Es el tiempo promedio de las tareas de dialogo mayor que en días, semanas o meses pasados? Si es así es necesario investigar los cambios realizados en el sistema antes de que se presente la degradación del tiempo de respuesta y analizar donde se está consumiendo este tiempo extra.
- ¿Es el problema temporal o permanente? Es importante si el problema se presenta solo en periodos de tiempo específicos o si el problema dura todo el día. Desde la transacción ST03N podemos ver el perfil de tiempo de la carga del sistema e identificar si el tiempo promedio se ve afectado en las horas pico.
- ¿Existen recursos de hardware al límite en este periodo de tiempo? Es posible encontrar problemas generalizados por la capacidad de hardware como alto consumo de CPU o memoria.

Método preventivo.

En este caso contamos con herramientas para la recolección de información. Estas herramientas se deben explotar y realizar análisis detallados de tendencias en el rendimiento del sistema. Algunas de las tareas que se deben ejecutar por parte de los administradores son:

- Revisar los reportes de rendimiento automáticos generados por el Solution Manager, tales como el EarlyWatch Alert que cuenta con las métricas básicas de rendimiento semanal.
- Conservar estadísticas del consumo de CPU, memoria, espacio en disco, rendimiento del I/O y estadísticas de red.
- Crear tableros de monitoreo centralizados para poder detectar afectaciones generalizadas en el ambiente.
- Crear métodos de análisis mensuales para la identificación de tendencias negativas en los procesos más importantes.
- Tener niveles de servicios definidos con el área de negocios y no considerar solo los promedios de tiempos de respuesta.

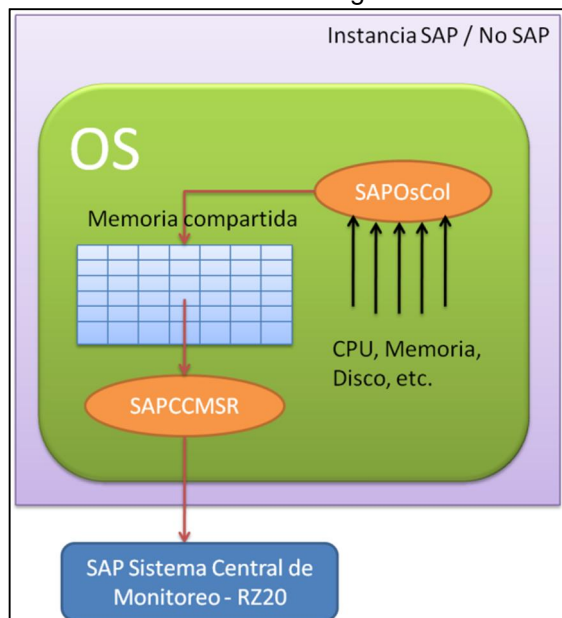
#### 7.2.1.1. Monitoreo de Hardware (ST06n)

##### SAPOsCol

Si recordamos la arquitectura de las instalaciones de un servidor de aplicación SAP ABAP, podemos identificar varios componentes de software administrados por un sistema operativo que provee recursos que se consumen como CPU, memoria física, memoria virtual, Manejo del sistema de archivos, discos físicos y red.

Problemas de rendimiento en cualquiera de estos elementos representa un problema de rendimiento para SAP y por lo tanto SAP cuenta con una herramienta que puede ayudar a localizar la fuente del problema a nivel sistema operativo.

SAP utiliza el colector de información llamado SAPOsCol para monitorear el consumo de estos recursos a nivel sistema operativo. SAPOsCol es un programa único que es dependiente de la plataforma y que corre en fondo en el sistema operativo colectando información del sistema utilizando un segmento de memoria compartida y transmitiendo esa información para su visualización y manejo al sistema central de monitoreo como se muestra en la figura 7.2.1.1-1:



**Fig. 7.2.1.1-1 Arquitectura de Monitoreo**

El servicio de “SAPOsCol” permite que la información colectada sea colocada en el segmento de memoria disponible para ya sea un agente de CCMS o un proceso de dialogo de una instancia de SAP. La información se puede desplegar en el sistema de monitoreo central o desde el monitor de sistema operativo de SAP (ST06, ST06N, OS07).

Para sistemas SAP ABAP, el SAPOSCOL es parte del kernel de SAP. El programa se puede encontrar en la ruta `/usr/sap/<SID>/DVBEMGS00/exe` a nivel sistema operativo y se puede controlar a nivel sistema operativo usando el comando `saposcol`. Para sistemas no-SAP se puede bajar el programa del centro de distribución de software e instalar para monitorear instancias como pueden ser un servidor de base de datos.

Los sistemas operativos que están soportados son los siguientes [SHOSC2011]:

- Microsoft Windows
- AIX
- SUN/SOLARIS
- HP-UX
- LINUX
- OS/390
- OS/400

Para sistemas virtuales SAP ocupa un agente llamado SAPHOSTAGENT que es el sucesor del SAPOsCol para versiones a partir de la 7.11 en el kernel de SAP con el propósito de habilitar el monitoreo en sistemas virtualizados y en la nube. Estas nuevas tecnologías necesitan de monitores especiales, pues la asignación de recursos es dinámica y por tanto el marco de referencia para las mediciones también lo es.

El programa SAPOsCol tiene un bajo consumo de recursos ya que desde el principio fue diseñado para no interferir lo menos con los recursos disponibles.

Colecta información de manera automática cada 10 segundos. Esa información se agrega cada hora y se guarda por las últimas 24 horas. Extrae información en tiempo real del sistema, aunque se tiene que renovar la pantalla de manera manual.

Los parámetros de rendimiento que podemos obtener son los siguientes:

- Promedio de la carga y utilización del CPU
- Utilización de Memoria
- Paginación de y hacia la memoria
- Información de la utilización de Disco (Depende de la configuración de discos)
- Actividad LAN (Limitado)
- Parámetros de configuración a nivel sistema operativo.

### Visualización con las transacciones ST06, ST06N y OS07 para monitoreo de Sistema Operativo

A continuación se mostrará un ejemplo de como las transacciones de monitoreo del sistema operativo ayuda a analizar problemas de hardware y del sistema operativo. Para iniciar estas herramientas se puede ejecutar la transacción ST06 o ST06N.

La pantalla del monitor del sistema operativo está dividido en tres áreas (figura 7.2.1.1-2):

1. Se puede visualizar la lista de computadoras que están siendo monitoreadas y se selecciona aquella para la cual se quiere hacer el análisis.
2. En esta ventana selecciona la información que deseas analizar y el periodo.
3. Se despliega la información del servidor y el análisis que se escogieron.

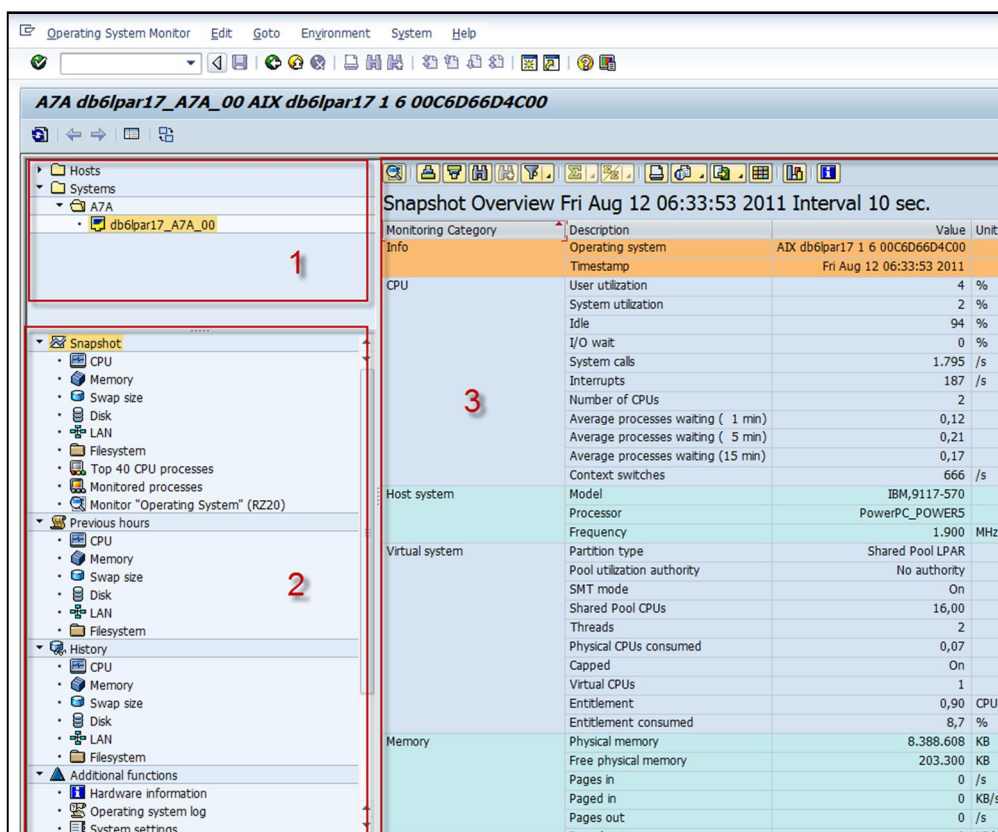


Fig. 7.2.1.1-2 Transacción ST06N – Monitoreo de Hardware

En base a estas tres áreas podemos empezar el análisis de información del sistema operativo. El primer paso es escoger el host para el cual queremos analizar la información. En seguida se escoge el periodo para el cual queremos analizar la información y finalmente vemos las diferentes vistas para ver cuáles temas nos interesa.

Los monitores que tenemos son los siguientes:

## CPU

La información que se despliega es por cada CPU en el cliente, y está separado en porcentajes por:

- Usuario
- Sistema
- Ausente que es el porcentaje en el cual es CPU no está realizando actividad o está en espera de una operación de entrada o salida.
- Espera de I/O. Que en realidad simplemente se puede considerar como tiempo ausente, pues no siempre está relacionado con I/O.

CPU			
	User utilization	2	%
	System utilization	1	%
	Idle	96	%
	I/O wait	1	%
	System calls	1.262	/s
	Interrupts	246	/s
	Number of CPUs	2	
	Average processes waiting ( 1 min)	0,07	
	Average processes waiting ( 5 min)	0,11	
	Average processes waiting (15 min)	0,11	
	Context switches	560	/s

**Fig. 7.2.1.1-3 Información de CPU**

Existe una variedad de factores que pueden influir el consumo excesivo de CPU y es por eso que se debe hacer un análisis detallado. Si el problema es el exceso de procesos en el servidor, se puede trasladar esos procesos a algún otro servidor para liberar carga. También se pueden incrementar el número de CPU en el host como remedio.

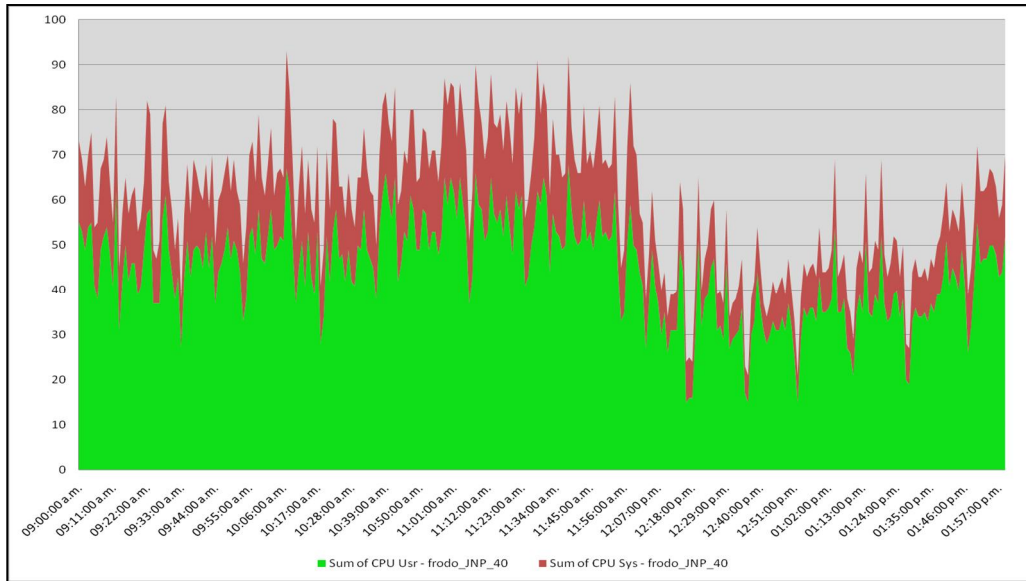
Algunos otros valores de importancia son el número de CPU por servidor (Solo en la vista instantánea), interrupciones por segundo, llamadas del sistema por minuto, cambios de contexto por minuto y el promedio de procesos esperando CPU hace un minuto, cinco y quince minutos.

De estos solo el promedio de procesos nos puede ayudar a detectar problemas de rendimiento en el CPU. Este número lo que indica es el número de procesos por cada CPU que están en la cola de CPU antes de ser asignados a un CPU disponible. Mientras se mantenga un promedio de 1 proceso en espera por cada CPU disponible, los recursos se pueden decir que son suficientes. Con promedios mayores a tres procesos en espera, un posible cuello de botella se puede empezar a investigar.

1. Con un alto consumo de CPU, un promedio alto puede indicar que existe una gran cantidad de procesos activos en el servidor.
2. Con un consumo bajo de CPU, un promedio alto puede indicar que la memoria no es suficiente y que las esperas son por la paginación en memoria.

Si el CPU tiene un consumo mayor a 75% es necesario investigar más a fondo, si la carga en todos los servidores del sistema es proporcional o si existen servidores con menor carga de CPU. Existen casos donde un solo servidor absorbe todo el trabajo mientras con consumos de 85-90% de consumo de CPU mientras otros tienen solo 30%. Si este es el caso lo mejor es configurar y habilitar correctamente los grupos de acceso, además de la distribución adecuada de trabajos de fondo y si es necesario paralelizar tareas utilizando código.

En la figura 7.2.1.1-4 podemos observar el consumo de CPU de un servidor con alto consumo. El CPU está diagramado en forma agregada con el CPU que consume el sistema (Sistema Operativo) y el usuario (Aplicaciones como Oracle o SAP). Si entramos a los procesos con mayor consumo, podemos observar cuáles de ellos son los causantes del alto consumo de CPU y a partir de ahí podemos seguir investigando. En la figura 7.2.1.1-4 se muestra procesos de la aplicación de SAP Java, SAP ABAP y de Oracle, lo cual puede indicar que en ese instante el sistema de ejemplo mostrado XNP está realizando un trabajo pesado que se puede investigar más en la aplicación de SAP.



**Fig. 7.2.1.1-4 Información de CPU con alta frecuencia en periodo pico.**

Utilizando la vista instantánea, horas previas y la historia podemos analizar más a fondo el comportamiento del servidor.

**Vista Instantánea:**

Podemos obtener información acerca del sistema operativo y características de HW como la siguiente información que incluye tipos de CPU, frecuencias de funcionamiento, información de memoria, tipo de sistema operativo y su versión.

```

Hostid: 82aa00ef
Release: 5.9
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain:
Kernel version: SunOS 5.9 Generic 122300-56 Oct 2010
OpenWindows version:
Solaris X11 Version 6.6.1 11 June 2008
    
```

```

System Configuration: Sun Microsystems sun4u Sun Fire E25K
System clock frequency: 150 MHz
Memory size: 94208 Megabytes
    
```

```

===== CPUs =====
Slot ID   CPU ID   Run MHz   E$ MB   CPU Impl.   CPU Mask
-----
/SB00/P0  0, 4    1950    32.0  US-IV+    2.4
/SB00/P1  1, 5    1950    32.0  US-IV+    2.4
/SB00/P2  2, 6    1950    32.0  US-IV+    2.4
    
```

Se pueden también obtener los procesos que más consumen CPU utilizando la vista de procesos con mayor consumo como se muestra en la figura 7.2.1.1-5.

Process I...	User	Process Name	Utilizatr	CPU Time	Resident...	Priority
20.345	xnpadm	jlaunch pf=/usr/sap	58,84	8504:01	2.576.384	0
23.579		dw.sapXNP_DVEBMGS60	53,33	081:58	249.080	0
4.048		dw.sapXNP_DVEBMGS60	52,85	015:18	229.016	0
25.538		dw.sapXNP_DVEBMGS60	52,27	045:08	247.632	0
17.219		dw.sapXNP_DVEBMGS60	50,91	054:01	209.256	0
25.356		dw.sapXNP_DVEBMGS60	50,33	083:58	255.848	0
28.794		jlaunch pf=/usr/sap	28,21	8262:11	2.455.056	59
26.167		dw.sapXNP_DVEBMGS60	21,44	037:21	264.208	2
894	tnpadm	jlaunch pf=/usr/sap	14,78	1348:54	2.608.128	59
3	root	fsflush	19,90	12418:5	0	60
4.050	oraxnp	oracleXNP (DESCRIPT	10,53	002:56	3.686.816	59
25.358		oracleXNP (DESCRIPT	9,66	014:05	3.687.248	59
25.542		oracleXNP (DESCRIPT	9,56	006:56	3.687.584	40
23.581		oracleXNP (DESCRIPT	8,50	013:53	3.687.488	59
2.023		oracleXNP (DESCRIPT	7,72	045:19	3.691.080	59

**Fig. 7.2.1.1-5 Mayor consumidor de recursos de CPU**

Teniendo los procesos con mayor consumo los podemos mapear de identificador de proceso con un proceso de diálogo en SAP con la transacción SM66 o SM50 y de ser necesario un se puede obtener más detalle y hacer una traza con la transacción SM30.

En el caso de que los procesos con más consumo sean de base de datos del tipo "oracleSID" para Oracle o "agentSID" para DB2, se puede hacer igual un mapea del proceso con el agente de la base de datos y ver que tareas está ejecutando. Esta información se puede obtener en la transacción ST04 en las sesiones de base de datos y de ser necesario se puede hacer una traza con la transacción ST05.

Si es otro proceso distinto a la base de datos o la aplicación de SAP, se debe de investigar su origen. En ocasiones existen monitores externos que causan problemas a las aplicaciones que monitorean.

### Historia y horas previas

Utilizando la historia podemos crear gráficas de capacidad y de consumo como las figura 7.2.1.1-6. Esta información esta agregada en promedios por hora, por lo cual los picos puede que sean pasados por alto.

Con la historia podemos graficar y entender gráficamente las horas de mayor consumo de CPU. Así como el historial y el efecto de cambios. En la gráfica notamos claramente el beneficio de agregar un par de CPUs al servidor reduciendo en promedio 30% de la utilización total de CPU, ver figura 7.2.1.1-6.



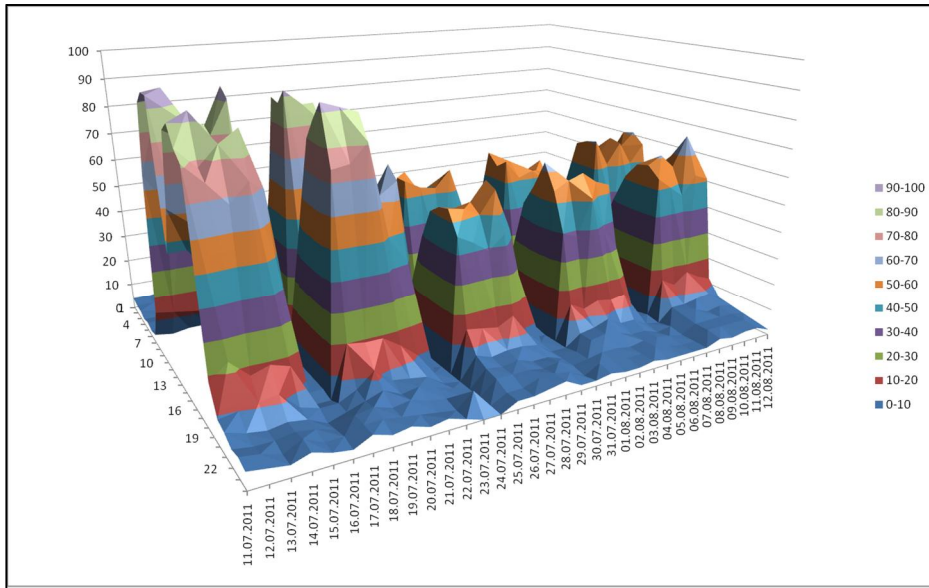


Fig. 7.2.1.1-6 Consumo de CPU histórico por horas

En esta otra gráfica 7.2.1.1-7 podemos observar claramente que de 00:00 a 06:00 y de 19:00 a 23:00 horas la carga del servidor es prácticamente nula, lo cual muestra un patrón claro de utilización de usuarios y no de trabajos de fondo. Esto pone en perspectiva que la mejora se debe enfocar en las transacciones de dialogo que corren en ese horario pico.

Observamos que el trabajo es todos los días de la semana y el comportamiento es muy similar a lo largo de las semanas.

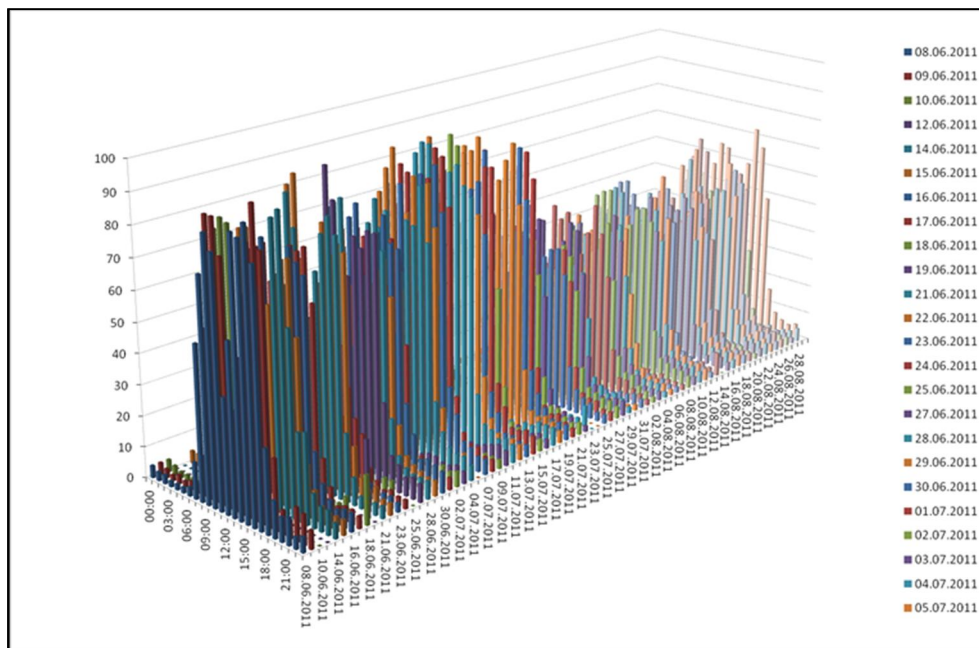


Fig. 7.2.1.1-7 Consumo de CPU histórico por horas

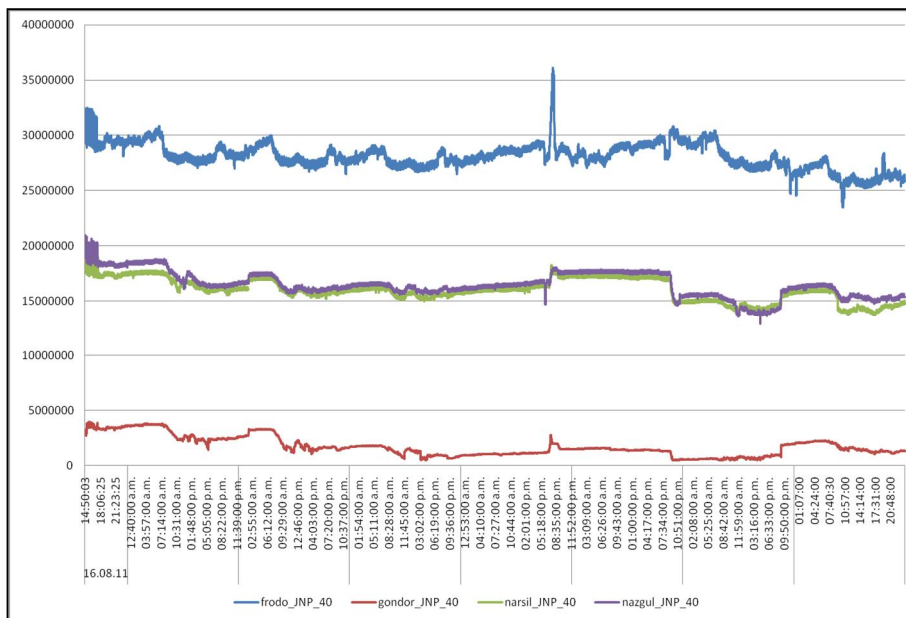
**Memoria:**

La información más importante sobre memorias es:

- Memoria física libre en el servidor
- Tasa de paginación (intercambio de páginas entre la memoria principal y el archivo de paginación en el disco). En caso de observar un gran número de páginas que salen o entran al archivo de paginación, se tiene un problema de recursos de memoria física. Para sistemas

UNIX la paginación hacia fuera es la que hace sentido y en sistemas Windows es las páginas que entran en Kilobytes. [ST2010]

En la gráfica de la figura 7.2.1.1-8 podemos observar la cantidad de memoria disponible en cada uno de los servidores y el consumo a través de un período de una semana. Se observa claramente que el servidor representado por la línea roja se debe monitorear más de cerca para ver si la falta de memoria libre empieza a afectar su rendimiento y crear paginación.



**Fig. 7.2.1.1-8 Monitoreo de la memoria libre.**

Es por eso que concluimos que la distribución de usuarios en este sistema en particular, como se muestra en la gráfica 7.2.1.-8, no está balanceada, considerando que todos los servidores cuentan con la misma cantidad de memoria.

Una vez que identificamos poca memoria libre es buena idea verificar si el ratio de paginación por hora es mayor 20% de la memoria física. Si es así tal vez buena idea empezar a redistribuir procesos de trabajo de SAP. Hay que recordar que la asignación de memoria en SAP es un tanto estática y está estrechamente ligada con el número de procesos definidos y la configuración por proceso.

Otra transacción donde podemos indagar más sobre el consumo por usuario es la SM04 donde nos muestra el consumo por usuario en memoria extendida de SAP, de roll y privada.

Al ser el proceso de trabajo en SAP una máquina virtual, el colector de basura puede tardar un poco para liberar la memoria que un usuario ha consumido y puede confundir si no se analiza con un período mayor de tiempo.

### 7.2.1.2. Monitoreo de la Base de Datos (DBACOCKPIT)

El DBACOCKPIT es como su nombre lo dice en la lengua inglesa un panel de control centralizado para las operaciones de administración y monitoreo de la base de datos. Es una interfaz gráfica que cubre todas las principales acciones del manejo de una o varias bases de datos ver figura 7.2.1.2-1.

Para entrar a esta transacción solo se manda llamar con el código de transacción DBACOCKPIT y aparece la pantalla principal.

Dentro de la pantalla principal tenemos varios paneles, cada uno con enfocado a facilitar la administración y monitoreo, además que conjunta varias transacciones individuales utilizadas anteriormente como la DB02, DB01, DB24, ST04, DB12 y DB13 pero con la incomodidad de la casi nula integración entre ellas.

En la figura 7.2.1.2-1 observamos cinco áreas distintivas:

1. La Barra de herramientas del ambiente. Es un punto centralizado para manejar todo el conjunto de tus sistemas de base de datos. Desde aquí se puede agregar bases de datos a monitorear o administrar, manejar sus conexiones y te permite escoger el sistema que se desea analizar. En la figura observamos un sistema con RAC de Oracle y es por eso que observamos dos instancias de base de datos.
2. Información Central. Esta área es común en la mayoría de las acciones por las que navegamos y lo único que ayuda es a ver información sobre el inicio de la base de datos, el nombre de la base de datos y cuando fue el último "refresh".
3. Panel de navegación. Tiene una estructura de árbol dividida en la parte superior en las tareas principales de administración de una base de datos. Estas tareas son por ejemplo, monitoreo del rendimiento, manejo de espacio y manejo de trabajos de fondo. Cada una de estas tareas cuenta a su vez con nodos de acción.
4. Área de acción. Despliega los detalles del nodo de acción que se ha seleccionado.
5. Área de Mensajes. Este es una plataforma de mensajes más robusta que la tradicional del SAP GUI donde se puede ver la historia y tener un mejor manejo de las alertas de la sesión de trabajo.

Algunas de las actividades que podemos realizar son el navegar en diferentes acciones manteniendo la información, lo cual permite una mejor navegación. También permite un manejo centralizado para ambientes complejos con muchas bases de datos y monitoreo remoto de bases de datos sin el servidor de aplicación de ABAP.

Para agregar un sistema se puede hacer con alguna de las siguientes opciones:

- Entrada directa usando el DBACOCKPIT -> add system entry
- Importar automáticamente con el SLD usando el DBACOCKPIT
- Utilizando el paso a paso del SAP Solution Manager Work Center

Para más información se puede consultar [TT2012].

The screenshot shows the 'PRD: Performance - Performance Overview' window. It features a navigation tree on the left, a central data table, and a right-hand pane with detailed metrics. Red boxes highlight the five key areas mentioned in the text.

DB Name	Started	DB Server	DB Release
PRD	19.11.2011		10.2.0.4.0

DB instance	Total	Day, Time
DB instance	All	00:00:00
DB node	All	Start up at 00:00:00
DB release	10.2.0.4.0	Sec. since start 0
		Sec. btw. snaps. 0

Data Buffer			
Size (kB)	4,521,984	Logical reads	9,510,781,065
Quality (%)	88.2	Physical reads	1,087,668,531
Size default pool (kB)	4,521,984	Physical writes	66,025,291
Size keep pool (kB)	0	Buffer busy waits	2,923,088
Size others (kB)	0	Buffer wait time (s)	7,962

Shared pool		Log buffer	
Size (kB)	9,043,968	Size (kB)	2,080
DD-cache Quality (%)	98.9	Entries	404,735,076
SQL area getratio(%)	99.2	Allocation retries	26,998
SQL area pinratio(%)	100.0	Alloc fault rate(%)	0.0
SQLA.Reloads/pins(%)	0.0003	Redo log wait (s)	313
		Log files (n use)	32 ( 32 )

Calls			
User calls	508,941,413	Recursive calls	356,329,985

Fig. 7.2.2-1 Transacción DBACOCKPIT

Dentro de esta transacción como lo mencionamos anteriormente se puede planear las tareas administrativas de una base de datos. La transacción cuenta con un calendario donde podemos calendarizar y programar actividades como archivar los archivos de log, reorganizar segmentos, actualizar estadísticas, respaldar nuestra base de datos y sus "logs" o bitácora de transacciones, revisar inconsistencias e incluso inicializar cintas de respaldo.

Hacer todo esto tiene una ventaja práctica que permite hacerlo sin invocar las herramientas propias de cada proveedor lo cual da cierta flexibilidad y comodidad, no obstante aun así es necesario tener personal capacitado en la base de datos que pueda confirmar que las tareas de respaldo principalmente se están llevando adecuadamente.

Hacer un análisis también del crecimiento de los segmentos de Oracle es sencillo y podemos revisar un histórico para tener la planeación correcta de disco o espacio en nuestra SAN. Los datos que podemos obtener directamente de la DBACOCKPIT son el tamaño aproximado actual de la base de datos y que tan llena esta, los ritmos de crecimiento de segmentos, objetos que ocupan cantidades críticas de espacio.

En la parte de monitoreo dentro del DBACOCKPIT se integró la transacción ST04 desde donde podemos monitorear varias aspectos relevantes al rendimiento de la base de datos. Este es el punto principal de partida para poder empezar el análisis de rendimiento de la base de datos dentro de un sistema SAP.

En la venta del panel de navegación de la transacción podemos ver la rama referente a rendimiento y si abrimos el "Performance Overview" nos despliega información general de la salud de la base de datos, pero sin darnos detalles suficientes para juzgar por completo su rendimiento.

Podemos encontrar la calidad del uso de los buffers, el consumo de CPU, promedio de lecturas físicas por llamadas, el cache de SQL, la peticiones SQL, las sesiones actuales, lo bloqueos y uso de los recursos actuales.

Las siguientes son funciones vitales para el monitoreo:

- Revisión general de los buffers de la base de datos (hay que tener cuidado ya que una lectura con valores aceptables puede ser poco asertiva si no se contextualiza adecuadamente). El buffer más importante es el buffer de datos del cual se obtiene toda la información que las aplicaciones de SAP requieren, o al menos gran parte. La calidad de este buffer nos muestra que tanta información se obtiene directamente del buffer sin necesidad de acceder a lecturas en disco.
- Revisión de las operaciones que está ejecutando en el momento la base de datos, en especial las sentencias SQL. Estas pueden tener un efecto generalizado en el rendimiento del sistema y causar problemas con otros procesos de negocios que estén en el mismo servidor de aplicación al desplazar la información fuera del buffer de datos. Es muy común entrar sistemas en los cuales no más de diez sentencias de SQL causan el 30-40% de carga total de la base de datos.
- Revisión de las estadísticas de las sentencias SQL y sus costos respectivos.
- Revisión de la distribución de lectura/escritura para los tablespaces para distinguir puntos calientes.
- Revisión de las situaciones de bloqueos. Revisar cuales son los reporte o sesiones que se están bloqueando.
- Revisión de la situación de los índices y las estadísticas del optimizador.
- Revisión de errores en el log de la base de datos y la situación actual de los parámetros de la base de datos.

#### **7.2.1.2.1. Análisis de tiempos de espera en Oracle [FM2009] [FM619188]**

A partir de la versión 10g de Oracle se puede estudiar un problema de lentitud de una consulta aun cuando ésta ya no exista, cosa que en la versión 9i era imposible.

Para medir que tantos datos obtenemos de memoria, Oracle tiene un indicador llamado: "Buffer cache ratio" que indica porcentaje de numero de bloques que se pueden acceder desde el buffer de la instancia de Oracle sin necesidad de acceso al disco, sin embargo las estadísticas internas de Oracle no reflejan que esta métrica ayude siempre a clasificar a un sistema de base de datos como un sistema de buen desempeño. Este indicador se puede observar en la transacción ST04 (DBACockpit) de SAP y se recomienda que tenga un valor mayor iguala a 94%, sin embargo aun cuando un sistema tenga valores aún mayores a 94% no significa que no existan problemas de rendimiento en la base de datos, de hecho hay ocasiones que una sentencia SQL costosa que se ejecuta frecuentemente este causando Hit Ratio de 99%, pero el sistemas este en muy malas

condiciones y el resto de sentencias tardan demasiado tiempo porque son desplazadas fuera del buffer y entonces tenemos mayor número de accesos a disco.

Es por eso que necesitamos más que un simple indicar para analizar y mejorar el rendimiento de una base de datos.

El principio básico de una base de datos es un buen diseño de la misma y un acceso optimizado a esa información. No es parte de este análisis el diseño de la base de datos, ya que la mayoría de los sistemas SAP requieren poca intervención en este sentido, por lo cual el enfoque principal es tener accesos personalizados a los clientes que sean efectivos.

El factor más importante dentro del rendimiento del sistema ERP de SAP es el tiempo que le toma ejecutar cierto trabajo y entregar un resultado. No importa si la calidad de los buffers o algunos otros indicadores muestren aspecto positivos si el tiempo de respuesta esperado no es el adecuado. Es por eso que el tiempo de respuesta debe ser el factor motivante de nuestro análisis.

Dentro de DBACOCKPIT tenemos una rama del árbol llamada "Wait Event Analysis" que nos permite de manera eficiente identificar en que parte de los procesos de ejecución de tareas se están consumiendo el tiempo.

Para analizar estos procesos que aún están siendo ejecutados tenemos una vista que muestra las sesiones actuales de Oracle a nivel de procesos activos dentro del manejador. Este monitor despliega las sentencias SQL, que proceso de trabajo de SAP la solicitó y una columna vital llamada "evento" que despliega la situación actual del proceso o evento en el que está el proceso. Algunos de los eventos más importantes son:

- CPU. El cual indica que la base de datos está ejecutando en si alguna acción como por ejemplo un ordenamiento.
- DB File Sequential Read. Indica que la base de datos está esperando por un bloque de disco.
- SQL\*Net. Generalmente indica la transferencia de datos desde el proceso de Oracle al de SAP o viceversa.
- Enq:TX. Contención a nivel de fila, lo cual indica que el proceso de Oracle está esperando por un bloque.

Información estadística sobre estos eventos se puede encontrar también en la vista de los eventos del sistema, donde se muestra una estadística de cuál es el evento que más consumo tiempo dentro de la base de datos y el cual nos puede ayudar a identificar problemas generalizados de performance como IO.

CPU:

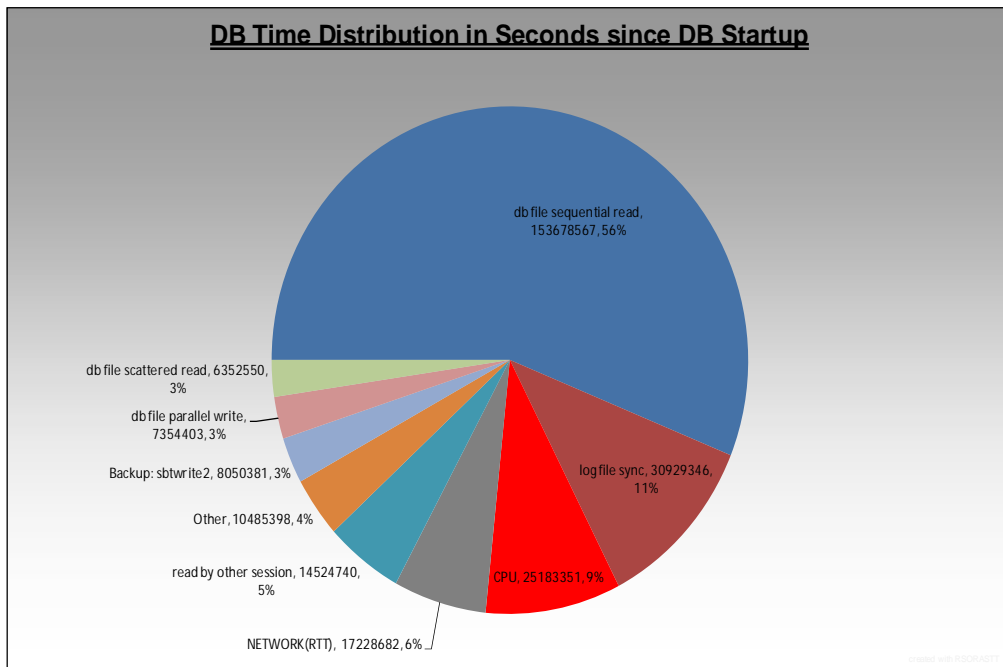
Para el evento de CPU, el significado es que el motor de base de datos realmente está trabajando en el procesador y memoria, lo cual de cierta manera es un signo positivo.

El método de afinación para una base de datos con este comportamiento es optimizar las sentencias de SQL que consumen mucho CPU o que tienen muchas lecturas a búfer o conseguir procesadores más veloces.

IO:

La parte de IO suele ser más problemática y los métodos de afinación son más complejos y requieren un análisis más detallado sobre varios componentes.

En la siguiente gráfica podemos observar de acuerdo al porcentaje que ocupa un evento, el potencial de optimización y lo más importante en referente al tiempo de respuesta es donde se está tomando la mayoría del tiempo en la base de datos.



**Fig. 7.2.1.2.1-1 Distribución del tiempo en eventos de Oracle**

La gráfica 7.2.1.2.1-1 nos da un ejemplo muy claro que el evento “db file sequential read” ocupa casi un 56% del tiempo total de respuesta lo cual significa que se gasta mucho tiempo leyendo cada bloque del disco. La forma en cómo se puede optimizar son las siguientes:

- Tener un búfer de datos con mayor capacidad con el objetivo de evitar ir y a disco teniendo más bloques en memoria.
- Optimizar las sentencias que provocan un nivel alto de I/O (Aquellas que piden demasiados datos) y que se pueden obtener con las estadísticas del kernel y ordenarlas por el número de lecturas a disco.
- Finalmente se tienen que hacer una afinación a nivel de disco para que los tiempos de respuesta en la capa de persistencia disminuyan, si es que se ha detectado tiempos altos de respuesta de la capa almacenamiento (SAN o discos).

Otra gráfica interesante que se puede generar es un histograma, ver gráfica 7.2.1.2.1-2, para identificar problemas en la escritura del log. En el histograma tenemos la frecuencia del porcentaje de espera y el porcentaje de tiempo acumulado. Es decir que entre mayor densidad hacia la derecha de la gráfica mayor es el tiempo de espera.

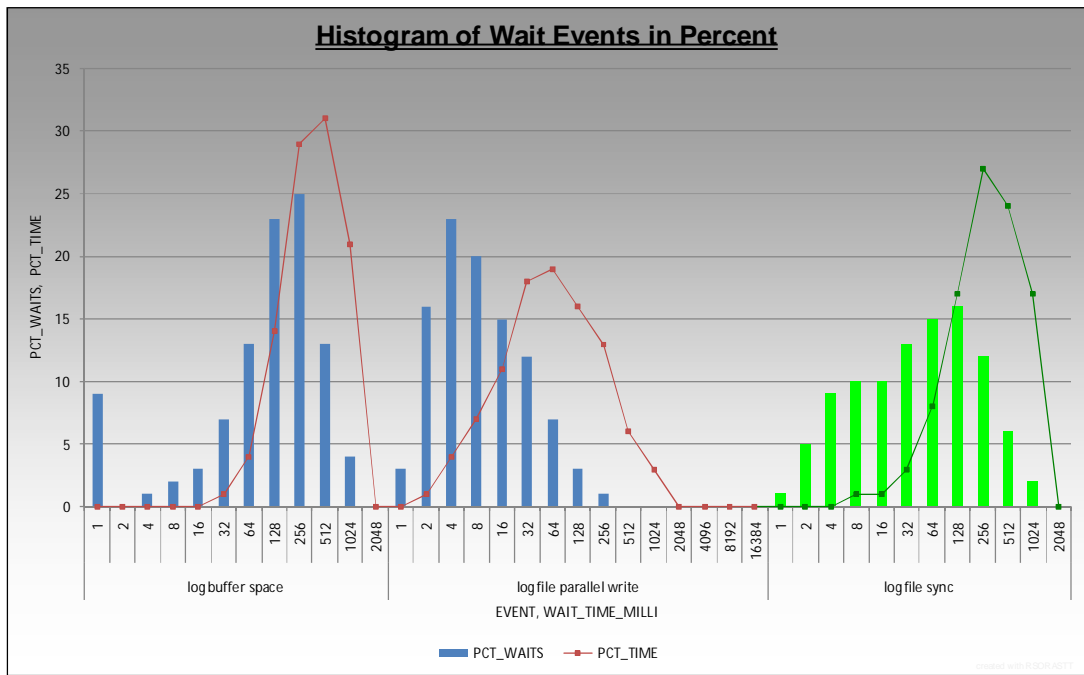


Fig. 7.2.1.2.1-2 Histograma de los eventos relacionados los logs de base de datos.

Podemos observar en la figura 7.2.1.2.1-2 que el evento de "log file sync" que se genera durante un COMMIT cuando el proceso de LGWR escribe los cambios de información del búfer de redo al área de los redo logs. Esto sucede de manera síncrona y el COMMIT solo puede terminar una vez que toda la información se ha grabado en disco.

El promedio de "log file sync" se esperaría tener en valores no mayores a 10 ms de acuerdo a [FM619188]

En la tabla 7.2.1.2.1-1 presentamos un ejemplo del histograma para el evento "log file sync", que se puede obtener de la vista in V\$EVENT\_HISTOGRAM:

EVENT	UPPER_WAIT_TIME_MS	WAIT_COUNT	TIME_WAITED_MS	PERCENT	CUM_PERCENT
log file sync	1	4320302	2160151	0.01	0.01
log file sync	2	19888080	29832120	0.09	0.1
log file sync	4	31729453	95188359	0.3	0.4
log file sync	8	36677790	2.2E+08	0.69	1.08
log file sync	16	37633213	4.52E+08	1.41	2.49
log file sync	32	45849052	1.1E+09	3.44	5.93
log file sync	64	53902637	2.59E+09	8.08	14.01
log file sync	128	57071681	5.48E+09	17.11	31.11
log file sync	256	45032338	8.65E+09	27	58.11
log file sync	512	20516436	7.88E+09	24.6	82.71
log file sync	1024	7211656	5.54E+09	17.29	100
log file sync	2048	202	310272	0	100

Tabla 7.2.1.2.1-1 Tabla del histograma para el evento log file sync.

En la tabla con el código de colores podemos observar que 1.08% del total de sincronizaciones ocurre en menos de 16 ms, 14% entre 16- 64 ms y 82.71% en más de 64 ms, lo cual demuestra que se tiene un problema con la escritura del LGWR o que la cantidad de información que se tiene que escribir a disco en el log de redo.

Algunas de las causas que pueden presentar este problema son:

Tamaño inadecuado de los bloques en el sistema operativo lo que puede causar un efecto en el que solo una parte del bloque se escribe a disco. Este problema es conocido en AIX y se puede encontrar

si se utiliza un sistema de archivos CIO (Concurrent IO) y JFS2 (Journal File System 2). Para este problema en específico se deben configurar bloques de 512 bytes ya que Oracle precisamente escribe los logs con esa longitud. El comando `lsfs -q` para averiguar esta particularidad.

Se recomienda adicionalmente no tener los archivos de datos demasiado tiempo en modo de respaldo, ya que eso causa que toda la información transaccional se escriba al log temporalmente.

Finalmente se debe también pedir una inspección del proveedor de hardware para investigar los tiempos de respuesta en los dispositivos donde se escribe esta información. La nota de SAP 793113 puede brindar más información sobre la configuración de IO para Oracle.

Como conclusión podemos decir que en base a los eventos que reporta globalmente la instancia de base de datos podemos empezar a buscar elementos de optimización que nos brinden los mejores resultados y seguir ese método para obtener resultados óptimos.

En el capítulo 7.5.2.2 podemos profundizar más en el tema particular de optimización de sentencias de SQL.

### 7.2.1.3. Análisis de la configuración de Memoria (ST02)

El análisis de búferes permite al ingeniero ver información relacionada a los búferes o el rendimiento de la memoria principal del sistema SAP con el fin de ajustar adecuadamente los parámetros respectivos. Esto es muy importante ya que si llega a presentarse con cierta frecuencia el fenómeno de "swapping" (transferencia de información entre la memoria física y la memoria virtual en el disco) puede afectar severamente el rendimiento generalizado del sistema. Este problema se puede atacar agregando memoria física al sistema.

Revisar con regularidad el comportamiento de los buffers de SAP permite distinguir tendencias y ver el comportamiento de estos segmentos asignados de memoria.

El procedimiento a seguir es:

1. Entrar a la transacción ST02. Desde esta transacción podemos ver un el estatus actual de cada uno de los principales buffer dentro de SAP.
2. Dentro de la Pantalla existen dos valores de nuestro interés principal.
  - a. HIT RATIO. Este radio representa en porcentaje cual es la cantidad de peticiones que se han podido cubrir sin necesidad de hacer swapping. El valor esperado es aproximadamente de 98% o mayor (con excepción de los búferes de programa, de export/import y single record buffer donde valores menores son aceptables). Hay que tomar en cuenta que el sistema necesita de cierto tiempo para poder llenar estos buffers, por lo que cada vez que se reinicia un servidor de aplicación estos valores pueden permanecer bajos por algún tiempo mientras "calienta". Este tiempo puede variar de acuerdo a la utilización misma del sistema y es desde horas hasta un día.
  - b. SWAPS. El valor al cual se debe de aspirar en todos los buffers es de cero. Solo existe una excepción en el búfer de programa donde 10,000 swaps aún son considerados aceptables. Esto sucede si el sistema no cuenta con la información requerida y tiene que ir a la base de datos para obtenerla. Recordemos que la mayoría de la información necesaria para correr el sistema SAP ABAP se tiene guardada como objeto a nivel de la base de datos. Entonces los SWAPS los podemos entender como objetos que se desplazan del búfer. En este particular caso como el objeto no puede ser colocado en memoria por la falta de espacio o fragmentación consecuentemente es necesario crear más memoria para la nueva información (Swapping). En caso de que el sistema sea reiniciado este valor será cero y deberá permanecer así, de lo contrario se necesita optimizar los búferes del sistema siempre y cuando exista memoria física disponible.



- c. FREE MEMORY. Para evitar los SWAPS se tiene que asegurar que cada uno de los búferes cuente con suficiente memoria y suficientes entradas libres indicadas en la pantalla como *Free Directory Entries*.

Buffer	HitRatio %	Alloc. KB	Freesp. KB	% Free Sp.	Dir. Size	FreeDirEnt	% Free Dir	Swaps	DB Accs
System: morgai_JNP_40 Tune summary Date + Time of Snapshot: 08.06.2011 10:47:04 Startup: 08.05.2011 16:29:05									
nametab (NTAB)									
Table definition	99,84	20.375	7.971	46,64	59.938	27.952	46,63	0	739.638
Field definition	99,99	64.695	10.145	16,90	59.938	44.085	73,55	5.762	21.784
Short NTAB	100,00	11.877	9.384	93,80	14.984	12.585	83,99	0	2.407
Initial records	61,05	9.873	2.018	25,23	14.984	7.187	47,96	5.525	13.340
program	99,96	1.200.000	67.256	6,10	300.000	264.933	88,31	1.557	112.461
CUA	99,98	20.000	8.842	55,50	10.000	9.568	95,68	0	462
Screen	99,99	19.532	9.483	51,22	10.000	9.052	90,52	0	993
Calendar	100,00	488	143	29,92	200	50	25,00	0	150
OTR	99,37	16.384	8.680	66,11	8.000	659	8,24	223.504	0
Tables									
Generic Key	99,98	165.040	10.561	6,95	42.000	4.404	10,49	557	2.869.810
Single record	99,79	60.000	8.553	14,31	500	358	71,60	0	1.159.532
Export/import	81,68	205.000	83.167	46,46	64.000	35.124	54,88	583.000	0
Exp./ Imp. SHM	95,98	81.920	70.501	95,54	20.000	18.996	94,98	0	0

Fig. 7.2.1.3-1 Transacción ST02 Vista General de búferes

- 3. El servidor de aplicación ABAP de SAP automáticamente señala con rojo cada uno de los búferes para los cuales existen swaps, como se muestra en la figura 7.2.1.3-1 donde existe una gran cantidad de búferes con problemas de swaps. En este caso al ser solo un vistazo del momento actual vale la pena revisar con mayor detalle el comportamiento de ese búfer a través del tiempo y corroborar el patrón de comportamiento. En las siguientes figuras se muestra el detalle de un búfer y su comportamiento a través del tiempo. Observamos que fue a partir de cierto día cuando se elevó el número de desplazamientos y es por eso importante investigar un poco más el detalle de ese día como se muestra a detalle en la figura 7.2.1.3-2.

Efficiency	HITRATIO %	HITS	REQUESTS	DB access quality %	DB access	DB access saved	Reorgs	Size	Allocated KB	Available KB	Used KB	Free KB	Directory entries	Available	Used	Free	Swaps	Objects swapped	Frames swapped	Resets	Total
	71	18.478	25.932	100	0	0	0		4.096	3.375	2.975	4		2.000	1.112	888		676	9.683	0	0

App. Server	Buffer Name	Date	Hit rate	DB quality	Al. adr. sp.	St. avail.	Free Stor.	DB acc. Max.	obj.	Free Obj	Swapped	o.	Time
db61par17_A7A_0	Export/Import buffer (EIBUF)	Fri 14.10.2011	71,0638	100,0000	4,096	3,375	1	0	2,000	868	9,997	04:37:45	
		Thu 13.10.2011	71,2384	100,0000	4,096	3,375	5	0	2,000	990	9,883	23:00:58	
		Wed 12.10.2011	69,1720	100,0000	4,096	3,375	6	0	2,000	911	2,858	23:00:52	
		Tue 11.10.2011	68,9505	100,0000	4,096	3,375	15	0	2,000	916	2,782	23:00:50	
		Mon 10.10.2011	69,4552	100,0000	4,096	3,375	15	0	2,000	953	1,025	23:00:48	
		Sun 09.10.2011	69,9387	100,0000	4,096	3,375	156	0	2,000	1,027	0	23:00:47	
		Sat 08.10.2011	69,6592	100,0000	4,096	3,375	157	0	2,000	1,029	0	23:00:46	
		Fri 07.10.2011	69,4114	100,0000	4,096	3,375	160	0	2,000	1,033	0	23:00:45	
		Sun 25.09.2011	82,5939	100,0000	4,096	3,375	3,245	0	2,000	1,941	0	23:00:32	
		Sat 24.09.2011	78,2361	100,0000	4,096	3,375	3,245	0	2,000	1,941	0	23:00:31	
		Average	71,9717	100,0000	4,096	3,375	999	0	2,000	1,151	2,635		

Fig. 7.2.1.3-2 Reporte diario del número de invalidaciones en búferes

- 4. En caso de que los desplazamientos o SWAPS sean demasiados se debe proceder a:
  - a. Revisar si el búfer es demasiado pequeño en la columna de *Free Space* o si el máximo número de entradas en búfer es demasiado pequeño en la columna de *Free Directory Entries*.
  - b. Dependiendo del resultado de esta verificación se puede ir incrementado el valor del búfer y del número máximo de entradas permitidas en porcentajes de 20% a 50%. Para saber que parámetros modificar se puede dar clic en el botón de *Current Parameters* y de ahí se puede ver los parámetros involucrados con este búfer como se muestra en la siguiente figura 7.2.1.3-3 para el caso del búfer de Export/import.

Buffer Name	Comment
rsdb/obj/buffersize	4096 kB Size of export/import buffer
rsdb/obj/max_objects	2000 Max. number of objects in the buffer
rsdb/obj/large_object_size	8192 Bytes Estimation for the size of the largest object
rsdb/obj/mutex_n	0 Number of mutexes in Export/Import buffer

**Fig. 7.2.1.3-3 Parámetros relacionados con el búfer afectado**

Siempre antes de incrementar hay que revisar que se cuenta con suficiente memoria física para evitar problemas de swapping a nivel sistema operativo.

Dentro de la misma transacción ST02 también podemos observar los valores relacionados a la Memoria de SAP, en esta memoria se incluye las áreas de memoria de Roll, de Page, de extendida y de Heap, ver figura 7.2.1.3-4.

SAP Memory	Curr.Use %	CurUse[KB]	MaxUse[KB]	In Mem[KB]	OnDisk[KB]	SAPCurCach	HitRatio %
Roll area	0,40	1.042	19.840	131.072	131.072	IDs	96,01
Page area		10	952	65.536	196.608	Statement	97,00
Extended memory	3,03	126.976	729.088	4.186.112	0		0,00
Heap memory		0	0	81.443	0		0,00

**Fig. 7.2.1.3-4 Áreas de memoria por proceso de trabajo en la instancia**

Para monitorear esta áreas se recomienda lo siguiente:

1. Para Memoria Extendida el valor máximo de uso (*MaxUse*) debe ser al menos de 20% menor que el valor en la columna de *In Memory*. Esto asegura que se tiene un margen de seguridad para la memoria extendida.
2. Para el área de Roll, el valor máximo no debe pasar el valor que se tiene en memoria, es decir que el archivo de roll no se debe usar en ningún momento.

En los sistemas ERP de SAP se observa un mal rendimiento cuando la memoria extendida está llena, por lo que se le debe dar alta prioridad al monitoreo de esta área de memoria. Lo más recomendable es dar un gran valor a la memoria extendida, ya que si no se utiliza el sistema operativo llevará esa memoria a la memoria virtual (disco). Como reglas generales podemos decir que cada usuario requiere de 6 a 10 MB de memoria extendida y que del 70% al 120% de la memoria física se puede asignar a la memoria extendida.

Existe un parámetro de memoria que delega la tarea de administración de memoria con base en el hardware disponible. Con este parámetro se vuelve casi innecesario modificar los valores, aunque existen excepciones. Este método se conoce como *Zero Administration Memory Management* y más detalle se puede encontrar en la nota de SAP 88416. El parámetro a modificar dentro del perfil de SAP es *PHYS\_MEMSIZE*, el cual define cuanta memoria principal tiene la computadora y de ahí la configuración se hace automáticamente a la hora que inicia el servidor.

#### 7.2.1.4. Análisis de los procesos de SAP (SM50, SM51, SM66)

La herramienta para el análisis de procesos de trabajo de SAP se puede ejecutar llamando la transacción SM50 para un servidor de manera local o la transacción SM66 para ver una lista global de los procesos. A continuación mostramos la figura 7.2.1.4-1 con una imagen de la transacción SM66 para mostrar los campos o columnas relevantes que se describirán a lo largo del capítulo.

Server Name	No.	Type	PID	Status	Reason	Sem	Start	Error	CPU	Time	User	Report	Action	Table
sigafappl1_MHP_00	2	DIA	3712	Running			Yes	2		2637	CPOONSITE	ZRE_SIGA		

Fig. 7.2.1.4-1 - SM66 Vista Global

Si en principio queremos solo monitorear un solo servidor de aplicación, lo mejor es utilizar la SM51 para seleccionar el servidor al cual se quiere monitorear.

La SM50/SM66 solo obtiene la vista inmediata de los procesos que están siendo ejecutados en el instante y para poder monitorear un proceso se necesita refrescar la vista continuamente. Solo en casos de emergencia donde no sea posible entrar al sistema SAP por medio del GUI, se puede utilizar la herramienta de *dpmon* a nivel sistema operativo con el siguiente comando `dpmon pf=<instance profile>`

La transacción puede ser utilizada para poder analizar el rendimiento de un programa de ABAP y es más conveniente su uso si está relacionado con un programa que se ejecuta por un tiempo considerable. Los programas que tiene un tiempo de respuesta muy corto se pueden analizar con otro tipo de herramientas como la ST12.

Para identificar claramente un proceso a nivel sistema operativo se tiene un campo PID (Process ID) que en conjunto con el nombre del servidor, permite la identificación del proceso a nivel sistema operativo. Este puede ayudar a identificar el proceso en otros monitores como la transacción ST06 en los *TOP CPU Processes* y ST04 en *Applications/Sessions/Lock Waits and Deadlock*.

Otra columna importante dentro de la vista es el de semáforos. El principio de semáforos es a nivel sistema operativo y se utilizan para controlar el uso de recursos del servidor de aplicación. Si un proceso pone un semáforo a un recurso, algún otro proceso que desee tener acceso a ese recurso necesita esperar a la terminación de la acción. En el caso de que varios procesos estén esperando a un semáforo, el sistema puede llegar a congelarse y se dice que los semáforos se serializan. En Unix con el comando `'ipcs -p'` nos da una lista de los semáforos activos. Ejemplos comunes son la utilización de buffers compartidos en el sistema.

Para encontrar una lista de los semáforos relacionados con SAP se puede revisar la nota de SAP 33873 o presionando F1 sobre el campo de semáforos en la transacción SM50/SM66.

### Analizar un proceso de trabajo

Por ejemplo si un programa está siendo ejecutado y el tiempo esperado de finalización ya superó las expectativas, entonces se puede iniciar la herramienta y ver exactamente cuáles son los pasos que están siendo ejecutados por el proceso de trabajo en ese preciso instante. Por lo tanto con esta herramienta no se pueden analizar situaciones del pasado ni del futuro que van a ser ejecutadas por el programa.

Un proceso de trabajo de ABAP puede asumir 4 estatus:

1. WAITING. El proceso está libre e interactivo para cualquier petición disponible
2. RUNNING. EL proceso está ejecutando código de ABAP.
3. ON HOLD. El proceso está en estado de espera. Por ejemplo cuando se llama un RFC síncrono, el proceso tiene que esperar la respuesta a la comunicación.
4. STOPPED. El proceso fue abortado, pero no reiniciado.

Para el análisis de problemas de ABAP los estatus importantes son el de RUNNING y de ON HOLD, porque son los únicos en los cuales se está ejecutando código de ABAP.

Para los que tiene el estatus ON HOLD se puede determinar la razón de la espera si se observa la columna REASON. Si utilizamos la ayuda de SAP con F1 obtendremos una descripción de cada una de estas razones.

Para los procesos con estatus RUNNING en principio se pueden tener tres acciones:

➤ Acciones de Base de Datos:

Incluyen SEQUENTIAL READ, DIRECT READ, INSERT, UPDATE y COMMIT. Esto quiere decir que la interfaz de base de datos ha realizado una solicitud al proceso o hilo de la base de datos y que el proceso de BD está ocupado realizando esta operación. Existe un campo adicional con el nombre del objeto de base de datos con el que está trabajando.

➤ Acciones del sistema SAP

Incluyen LOAD REPORT, LOAD CUA, ROLL-IN y ROLL-OUT. Estas acciones están relacionadas con tareas del sistema SAP, como por ejemplo cargar un programa. En un sistema saludable nunca se deberían tener esperas en estos tiempos. Si se llegan a detectar se puede analizar con el administrador del sistema, tal vez existan acciones de administración que no han sido ejecutadas como el SGEN.

➤ Sin acciones (el campo está en blanco)

Esto generalmente involucra la ejecución real de código de ABAP como READS o LOOPS. Si se utiliza el botón de CPU, se puede ver el consumo de CPU que se realiza por ejecución de ese programa de ABAP.

Algunos problemas comunes que se pueden detectar desde este monitor son:

➤ Procesos de UPDATE completamente ocupados

Si todos los procesos de UPDATE se encuentran ocupados, se tiene un problema en el sistema y se debe revisar la transacción SM13 para ver si por alguna razón se desactivaron. Se debe mirar la transacción SM21 y ver el log del sistema para buscar errores relacionados y una vez resuelto el problema activar de nuevo los procesos en la SM13.

➤ Problemas con el manejo de memoria

Estos problemas se pueden detectar si en el sistema se observan los procesos con el semáforo 6 como se muestra en el ejemplo de siguiente de una situación real:

No	Ty.	Pid	Status	Cause	Start	Err	Sem	CPU	Time	Program	Cl	Use	Action
0	DIA	3770	Run		yes	0	6		267	SAPMSEM1	100	CARVALM	
1	DIA	3694	Run		yes	0	0		2302	CL_SQL_R	100	RUDDERS	Sequential Read
2	DIA	3670	Run		yes	0	0		7131	CL_SQL_R	100	CARROLD	Sequential Read
3	DIA	3657	Run		yes	0	6		12899	SAPMSSY1	100	FALLEGV	
4	DIA	3785	Run		yes	0	6		436	SAPMSSY1	100	FALLEGV	
5	DIA	3727	Run		yes	0	6		355	SAPMSSY1	100	ALEREMOTE	
6	DIA	3685	Run		yes	0	6		355	SAPMSSY1	100	ALEREMOTE	

Se necesita analizar a detalle el archivo log del proceso de trabajo (dev\_w# en el directorio work) o utilizando el menú -> Process-> Trace ->Display File para poder ver a mayor detalle la descripción del problema.

Si muchos procesos de trabajo están en modo PRIV, quiere decir que el proceso de trabajo está consumiendo mucha memoria y entonces el proceso no se puede liberar hasta que termine esta tarea, esta situación limita la asignación de procesos de trabajo a otros usuarios del sistema y por tanto el rendimiento.

➤ Carga desbalanceada de usuarios

Desde el monitor global se puede ver si la carga de procesos que están activos es uniforme en todo el ambiente (servidores de aplicación disponibles). Si se encuentra esta distribución desbalanceada de trabaja y con la información sobre la capacidad de cada servidor de aplicación

➤ Problemas de RFC (Comunicación entre sistemas)

Si el proceso está en estado STOPPED y la razón que muestra es CPIC, quiere decir que está esperando comunicación por parte de un RFC. En estos casos lo ideal es buscar en los destinos RFC en la transacción SM59 los RFC que no está disponible o que presentan problemas de lentitud. Otra recomendación es revisar el sistema al cual apunta el RFC con el fin de verificar si tiene recursos disponibles para procesar las llamadas de función en forma eficiente. En caso de un sistema SAP se puede verificar con la misma transacción SM50/SM66 que no todos los procesos de trabajo están ocupados.

A continuación presentamos un caso en el cual por medio del monitoreo se puede extrapolar el tiempo de ejecución de un proceso de fondo para calcular el tiempo de finalización y también investigar antes de que termina las causa de porque está tomando demasiado tiempo.

En este caso lo que hacemos es entrar en la transacción SM66 para ver una vista global del proceso de fondo ver figura

Server Name	No.	Type	PID	Status	Reason	Sem	Start	Error	CPU	Time	User Names	Report	Action	Table
proas02_PRO_11	*7	BTC	1360102	Running			Yes			113343	BATCH	ZTRIRB045		
proas01_PRO_11	*7	BTC	962778	Running			Yes			71364	RMENDOZAM01	SAPLARCJ	Update	ADMI_RUN
proas02_PRO_11	*9	BTC	962742	Running			Yes			39600	CONTROLM	SAPLTHFB	Direct Read	REPOTEXT
proas05_PRO_11	*4	BTC	729300	Running			Yes			35823	CONTROLM	SAPFF50P	Sequential Read	FA9320

Fig. 7.2.1.4-2 - SM66 Vista Global

En esta transacción identificamos el proceso que queremos analizar y obtenemos los detalles del proceso para obtener más información sobre la sentencia SQL o línea de código que está ejecutando el proceso de trabajo. En nuestro caso se trata de actualizar una tabla llamada ZRT\_LOG, ver figura 7.2.1.4-3.

```

Last SQL statement
-----
DELETE
FROM
  "ZRT_LOG"
WHERE
  "MANDT" = :A0 AND "JOB_NO" = :A1 AND "ITEM" = :A2#
    
```

Database	Number	Time (usec)	Recs.
Direct Read	63	67,486	9
Sequential Read	166,442	884035090	831,387
Insert	1,142,240	*95042022-	571,120
Update	0	0	0
Delete	571,120	804680126	571,120
Sources	18,559 (Bytes)		
RSQL	121184456 (Bytes)		
Commit		0	
DB Procedure Calls	0	0	

Fig. 7.2.1.4-3 - Detalles del proceso de trabajo

Si tomamos pequeñas ventanas de tiempo a nivel de este detalle como se muestra a continuación en la figura 7.2.1.4-4 podemos extrapolar el resultado.

Database	Number	Time (usec)	Recs.
Direct Read	63	67,486	9
Sequential Read	166,442	884035090	831,387
Insert	1,145,142	*87323625-	572,571
Update	0	0	0
Delete	572,571	806211239	572,571
Sources	18,559 (Bytes)		
RSQI	121184456 (Bytes)		
Commit		0	
DB Procedure Calls	0	0	

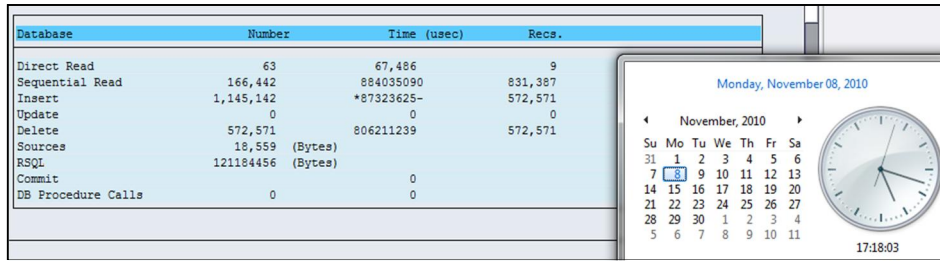


Fig. 7.2.1.4-4 Actualizaciones del proceso en ventana de tiempo cortas

Tomando como referencia los datos de las pantallas anteriores podemos afirmar que como va pasando el tiempo el programa logra tener un menor procesamiento por registro. Indicando un claro problema de linealidad como mostramos en la tabla 7.2.1.4-1.

Registros	Tiempo	Diferencia Registros	Diferencia Tiempo	Segundos	Registro /segundo	Horas estimadas para completar
572571	17:18:03					
573089	17:18:27	518	00:00:24	24	21.58	7.4
573328	17:19:07	757	00:01:04	64	11.83	13.5
573866	17:20:36	1295	00:02:33	153	8.46	18.8
577928	17:31:42	5357	00:13:39	819	6.54	24.5

Tabla 7.2.1.4-1 Extrapolación de los números de registros y tiempo de ejecución

### 7.3. Análisis de Carga en el sistema

Un análisis de carga es un estudio profundo del sistema en términos de capacidad de procesamiento, cargas del sistema y tiempos de respuesta para los sistemas SAP y sus componentes. Este análisis debe dar a conocer las áreas de oportunidad de mejora en el sistema y con esas áreas identificadas se debe hacer un análisis técnico más detallado.

Con un análisis de carga podemos enfocar los esfuerzos a las mayores problemáticas y así obtener el mejor beneficio. Un ejemplo claro es que durante la revisión de un sistema se encuentran problemas con la base de datos y con la configuración de memoria en SAP; ¿Cuál es el problema que debemos atacar primero?; ¿cuál brindara el beneficio esperado?; bueno pues un buen análisis de carga debe ayudar a contestar ese par de preguntas.

Un análisis de distribución de carga de un sistema es importante. Los motivantes pueden ser varios e involucrar a varios grupos dentro de la empresa. Algunos hechos que son indispensables mencionar para poder entender la importancia son:

- Hardware. CPU, Memoria y disco en su conjunto son costosos. Si se quiere controlar el costo total de propiedad (TCO por sus siglas en inglés) estos recursos necesitan utilizarse de la manera más eficiente.
- Siempre existen transacciones y programas que no son óptimos. Detectar y mejorar esas transacciones es necesario para mantener los costos bajo control, sin embargo hay que considerar que el análisis de rendimiento y optimización es costoso.
- En un sistema SAP ERP actual existen más de 2500 transacciones, reportes e interfaces que se ejecutan cada mes. Es imposible optimizar todas ellas y para minimizar el TCO (incluyendo los gastos de mejoras al sistema) uno necesita saber cuáles transacciones, reportes e interfaces son las más críticas y de estas cuales son las que están presentado un problema para el sistema.

El análisis de carga de un sistema debe ser la herramienta a elegir para proactivamente detectar esas transacciones, reporte e interfaces que tienen problemas de rendimiento y optimizar esos programas.

El análisis de carga además nos puede apoyar a saber cuál de los recursos es el que está resintiendo esa sobrecarga y darnos algunas pistas de donde buscar cuellos de botella.

La forma de realizar dicho análisis de carga en sistemas SAP se basa en las estadísticas colectadas por el kernel de SAP y agregadas en la transacción de análisis de carga la ST03, la cual detallaremos en el segundo tema del capítulo.

El análisis de carga se dispara generalmente por comentarios o peticiones de los usuarios del sistema. Estas observaciones subjetivas se deben verificar y encontrar la causa raíz del problema que la está causando.

A partir de este análisis inicial podemos ver si el problema reportado y analizado es de general o solo afectado a un conjunto de aplicaciones.

Los problemas generalizados dan como resultado tiempo de respuesta malos y poca capacidad de cómputo en general en todas las transacciones. Este tipo de problemas puede causar pérdidas financieras y causar problemas a los procesos de negocio del cliente.

Los problemas específicos solo afectan a una transacción o programa, sin embargo si este es parte de un proceso crítico como un cálculo de nómina también tienen una afectación fuerte en términos financieros.

### **7.3.1. Distribución de la carga de trabajo en los sistemas SAP ABAP**

Es importante mantener un equilibrio de las capacidades de cómputo que se utilizan y es por eso que se tiene que distribuir la carga de forma óptima utilizando la máxima capacidad de hardware con el mejor desempeño aceptado por el negocio. El servidor de aplicación de ABAP al ser escalable en términos de la cantidad de servidores, permite agregar hardware para distribuir la carga.

El balanceo de carga de usuarios/procesos al sistema mejora la distribución de capacidad.

Para lograr ese objetivo se tiene que distinguir la distribución de carga que generan los diferentes grupos de trabajo dentro del sistema y su impacto de rendimiento a los procesos de negocio. La distribución para ser eficiente se tiene que hacer en base a los lineamientos anteriores alineada con la capacidad de cómputo que se tiene en los diferentes servidores de aplicación.

Los grupos de LOGON se utilizan se crearon con la idea que los usuario de dialogo no tuvieran un servidor de aplicación dedicado a donde trabajan y en su lugar a partir de un algoritmo se asigne al usuario al servidor con menor carga que puede ser el que tiene mejor tiempo de respuesta. Este método cuenta con dos ventajas principales:

1. Los usuarios no se asignan a servidores fijos que un momento dado se pueden sobrecargar.
2. Como los usuarios tienen configurado un grupo de LOGON en caso de que se tengan que cambiar o agregar nuevos servidores, el cliente de SAPGUI no se tiene que reconfigurar para los usuarios finales.

En un ambiente con pocos servidores de aplicación un grupo común se usa para todas las instancias, este grupo se puede llamar PUBLIC como se acostumbra comúnmente.

En un ambiente con muchos servidores de aplicación, lo mejor es designar recursos dedicados a grupos de trabajo que así lo requieran por la criticidad o desempeño del sistema requerido, mientras que el resto se puede hacer con un conjunto más variable.

Para poder hacer esos grupos de acceso al sistema, se usa la transacción SMLG. En esta transacción el sistema muestra los nombres de grupos definidos, las instancias que lo conforman y su estatus actual. Desde aquí se pueden definir, borrar o modificar estos grupos.

También se pueden monitorear una vez que ha sido asignado para ver la cantidad de recursos, el tiempo de respuesta promedio de cada grupo y algunos valores históricos con fines de comparación.

Los valores de usuarios y tiempo de respuesta son umbrales más que límites absolutos. Estos valores sirven para el cálculo del grupo de entrada y son independientes para cada uno de las instancias dentro de grupo.

Dentro de la transacción SMLG podemos ver que existe un campo de estatus donde se muestra un indicador en forma de semáforo que sirve para indicarnos la situación actual respecto al umbral. Si por ejemplo el color es amarillo o rojo, eso indica que hemos alcanzado el umbral y que lo mejor es hacer un rebalanceo de los grupos de entrada.

En la figura 7.3.1-1 observamos un sistema con 16 servidores de aplicación entre los cuales se tiene diferentes grupos de acceso. En ellos vemos el tiempo de respuesta promedio y la cantidad de usuarios que sirven como entrada del algoritmo de balanceo.

Instance	St	Resp.time(ms)	Thrshd	User	Thrshd	Sample	Quality	Dialog steps
00_00_00	●	278	100	372	20	19:33:27	8,978-	33
01_00_01	●	1,217	50	86	10	19:29:52	6,134-	282
10_00_00	●	0	100	70	20	19:30:34	1,150-	0
11_00_00	●	0	100	73	20	19:30:21	1,225-	0
12_00_00	●	574	100	19	20	19:30:24	449-	0
13_00_00	●	1,774	100	187	20	19:34:39	5,849-	549
14_00_00	●	1,860	100	184	20	19:34:28	5,860-	583
1_00_00	●	1,223	100	217	20	19:29:56	6,048-	607
2_00_00	●	2,757	100	194	20	19:33:03	7,007-	512
3_00_00	●	492	100	98	20	19:30:02	2,342-	105
4_00_00	●	335	100	106	20	19:32:33	2,385-	133
5_00_00	●	1,833	100	185	20	19:33:14	5,858-	511
6_00_00	●	1,495	100	197	20	19:32:12	5,820-	544
7_00_00	●	1,357	100	206	20	19:30:31	5,907-	586
8_00_00	●	0		0		00:00:00	0	0
9_00_00	●	0	100	66	20	19:30:16	1,050-	0
* Summary				2,260				4,445

Fig. 7.3.1-1 Balanceo de carga

Otra forma de obtener información sobre los tiempos de repuesta y el estado de cada uno de los servidores de aplicación de forma global es mediante la transacción ST03, la cual detallaremos en la segunda parte de este capítulo, en el menú de análisis detallado y en comparación de todos los servidores. Ahí podemos verificar que los tiempos de repuesta y el número de pasos estén bien distribuidos de acuerdo al tamaño de cada uno de ellos.

### 7.3.2. ST03N

La transacción que utilizaremos para hacer un análisis de carga se llama Monitor de Carga y se puede tener acceso por medio de la transacción ST03. Una vez que la ejecutamos, obtenemos una pantalla similar a la figura 7.3.2-1 -> Cambiar



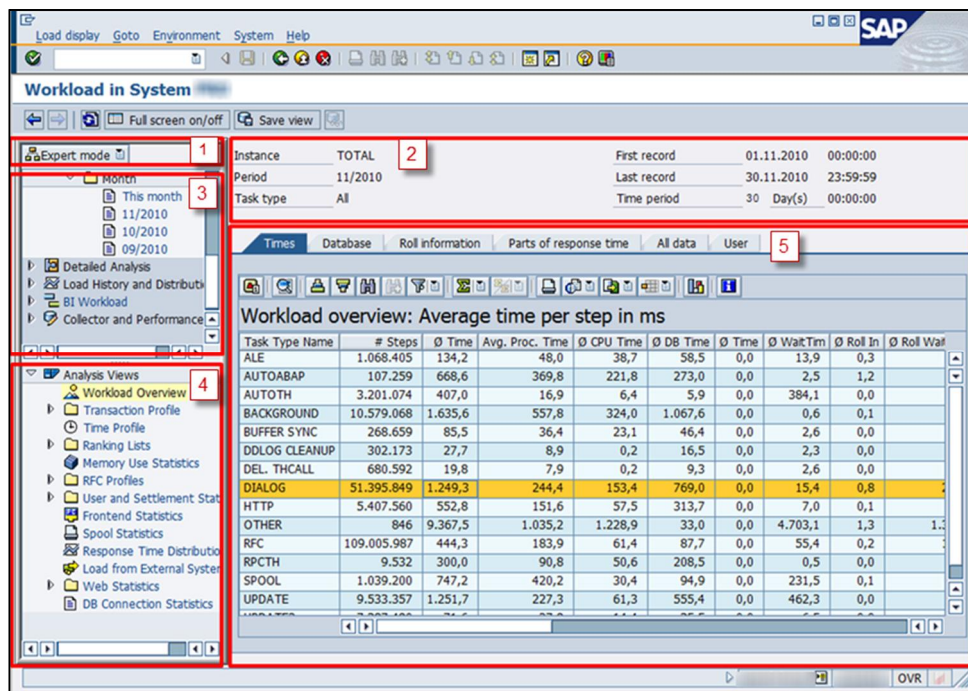


Fig. 7.3.2-1 Análisis de Carga en ST03

Recordemos que esta información es un agregado de las estadísticas que se obtienen a nivel del núcleo de SAP por cada transacción que se ejecuta en el sistema.

Esta transacción consta de varios controles que nos ayudan a navegar y obtener la información de acuerdo a como la necesitamos y de acuerdo al número de la figura 7.3.2-1, vamos a describir brevemente cada recuadro en rojo.

1. El Modo de usuario. Antes de empezar a utilizar el monitor de carga, se debe seleccionar un modo de análisis de acuerdo al perfil técnico del usuario, lo cual permite tener acceso a funciones que son apropiada al rol.
2. La información de cabecera que especifica la instancia, periodo y tipo de periodo de los campos que se están visualizando en el área de salida de información.
3. En la pantalla de funciones, se puede seleccionar con un doble clic las siguientes funciones:
  - Carga
  - Análisis detallado
  - Historia de carga y distribución
  - Distribución de carga en BW
  - Colectores y base de datos de rendimiento

Dependiendo del modo de usuario elegido, se puede la instancia y periodo que se quieren analizar.

4. La vista de análisis nos ayuda a especificar cuáles aspectos de las estadísticas del análisis de carga queremos visualizar. Algunos de ellos son:
  - Vista general de carga
  - Usuarios
  - Transacciones
  - Perfiles de RFC
  - Perfil de tiempo
5. Área de salida. Aquí es donde se despliegan los resultados de nuestra selección basada en las opciones o filtros que se colocaron en los puntos uno al cuatro.

Para utilizarlo se recomienda utilizar Experto para el modo de usuario, el último mes (o periodo de tiempo que se quiera analizar) y la vista general de la carga en el sistema. En el área de salida podemos visualizar los tiempos de respuesta agregados por tipo de tarea con valores como el tiempo de respuesta promedio, el número de pasos, el tiempo de CPU, el tiempo de BD, etc.

Ahora si comparamos dos sistemas podemos observar tiempos acumulados de repuesta muy diferentes, como puede ser 1.871 y 1.249 segundos para el tiempo de respuesta en dialogo, que es el tiempo promedio experimentado por los usuarios para todas las transacciones.

La pregunta que es importante plantearse en este punto es que tan significativo es este promedio. ¿Qué podemos concluir acerca de este sistema con este promedio?

En si este valor y los valores de la vista general del análisis de carga no representan un valor definitivo y no se puede hacer un juicio completamente acertado sobre el estado actual de rendimiento de ese sistema. El prerrequisito aquí es traer la información en el contexto adecuado.

Para poder obtener ese contexto necesitamos información comparativa, es decir necesitamos responder algunas preguntas antes de poder concluir. Las preguntas principales que hay que responder son:

- ¿Para qué se utilizar primordialmente el sistema, cuáles son los principales módulos que se utilizan?
- ¿Cómo es la arquitectura de hardware, sistema operativo, memoria y CPU?
- ¿De dónde proviene la principal carga al sistema, de usuarios finales, reportes o interfaces?
- ¿Existen picos en el sistema? ¿Cuáles son sus horarios y temporadas?

Toda esta información es útil para entender el sistema, pero en términos de rendimiento existe otro gran problema.

Cómo podemos traer esa información al contexto adecuado, si sólo podemos compararlo con otro sistema que tenga características similares. El problema parte de que aunque en SAP se tenga una base instalada de sistemas ERP de más de 10,000 sistemas, la cantidad de características propias de cada uno, hace casi imposible compararlos entre sí. Los sistemas defieren en sistema operativo, base de datos, SAN, red, personalización, configuración, tamaño, procesos, tiempos pico, y una lista interminable de diferencias.

En resumen podemos decir que no existe una escala adecuada para tener una lista de rendimiento. Solo podemos comparar un sistema con otro sistema y que no existen dos sistemas suficientemente iguales para poder compararlos, lo cual no lleva a una contradicción para nuestro análisis.

La solución que podemos ofrecer es: si no existen dos sistemas que sean lo suficientemente similares para comparar, entonces comparemos el mismo sistema con diferentes puntos en el tiempo. Entonces el mejor contexto que podemos ofrecer a esta información es con información histórica del mismo sistema.

Un análisis de carga histórico permite identificar como el rendimiento del sistema se va desarrollando a través del tiempo y cuales programas muestran una tendencia negativa de rendimiento. Lo que buscamos es finalmente mantener un sistema con rendimiento estable y mejorar poco a poco los programas que no son óptimos.

Se recomienda guardar el concentrado de las 100-300 transacciones más costosas en términos de pasos de dialogo, tiempo total de base de datos, tiempo total de CPU y tiempo total de procesamiento. El contar con esta información agregada en una hoja de Excel nos puede ayudar a realizar un análisis de comportamiento a través del tiempo de las transacciones y de la carga general del sistema.

En la figura 7.3.2-1 podemos observar como la transacción VA01 para la creación de órdenes de venta a mantenido un comportamiento un poco irregular en los primeros meses del año seguido por un comportamiento más regularizado después de una seria de cambios en el sistema con un tiempo promedio de ejecución promedio de 0.8 segundos.

Este grafico muestra además el tiempo de respuesta acumulado en las piezas individuales más importantes que los componen que son el tiempo de procesamiento, de base de datos, de tiempo de espera en roll y tiempo de espera por cada paso. Además se agrega el número de pasos que ejecuta para verificar que tiene la misma cantidad de ejecuciones que en el pasado.

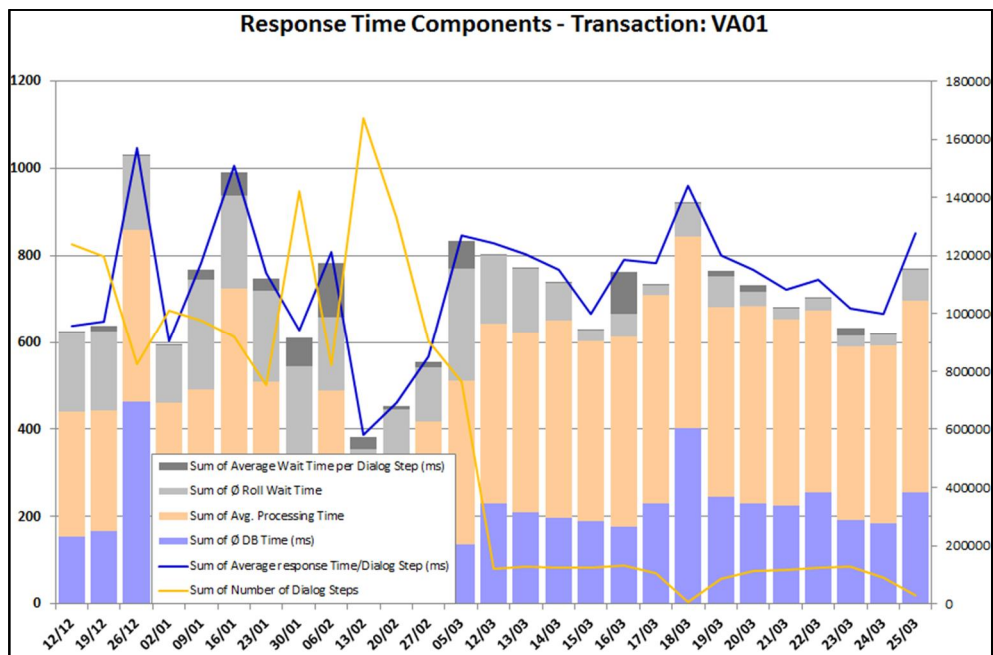


Fig.7.3.2-1 Historia del tiempo de respuesta para la transacción VA01

#### 7.4. Problemas de desempeños específicos

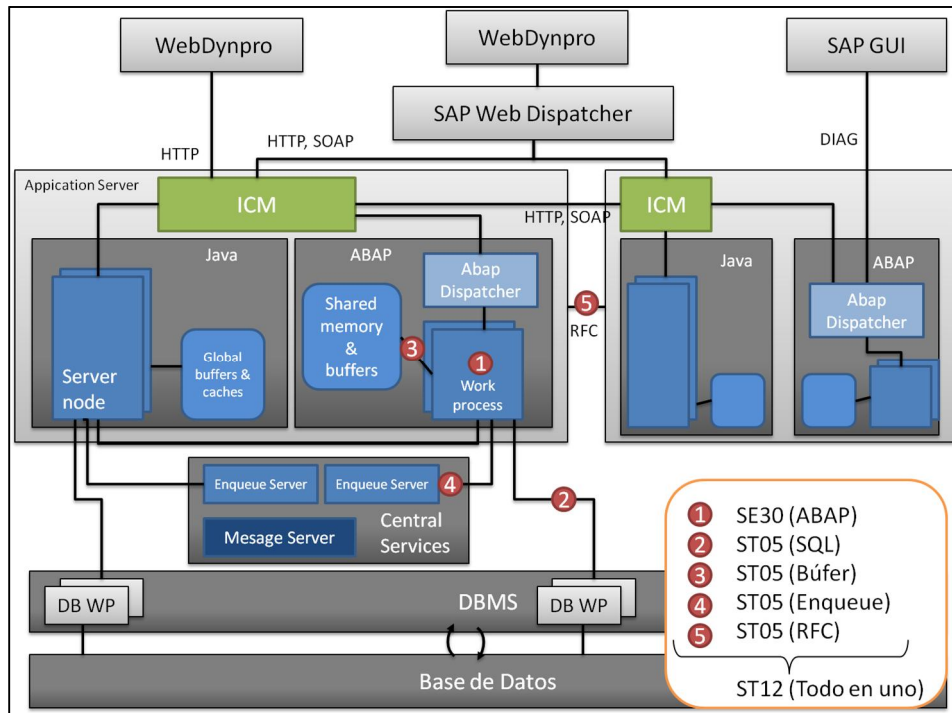
En este capítulo veremos las herramientas que se utilizan para el análisis de programas en tiempo de ejecución. Para el análisis de problemas de rendimiento en un programa ABAP tenemos varias herramientas disponibles que podemos utilizar en distintos tiempos y de acuerdo a cada situación.

Los problemas específicos entran dentro del método de mejoramiento en ABAP, este requiere sin embargo experiencia y la realización de pruebas. Se necesita comprender muy bien la arquitectura del Servidor de Aplicación ABAP de SAP, conocimiento del lenguaje programación de ABAP y manejo del mecanismo de acceso en la capa de base de datos.

En el capítulo describiremos como se utilizan estas herramientas, pero no se darán todos los detalles sino la forma común de utilizarlas y las opciones más importantes para encontrar la causa raíz de los problemas.

Cabe mencionar que algunas son herramientas de desarrollo en ABAP y otras son de administración, por lo cual un técnico especialista en el servidor de aplicación ABAP de SAP debe conocer y familiarizarse con ambas.

Las herramientas más significativas para este análisis a nivel transaccional y de programas es sin duda las transacciones STAD, SE30, SAT, ST05, SCI, ST12 y SM66, estas se pueden identificar dentro de la arquitectura de los servidores de aplicación de ABAP, como se muestra en la figura 7.4-1 Herramientas de monitoreo en la arquitectura de AS ABAP.



7.4-1 Herramientas de monitoreo en la arquitectura de AS ABAP.

A continuación describimos cada una de estas herramientas [ST2010]:

**STAD** (Análisis de transacciones de negocio). Sin duda es la herramienta más esencial para el análisis de rendimiento en un sistema. El nombre STAD proviene de su significado en inglés *Single Record Statistics*.

**SE30** (Realizar Traza de ABAP). Con el análisis de tiempo de ejecución, con esta herramienta se puede analizar el rendimiento de un programa de ABAP.

**SAT** (Realizar Traza de ABAP). Esta transacción es la sucesora de la transacción SE30, y está disponible a partir de SAP Netweaver EhP2 y sirve para hacer trazas de rendimiento, buscar cuellos de botella, análisis de flujo de programa y problemas de consumo de memoria.

**ST05** (Traza de rendimiento)

- Traza de SQL. Se puede analizar los accesos a base de datos, cuánto duran y cuántos registros se procesan.
- Traza de RFC. Se puede analizar cuáles llamadas remotas ejecuta el servidor de aplicación, en qué dirección se ejecutan, cuánta información se envía y cuánto toma en total el llamado.
- Traza de Búfer. Con una traza a nivel búfer podemos identificar los accesos a tablas que están en el búfer del servidor de aplicación, cuánto toma traer la información y como se carga la información al búfer.
- Traza de Enqueue. Se puede analizar las peticiones de encolamiento de tu aplicación o las que el sistema SAP ejecuta en objetos de bloqueo y que parámetros utiliza para los bloqueos.

**SCI** (SAP Inspector de Código). Se utiliza para hacer un análisis estático del código en términos de rendimiento, seguridad, sintaxis y adherencia a las convenciones de nombre.

**ST12** (Traza de rendimiento y ABAP). Se utiliza para ejecutar una traza de ABAP (SAT, SE30) y rendimiento (ST05) al mismo tiempo.

**SM66/SM50** (Monitor de procesos). Se utiliza para obtener información sobre los procesos que están siendo ejecutados actualmente.

En la siguiente figura podemos identificar donde podemos activar trazas a nivel de arquitectura. Existen trazas adicionales en el resto de los componentes como por ejemplo el ABAP Dispatcher, pero

estos trazes solo deben ser ejecutados a petición de SAP y se pueden analizar inicialmente con los logs que generan y que se encuentran el directorio de trabajo.

#### **7.4.1. Ejecución de trazas y su utilización (ST12, SAT, SE30, ST05)**

Para poder hacer un análisis de rendimiento SAP cuenta con herramientas para analizar sus sistemas. Se necesita saber cuáles son esas transacciones y como se utilizan para obtener la información necesaria para el análisis. Las transacciones se modifican de acuerdo a la versión de SAP, base de datos y sistema operativo.

##### **Traza de ABAP (SAT / SE30)**

SAT es una nueva transacción que se utiliza para realizar trazas en ABAP, sin embargo la vieja transacción SE30 aún se puede ejecutar. La principal diferencia es que la SAT guarda ciertos cambios que extrae de los archivos a nivel sistema operativo en la base de datos para futura referencia.

El análisis en tiempo de ejecución, también conocida como traza de ABAP sirve principalmente para hacer trazas de aplicación. Esta se encarga de guardar registros de eventos durante la ejecución de la aplicación que son llamadas por código ABAP y que son ejecutadas dentro del proceso de trabajo de ABAP.

Los eventos que computacionalmente son costosos se guardan en los registros del trace y se ignoran los que no representan costo significativo, como son las asignaciones, comparaciones y cálculos.

Los eventos que son rastreados están definidos dentro del kernel de ABAP, desafortunadamente no todos los eventos que pueden ser costos están definidos como son la asignación de una tabla interna a una tabla ordenada, crear o actualizar índices secundarios de tablas internas o las transformaciones. En estos casos los tiempos se asignan a la línea de código que las manda llamar.

La investigación detalla de cada uno de los eventos que se guardan esta soportada por otras herramientas como son la navegación directa al código y al diccionario DDIC para las tablas de la Base de Datos.

La herramienta SAT no está diseñada para cuando se tiene un problema generalizado de performance en el sistema. Sin embargo como ya lo mencionamos la herramienta permite investigar aplicaciones individuales con respecto al tiempo de ejecución e incluso el consumo de memoria.

Para los proyectos de desarrollo puede incluso ayudar a que los desarrolladores se adhieran a estándares de desempeño del producto como sucede en SAP.

Algunos de los puntos de dolor que se pueden identificar son las operaciones costosas en tablas internas, llamadas innecesarias a bloques de código ramas completas de código, unidad de modular con tiempos altos de respuesta y estructuras anidadas no lineales.

Una vez que identificamos el punto no óptimo de la aplicación es más fácil entonces generar los próximos pasos o métodos de optimización.

La SAT también puede ser utilizada como método de ingeniería inversa para deducir flujos de programa.

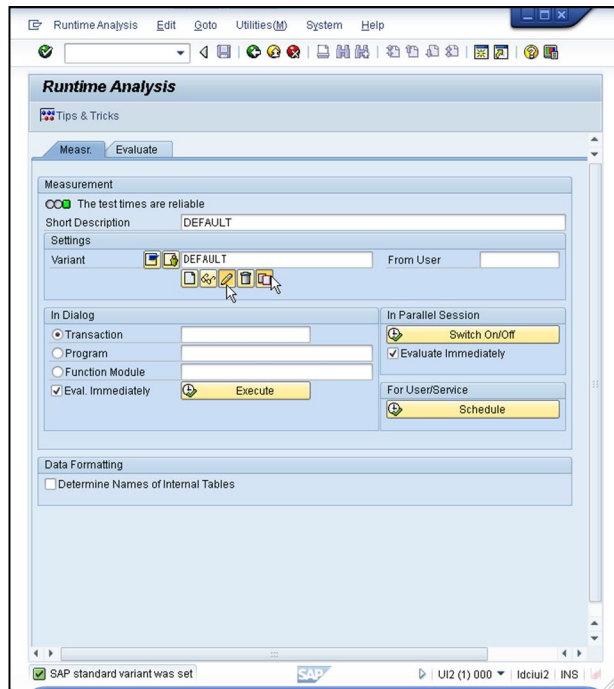
##### **Utilización de la Transacción SAT:**

Podemos separar en cuatro pasos principales para hacer uso adecuado de esta transacción:

1. Preparación. Hay que definir o seleccionar la configuración que se va utilizar para guardar la traza. Se recomienda ejecutar en varias ocasiones el programa sujeto a análisis antes para llenar adecuadamente los búferes y caches. En esta etapa definimos cómo medir, qué medir y dónde medir.

La configuración de la preparación se guarda en variantes. Recomendamos copiar la variante estándar de SAP llamada DEFAULT y partir de ahí para puede crear variante que se ajusten al programa que se quiere analizar. Lo remendable es grabar el menor número de eventos y activarlo solo por el momento de utilización ya que puede afectar en gran medida el

rendimiento del servidor de aplicación de ABAP. En la figura 7.4.1-1 podemos observar una vista de la transacción SAT.



**Fig. 7.4.1-1 Transacción SAT/SE30**

Configuración adicional que se necesita considerar es la siguiente:

Incrementar el tamaño del tamaño del archivo de traza al menor 10000 KB (10 MB), el valor estándar usualmente es insuficiente.

Agregación. Determina como se van a presentar de forma resumida los eventos que se grabe. Tenemos dos opciones “Ninguna” o “Por posición de llamada” y la diferencia principal es dentro la estructura de recursivas.

Activar la grabación de RFC y procesos de trabajo de actualización, si es que nuestra aplicación cuenta con rutinas de RFC o de actualización. El parámetro de perfil `rstr/accept_remote_trace` debe tener el valor TRUE como prerequisite.

Activar y desactivar explícitamente las mediciones. Esto nos permite tener un mejor control sobre los pasos de la interfaz de usuario que queremos grabar. El comando que se necesita ejecutar es `/ron` para inicializar o `/roff` para finalizar desde el campo de código en la transacción SAT.

Seleccionar los eventos.

- Bloques de procesamiento
- Pantallas
- Tablas internas
- Generación y carga
- Accesos a base de datos
- Transferencia de datos
- Miscelánea

Una traza inicial debe ser muy restrictiva, por ejemplo solo bloques de procesamiento y agregación por posición, y ser usado para identificar aquellas partes del código de la aplicación que se deben investigar a más detalle con una traza menos restrictiva.

1. Grabación de la Ejecución.

- a. Activar la grabación. Esta grabación está sujeta a la variante que se escogió para grabar y de ellos depende las características de la traza.
- b. Ejecutar la aplicación que se quiere grabar.
- c. Desactivar la traza. Es importante para la ejecución antes de hacer alguna otra cosa en el sistema para evitar grabar código no relacionado a nuestra aplicación.

La grabación se puede hacer por tres formas mutuamente excluyentes:

- En diálogo. Se especifica la transacción, programa o función
- En sesiones paralelas. Se selecciona un proceso de trabajo
- Para usuario o servicio. Se deja programado para que inicie una vez que las condiciones declaradas se cumplan en el servidor de aplicación.

2. Visualización. A continuación podemos visualizar le resultado en:

- a. Lista o resumen de las líneas de código (Hit List)

Todos los eventos que pasaron el filtro de eventos se agregan por llamada y en la vista default se tienen las siguientes columnas:

- Hits. Número de llamadas idénticas a ese evento.
  - Gross [microsegundos / Porcentaje]. Total en tiempo en microsegundos
  - Net [microsegundos / Porcentaje]. Tiempo total NET en microsegundos
  - Evento. Nombre del evento.
  - Programada llamado. Donde el evento fue ejecutado.
  - Programa que llama. Nombre del programa que llamo al evento.
- b. Las tablas de la base de datos involucradas.
  - c. En esta ventana podemos ver información relacionada las tablas de base de datos como el nombre, el tipo de acceso, el número de accesos el tiempo en Gross y Net, el tipo de búferes de la tabla, la categoría, el nombre de la tabla cluster o pool.
  - d. Los bloques de procesamiento. Restringe la vista a la jerarquía de llamada por unidades de modulares.
  - e. La jerarquía de llamadas de código. Despliega la jerarquía de todos los eventos que se grabaron. Para cada evento se muestran dos entradas una que corresponde al inicio del evento y otra a su final.
  - f. Lo tiempos consumidos por cada una de las sentencias ABAP ejecutadas.

La nueva transacción SAT sigue los lineamientos de las transacciones como el debugger y el DBACOCKPIT con escritorios y manejos de herramientas en la misma transacción, este arreglo así como las características arriba descritas se pueden observar en la siguiente figura 7.4.1-2.

The screenshot shows the SAP ABAP Trace Display interface. The main window is divided into several panes:

- Call Hierarchy:** A tree view showing the sequence of function calls. The root is 'Runtime analysis' (Gross: 4,827, Net: 545). It branches into 'Submit Report Z\_ITABS' (Gross: 4,282, Net: 466), 'Program Z\_ITABS' (Gross: 3,816, Net: 1,844), 'Load Report Z\_ITABS' (Gross: 59, Net: 13), and 'DB-Buffer: Get TABLP GET' (Gross: 46, Net: 46).
- Hit List:** A table listing individual hits with columns for Hit ID, Gross time, Net time, and Hierarchy Field. The first hit is 'Runtime analysis' with a Gross time of 4,827 and Net time of 545.
- Times:** A table showing performance metrics for various function calls. The first row is 'Runtime analysis' with a Gross time of 4,827, Net time of 545, and a ratio of 545/4,827.

Fig. 7.4.1-2 Información de una traza de ABAP

### 3. Análisis. Finalmente podemos analizar cada una de la líneas a detalle

- Desplegar información del diccionario de ABAP DDIC para las tablas de la base de datos.
- El código fuente de ABAP

En la pestaña de Mediciones (Measurements) se pueden definir las variantes para hacer las mediciones y guardar los registros de esas mediciones. Aquí existe dos nuevas opciones que antes no estaban disponibles: *evaluar inmediatamente* y *evaluar con nombres de las tablas internas* que en turno son muy importantes para el análisis de desempeño en ABAP como más adelante lo veremos. Algunas de las funciones nuevas son las siguientes:

- Borrar trazas que ya no se van utilizar
- Mandar Trazas por medio de RFC a otros sistemas
- Cambiar los períodos de retención en la base de datos
- Convertir a un formato universal de trace que permite mover el trace a otro sistema y guardarlo ahí
- Desplegar las traza con formato UML (sólo los no agregados)
- Compara los trazas con el fin de analizar el factor de escalabilidad de la aplicación o para detectar cambios en la lógica del programa.

Finalmente solo queda aclarar la diferencia entre lo que SAP llama Gross y Net time. El tiempo Gross para un evento se puede definir como la diferencia entre el tiempo que cuando termina e inicia. El tiempo de Event se calcula como el Gross time menos la suma de los tiempos gross de todos los eventos que se grabaron anteriormente.

La suma de todos los tiempos Net (independiente del nivel en la pila de llamadas) debe ser igual al máximo de todos los tiempos Gross, que a la vez debe ser igual al tiempo de ejecución.

### Consejos y Trucos de rendimiento.

Además de poder hacer una traza de los programas, esta transacción nos da una importante característica para poder comparar diferentes formas de implementación de código en términos de performance. Esta herramienta nos permite hacer por ejemplo una comparación entre la lectura de tablas internas con búsqueda binaria y sin esta.

En la imagen por ejemplo comparamos el uso de índices secundarios en tablas internas de ABAP, podemos ver que la diferencia entre usar una búsqueda lineal en la tabla y un índice secundario con la búsqueda binaria es aproximadamente en un factor de 12:1 microsegundos en tiempo de ejecución lo que permite al desarrollador evaluar las mejores opción de implementación de código en términos de rendimiento, un ejemplo de esta funcionalidad se muestra en la figura 7.4.1-3.



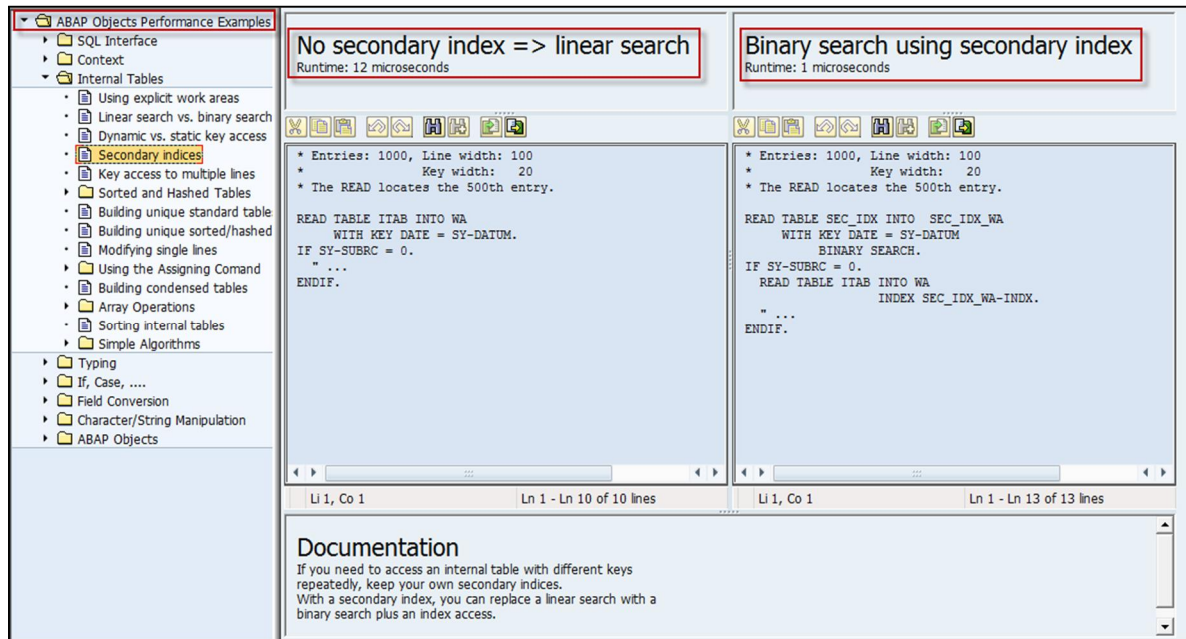


Fig. 7.4.1-3 Consejos y trucos de rendimiento en ABAP

### Traza de rendimiento (transacción ST05).

Esta es una transacción de traza de interfaces para los procesos de trabajo de SAP.

Graba los eventos de ejecución en la aplicación que salen del proceso de trabajo de SAP ya sea para obtener información externa, ejecutar servicios de fuentes externas o rastrear servicios que entran al proceso de trabajo de SAP para realizar llamadas a función remotas.

Esta transacción soporta las siguientes interfaces para realizar una traza:

- SQL – Monitorear accesos a la base de datos
- Buffer – guarda llamadas a los objetos del buffer de aplicación
- Enqueue – Rastrea sentencias de bloqueo y desbloqueo
- RFC – Rastrea llamados a funciones remotas.
- HTTP – Observa comunicación HTTP(S) o comunicación similar.

Estos rastreos están desactivados por default para evitar una sobrecarga en el sistema y se cuenta con varios niveles de agregación.

Las trazas con ST05 permiten identificar puntos críticos de rendimiento en las aplicaciones debido a sentencias SQL costosas, accesos innecesarios a la capa de persistencia, bloqueos de SAP con duración excesiva y problemas relacionados con llamados a función RFC.

La transacción ST05 no es una herramienta adecuada si el sistema entero o la base de datos esta lenta. Esta es una herramienta de análisis particular y es donde brinda el mayor de sus beneficios.

Del resultado de una traza con la ST05 debemos analizar los siguientes puntos:

- Sentencia costosas de SQL utilizando el Code Inspector de SAP (SCI)
- Sentencias que se ejecutan por mucho tiempo
- Sentencias que obtienen a muchos registros
- Sentencias con ejecuciones redundantes
- Sentencias que se ejecutan demasiado, especialmente cuando solo trae un registro por consulta
- Sentencias donde el tiempo por registro es alto (mínimo, promedio)

Para ejecutar la herramienta se deben de seguir los siguientes 4 pasos, que se pueden ejecutar desde la transacción ST05, ver figura 7.4.1-4:

1. Preparación. Ejecute la aplicación que se va analizar un par de veces antes de empezar, lo cual asegura que los buffers y chaches se llenen apropiadamente.
2. Grabado. Debe de activar el rastreo antes de ejecutar la aplicación, se recomienda utilizar los filtros proporcionados como nombre de usuario o programa para evitar interferencia. Una vez activado proceda a ejecutar la aplicación y una vez terminado, desactive el rastreo.

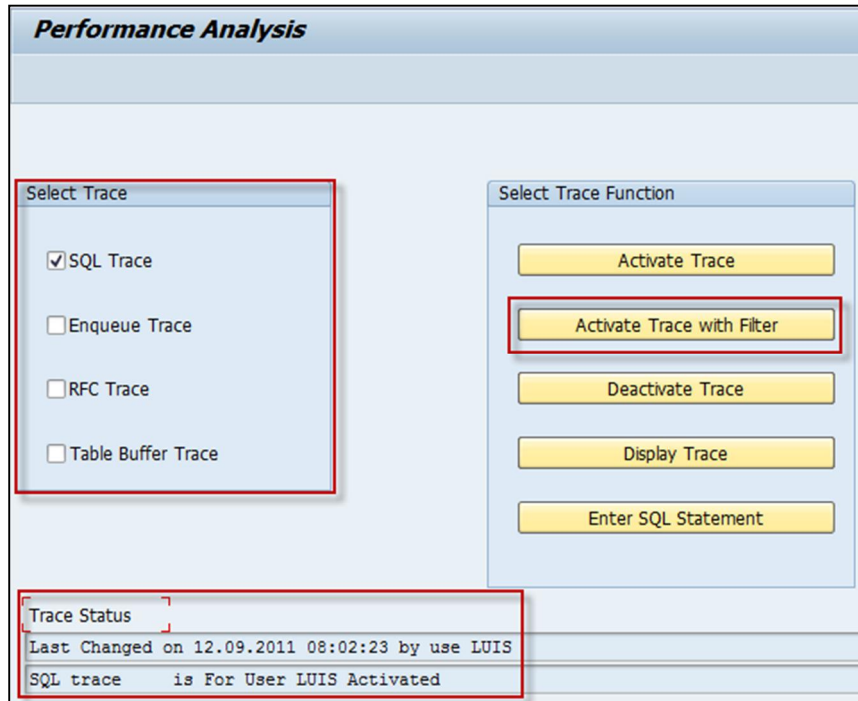


Fig. 7.4.1-4 Traza de interfaces (ST05)

3. Visualizar. El resultado del trace es muy amplio, pero se cuenta con reportes que agregan la información para mostrar las sentencias idénticas en estructura, en valores y sumariados por objetos.

La visualización soporta diferentes niveles de agregación:

- La vista mostrada en la figura 7.4.1-5 detalla muestra los registros que fueron granados sin ninguna agregación y muestra todas las operaciones realizadas por el DBSL como son el FETCH, OPEN, PREPARE, etc.

Transaction	ST12	Work process no	6	Proc. Type	DIA	Client	800	User	I820487	TransGUID	29E40CE1C93EF143914D00145ESB10DD	Date	12.11.2011
HR:MM:SS.MS	Duration	Program	Obj. name	Op.	Curs	Array	Recs.	RC	Conn	Statement			
05:09:49.513	13.311	/SSA/STA	TFDIR	OPEN	0				0 R/3	SELECT WHERE "FUNCNAME" = 'RSTR_GET_FILELIST' /* R3:KMN:0 T:TFDIR */ /*Dynamic SQL statement -- no cursor, upto 1,			
05:09:49.526	26	/SSA/STA	TFDIR	FETCH	0	1	1		0 R/3				
05:09:49.527	294	/./SLIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND "OBJKEY" = 'PROJ' ORDER BY "RELID","PID","OBJKEY","SRTFD" /* R3:/S			
05:09:49.527	49	/./SLIB	/SSF/PTAB	FETCH	592	32767	1		0 R/3				
05:09:49.528	327	/./SLIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND "OBJKEY" = 'SELC' ORDER BY "RELID","PID","OBJKEY","SRTFD" /* R3:/S			
05:09:49.528	35	/./SLIB	/SSF/PTAB	FETCH	592	32767	1		0 R/3				
05:09:49.531	356	/./ULIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND "OBJKEY" LIKE 'LOCK%' AND NOT ( "OBJKEY" LIKE 'LOCK_____%' ) /*			
05:09:49.531	28	/./ULIB	/SSF/PTAB	FETCH	592	284	0		0 R/3				
05:09:49.531	224	/./ULIB	TRDIR	OPEN	0				0 R/3	SELECT WHERE "NAME" = '/SSA/STA' /* R3:/SSF/ULIB:483 T:TRDIR */ /*Dynamic SQL statement -- no cursor, upto 1, conn.			
05:09:49.532	16	/./ULIB	TRDIR	FETCH	0	1	1		0 R/3				
05:09:49.532	187	/./SLIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND "OBJKEY" = 'UNIT F58' ORDER BY "RELID","PID","OBJKEY","SRTFD"			
05:09:49.532	37	/./SLIB	/SSF/PTAB	FETCH	592	32767	1		0 R/3				
05:09:49.533	193	/./SLIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND "OBJKEY" = 'SELC F58' ORDER BY "RELID","PID","OBJKEY","SRTFD"			
05:09:49.533	32	/./SLIB	/SSF/PTAB	FETCH	592	32767	1		0 R/3				
05:09:49.535	9	/./ULIB	/SSF/PTAB	PREPARE	592				0 R/3	SELECT WHERE "RELID" = @P000 AND "PID" = @P001 AND ( "OBJKEY" LIKE @P002 OR "OBJKEY" LIKE @P003 ) /* R3:/SSF/ULIB:15			
05:09:49.535	387	/./ULIB	/SSF/PTAB	OPEN	592				0 R/3	SELECT WHERE "RELID" = 'RG' AND "PID" = 'STA' AND ( "OBJKEY" LIKE 'LOCK_____F58			
05:09:49.536	29	/./ULIB	/SSF/PTAB	FETCH	592	284	0		0 R/3	%' OR "OBJKEY"			

Fig. 7.4.1-5 Vista detalla de la traza de SQL

- La lista de sentencias de la figura 7.4.1-6 con estructura idéntica conjunta las sentencias con una estructura común y le agrega información estadística y técnica.

Executions	Redundancy	Identical	Durtn	Records	Time/Exec	Rec/Exec	AvgTime/R	MinTime/R	Length	BfTp	TabType	Object name	Statement
57	25	44	67.197	229	1.179	4,0	293	26	124		TRANSP	SEOCOMPO	SELECT WHERE "CLE
44	33	75	50.353	0	1.144	0,0	1.144	536	536	GEN	TRANSP	TCVIEW	SELECT WHERE "MAN
18	10	56	35.265	136	1.959	7,6	259	41	514	GEN	TRANSP	SEOREDEF	SELECT WHERE "CLE
11	5	45	34.851	110	3.168	10,0	317	204	21.140		TRANSP	SEOCOMPODF	SELECT WHERE "CLE
7	3	43	30.658	63	4.380	9,0	487	205	21.140		TRANSP	SEOCOMPODF	SELECT WHERE "CLE
36	15	42	28.567	8	794	0,2	3.571	404	210	GEN	TRANSP	SEOTYPEPLS	SELECT WHERE "CLE
19	16	84	26.897	19	1.416	1,0	1.416	964	850	GEN	VIEW	VSEOCCLASS	SELECT WHERE "CLE
28	20	71	25.338	0	905	0,0	905	616	420		TRANSP	ENHOBJ	SELECT WHERE "OBJ

Fig. 7.4.1-6 Vista resumen de la traza de SQL

- La lista de sentencias de la figura 7.4.1-7 con valores idéntica conjunta las sentencias en un reporte donde se presentas sentencias con estructura y valores coincidentes.

Executions	Redundancy	Identical	Durtn	Records	Time/Exec	Rec/Exec	AvgTime/R	MinTime/R	Length	BfTp	TabType	Object name	Statement
13	0	0	24.009	0	1.847	0,0	1.847	725	0				COMMIT WORK ON
3	0	0	15.476	42	5.159	14,0	368	343	840	FUL	TRANSP	ICFATTRIB	SELECT WHERE "A
3	0	0	10.804	0	3.601	0,0	3.601	1.266	3.100		TRANSP	WDY_WB_SOURCEMAP	SELECT WHERE "R
6	0	0	10.572	0	1.762	0,0	1.762	1.027	3.100		TRANSP	WDY_WB_SOURCEMAP	SELECT WHERE "R
10	0	0	8.954	10	895	1,0	895	656	1.176	DESNG	TRANSP	USOB_AUTHVALTRC	SELECT WHERE "N
3	0	0	7.159	3	2.386	1,0	2.386	1.248	924		TRANSP	BDSPHF2	SELECT WHERE "P
4	0	0	6.991	4	1.748	1,0	1.748	803	138		TRANSP	DIRTREE	SELECT WHERE "TI
2	0	0	6.515	2	3.258	1,0	3.258	2.427	1.176	DESNG	TRANSP	USOB_AUTHVALTRC	SELECT WHERE "N
3	0	0	6.245	3	2.082	1,0	2.082	1.294	926		TRANSP	BDSPHIO2	SELECT WHERE "LI
3	0	0	6.162	3	2.054	1,0	2.054	1.795	308		TRANSP	BDS_CONN05	SELECT WHERE "C

Fig. 7.4.1-7 Vista de sentencia SQL idénticas

- La lista de resumida de tablas de la figura 7.4.1-8 consta de tres vistas. La primera combina todas las operaciones de base de datos de una sentencia; La segunda agrega los accesos por tabla y conexión de base de datos para todos los SELECT, INSERT, UPDATE y DELETE. Finalmente la tercera genera una vista jerarquizada por componentes con subtotales y descripciones de la tabla del diccionario de ABAP DDIC.

Work Proc. No.	PType	Client	HH:MM:SS.MS	Transaction	Table Name	Statement	ΣRecords	ΣAccess Time	DB Connection
0			09:50:47.645		BTCCTL	SELECT	1	2.128	R/3
			09:50:47.647		BTCCTL	UPDATE	1	1.488	R/3
			09:50:47.650			COMMIT	0	4.462	R/3
			09:50:47.656		BTCOMSDL	SELECT	0	2.316	R/3
			09:50:47.658		TBTC	SELECT	3	6.872	R/3
			09:50:47.666		BTCOPTIONS	SELECT	1	2.219	R/3
			09:50:47.668		BTCOPTIONS	SELECT	0	2.826	R/3

Figura 7.4.1-8 Operaciones de SQL agregadas por DML y tablas

4. Analizar las sentencias individualmente, se puede obtener información del diccionario, utilizar la explicación del camino de acceso de la base de datos, la línea de ABAP que lo manda llamar y la pila de ABAP.

Para el análisis de la sentencia de SQL se puede ver el capítulo 7.4.2.2 Análisis de rendimiento de sentencias de SQL.

Una traza puede estar activada por un servidor de aplicación a la vez, por lo que es importante desactivar el trace una vez que terminemos para no interferir con otros usuarios.

### **Registros estadísticos (Transacción STAD)**

Se escriben registros estadísticos de cada una de las acciones y transacciones que se llevan a cabo dentro del servidor de aplicación de ABAP. Esta característica está implementada dentro de la funcionalidad del kernel de SAP y no necesita activarse para poder usar la información recolectada en un análisis de rendimiento.

La transacción por su naturaleza se puede utilizar para un análisis post ejecución, lo cual es bueno si necesitamos obtener algunas métricas y el programa no es reproducible temporalmente.

El trasfondo técnico como lo indicamos viene del kernel de SAP que ejecuta las funciones a nivel de sistema operativo bajo la máquina virtual del proceso de trabajo de SAP. Algunas de estas funciones de hecho se pueden visualizar desde el monitor de procesos (SM50/SM66) en el mismo estatus del programa.

El mecanismo de recolección de información de la transacción STAD se muestra en la figura 7.4.1-5. La información se agrega en un registro estadístico al final de cada paso transaccional y también se crea si un programa termina. Todo este proceso se guarda en memoria por motivos de rendimiento y se escribe asincrónicamente a un archivo en el sistema operativo. Estos archivos generalmente se encuentran en el directorio `/usr/sap/<SysID>/<Instancia>/DATA` o también desde la AL11 puedes ver la definición de la variable DIR\_DATA. Los archivos deben tener la nomenclatura `statnnn` donde nnn es un número.

Cuando ejecutamos la transacción dependiendo del rango de tiempo que especifiquemos la información será obtenida este archivo a nivel sistema operativo y de lo más reciente que exista en memoria.

Para no perder esta información a los cuatro días como máximo, los registros se agregan aún más y se guardan en la tabla SWNCMONI de la base de datos. Este trabajo es ejecutado cada hora por un proceso de fondo llamado RSCOLL0. Esta información se puede consultar agregada y centralizada para toda la solución utilizando en la transacción ST03.

La STAD conserva la granularidad necesaria para identificar problemas que puede que no sean visibles con valores promedios del perfil de la transacción en la ST03.

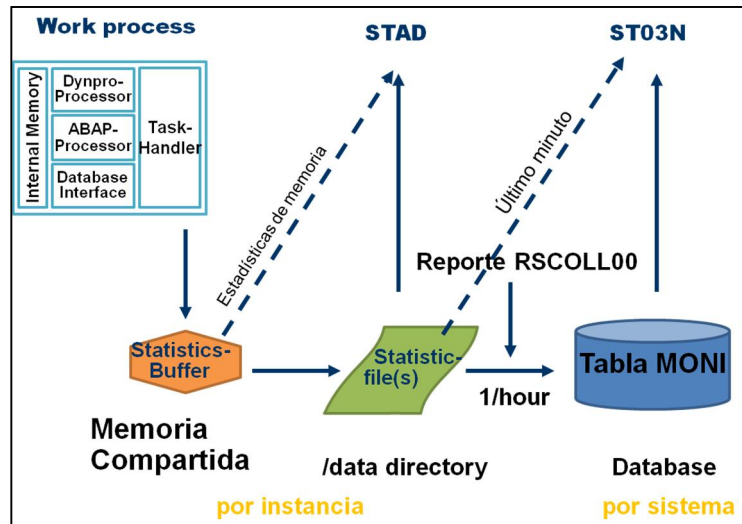


Fig. 7.4.1-5 Recolección de estadísticas [ST2010]

El número de archivos que se pueden guardar está determinado por el parámetro *stat/max\_files* que pueden almacenar aproximadamente una hora de procesos. Este archivo tiene un máximo valor de 99, lo que se traduce en 99 horas máximo que es aproximadamente 4 días.

#### Utilización de la transacción STAD.

Cuando se llama la transacción STAD se despliega una pantalla de selección que se muestra en la siguiente figura 7.4.1-6:

Fig. 7.4.1-6 Vista de la transacción STAD

A continuación describimos la utilización de cada uno de los cuadros de selección desglosando cada una de las áreas marcadas con un número:

1. En la primera sección Visualización podemos escoger la forma de desplegar la información, por default está organizada por tiempo.
2. En la segunda sección Intervalo de lectura tenemos campos para especificar el intervalo, para el cual queremos obtener información. Siempre tiene predefinido los últimos 10 minutos. La

selección de periodos muy grandes de tiempo puede consumir mucha memoria ya que es demasiada la información que se tiene que extraer.

3. En esta sección podemos filtrar por diferentes criterios para reducir el número de registros y apuntar mejor a nuestra búsqueda.
4. En la última sección tenemos opciones para obtener los registros de la memoria, incluir más servidores de aplicación y modificar el tiempo que se espera por RFC.

Con los criterios anteriores y la información necesaria se puede encontrar de manera fácil los registros que estamos buscando. Para el caso de los trabajos de fondo que duran varias horas a partir de la versión 7.00 del servidor de aplicación solo se necesita incluir dentro de la venta de tiempo los últimos 10 minutos en donde termino el trabajo de fondo y obtiene de manera automática todo los pasos de este trabajo.

Como resultado del criterio de búsqueda obtenemos una lista con cada uno de los registros estadísticos que cumplieron con los filtros de tiempo, usuario, duración, etc.

Dentro de la lista podemos utilizar el botón de “Disp. mode” para obtener diferentes vistas de esta lista, como por ejemplo organizados por tiempo, transacción y con totales incluyendo tiempos de RFC.

También podemos utilizar el botón de “Sel. fields” para poder visualizar otras columnas con información diferente. Por ejemplo existen los botones de análisis de base de datos que muestra todas las estadísticas referentes a las operaciones de base de datos y el botón de análisis de tiempo donde se muestran estadísticas basadas solo en tiempo.

La lista se organiza en dos partes principalmente:

La primera muestra información de registro como la hora en que comenzó, el servidor, la transacción, el programa, el tipo de tarea ((D = dialogo; R = RFC; U = update; B = background; H = BSP o Web Service HTTP; T = BSP seguro o Web Service HTTPS), el primero proceso de trabajo de SAP que se utilizó y el usuario.

La segunda muestra información estadística referente a este registro como es el tiempo de respuesta, tiempo que estuvo en el proceso de SAP, el tiempo de CPU, el tiempo de base de datos, el tiempo de espera, etc.

**SAP Workload: Single Statistical Records - Overview**

Download **Disp. mode** **Sel. fields** Server ID

System: A7A Number of RFCs which responded (without errors): 1 ( 1)  
 Analysed time: 12.09.2011 / 03:26:00 - 12.09.2011 / 03:46:00  
 Display mode: All statistic records, sorted by time

Started	Server	Transaction	Program	T Scr.	Wp	User	Response time (ms)	Time in WFs (ms)	Wait time (ms)	CPU time (ms)	DB req. time (ms)	VMC elapsed time (ms)	Memory used (kB)	Transfe kBytes
*	*	*				LUIS	0			0	0			
03:26:24	db61par17_A7A_00	STAD	RSSTAT26	D	0120	0	16	16	0	10	0	0	4.273	0
03:26:25	db61par17_A7A_00	STAD	RSSTAT26	D	0010	0	534	534	0	20	0	0	4.273	0
03:26:34	db61par17_A7A_00	STAD	RSSTAT26	D	0120	0	125	113	6	100	0	0	4.273	0
03:26:42	db61par17_A7A_00	SESSION_MANAGER	SAPLSMTR_NAVIGATION	D	0100	0	6.067	5.262	8	230	56	0	4.273	201
03:26:42	db61par17_A7A_00	STAD	RSSTAT26	D	0120	0	5	5	0	0	0	0	4.273	0
03:26:56	db61par17_A7A_00	DB6PERF	SAPLSDB6CCMS	D	0050	0	7.639	4.413	13	720	3.089	0	8.364	21
03:27:11	db61par17_A7A_00	DB6PERF	SAPLSDB6CCMS	D	0050	0	878	665	0	40	1	0	8.364	0
03:27:48	db61par17_A7A_00	ST03	SAPWL_ST03N	D	1000	0	761	222	0	160	33	0	4.273	76
03:27:52	db61par17_A7A_00	ST03	SAPWL_ST03N	D	1000	0	341	47	0	20	6	0	4.273	2
03:27:56	db61par17_A7A_00	ST03	SAPWL_ST03N	D	1000	0	1.387	721	0	260	414	0	8.364	336

**Fig. 7.4.1-7 Resultados no agregados de las estadísticas.**

Una vez que obtenemos esta lista inicial podemos empezar a hacer el análisis a mayor detalle. Esta tarea es posible si seleccionamos el registro y damos un doble clic para obtener mayores detalles.

En esta pantalla como se muestra en la figura 7.4.1-8 podemos ver a detalle el tiempo de respuesta y cada uno de sus componentes. Además podemos ver información referente a las llamadas a base de datos, la memoria, terminal con la que se está interactuando, la cantidad de bytes de información enviada y la información de RFC.

SAP Workload: Single Statistical Records - Details						
Record		Time	DB	Task / Memory	Bytes	
System:		A7A	Instance: db61par17_A7A_00			
Analysed time:		12.09.2011 / 03:26:00 - 12.09.2011 / 03:46:00				
Record:		03:26:56 - 03:27:03	DB6PERF	SAPLSDB6CCMS	D	LUIS
Analysis of time in work process						
CPU time	720 ms	Number	Roll ins	9		
RFC+CPIC time	0 ms		Roll outs	11		
			Enqueues	0		
Total time in workprocs	4.413 ms	Load time	Program	23 ms		
			Screen	0 ms		
			CUA interf.	0 ms		
Response time	7.639 ms	Roll time	Out	2 ms		
			In	19 ms		
			Wait	3.213 ms		
Wait for work process	13 ms	Frontend	No.roundtrips	9		
Processing time	1.281 ms		GUI time	3.716 ms		
Load time	23 ms		Net time	423 ms		
Generating time	0 ms					
Roll (in+wait) time	3.232 ms					
Database request time	3.089 ms					
Enqueue time	0 ms					
Analysis of ABAP/4 database requests (only explicitly by application)						
Connection	DEFAULT	Request time	3.089 ms			
Database requests total	325	Commit time	7 ms			
DB Proc. Calls	0	DB Proc. Time	0 ms			
Type of ABAP request	Database rows	Requests	Requests to buffer	Database calls	Request time (ms)	Avg.time / row (ms)
Total	1.688	325	621	1.621	3.089	1,8
Direct read	3	48	22		2	0,7
Sequential read	1.684	275	599	1.620	3.085	1,8
Update	1	1		1	1	1,0
Delete	0	1		0	1	0,0
Insert	0	0		0	0	0,0

Fig. 7.4.1-8 Resultados por registro estadístico

En recuadros rojos en la pantalla podemos ver el tiempo de respuesta para el registro estadístico que estamos analizando y en el siguiente recuadro vemos el detalle de este tiempo en los siguientes componentes [ST2010]:

**Tiempo de CPU:** el WP de SAP está el CPU, generalmente se utiliza CPU cuando se procesa sentencias de ABAP, para comunicación con la interface de base de datos y durante el Roll in y Roll out.

**Tiempo de respuesta:** Empieza cuando la petición del usuario entra a la cola del dispatcher y termina cuando la siguiente pantalla se muestra al usuario.

**Espera de proceso de trabajo:** El dispatcher está esperando que haya un WP libre

**Tiempo de procesamiento:** Tiempo de respuesta- (tiempo de espera + tiempo carga y generación time + Tiempo de Roll (in + wait) + Tiempo de peticiones a base de datos + tiempo de Enqueue)

**Tiempo de carga:** Tiempo necesario para cargar un objeto ejecutable del buffer al WP. En la derecha podemos ver qué tipo de objetos se cargan: Programa de ABAP, pantallas, elementos de CUA (menús, botones, etc.)

**Tiempo de generación:** Tiempo necesario para compilar el código fuente en un objeto ejecutable que se carga del búfer de SAP.

**Tiempo de Roll wait:** Tiempo en espera de un RFC síncrono o la comunicación con el GUI. También puede ser sentencias ABAP "WAIT UP TO n SECONDS" o "COMMIT WORK AND WAIT" ejecutadas en un proceso de diálogo.

**Tiempo de peticiones a base de datos:** Tiempo necesario para acceder y esperar a la interfaz de base de datos y por lo tanto la capa inferior la base de datos.

**Tiempo de Enqueue:** Tiempo que se tarda en las colas definidas en el código de SAP.

La siguiente lista incluye los problemas típicos que se pueden encontrar utilizando el registro estadístico:

- Tiempos altos de base de datos usualmente indican un problema en la selección o cambio de la información en la capa de base de datos. Se puede obtener más información si se ejecuta una traza de SQL con la transacción ST05. Usando los valores de *KBYTES TRANSFERED* o *DATABASE ROWS* dos tipos de problemas se identifican:
  - El tiempo de base de datos es alto aun cuando poca información se transfiere al servidor de aplicación. El tiempo óptimo de transferencia es un milisegundo por cada registro. Si el tiempo es considerablemente mayor, se debe analizar la sentencia SQL ya que traer la información toma demasiado tiempo.
  - El tiempo de base de datos es alto porque la cantidad de información que se transfiere es muy grande, pero el tiempo por cada registro es óptimo (~ 1 milisegundo). En este caso la sentencia de SQL puede que sea óptima, pero se necesita revisar si es necesaria esa carga de información.  
Se puede obtener más información sobre el mejoramiento de sentencia SQL en la sección 7.4.2.
- Si se encuentran tiempos de base de datos altos, pero estos solo son esporádicos se debe buscar dentro de los registros un mensaje de que las tablas se guardaron en el búfer de tablas. Esto significa que las tablas han sido transferidas de la base de datos al búfer de tablas de SAP. Para sistemas productivos generalmente estas tablas no se tiene que cargar frecuentemente de la base de datos y ese efecto ocurre, quiere decir que existe una cantidad considerable de desplazamientos en el búfer de tablas de SAP. En este caso se puede hacer un análisis con la transacción ST10 para obtener el número de invalidaciones.
- Tiempos altos de GUI y de roll-wait indican problemas de comunicación, los cuales se pueden analizar con una traza de SQL o de E2E.
- Tiempos altos de CPU indican que se están ejecutando operaciones que consumen mucho tiempo en el código de ABAP o que se está consultando frecuentemente al búfer de tablas interno de SAP. Registros con tiempos de CPU mayores a 60% se pueden analizar con una traza de ABAP con la SE30, SAT o ST12.

Algunos valores típicos que se pueden considerar para indagar más al respecto son:

**Tiempo de espera:** < 10% del tiempo de respuesta o < 50 ms  
Revisar porque todos los procesos de trabajo están ocupados, si hay memoria extendida disponible o los están en modo PRIV, si hay demasiados RFC sobrecargando el sistema o si hay bloqueos en NRIV.

**Tiempo de carga:** < 50 ms  
Repetir la ejecución del programa. Si el problema persiste generalmente es un problema de configuración de parámetros del Servidor de Aplicación ABAP. El búfer de programa es un buen principio para empezar a buscar sobre escrituras con la transacción ST02.

**Tiempo de Generación:** Debe ser cero después de la primera ejecución.  
Revisar el buffer de programa.

**Roll-in, Roll-out:** < 20 ms  
Revisar en la transacción ST02 si el área de Roll está siendo sobre ocupada.

**Tiempo de Procesamiento:** > 2x CPU (CPU POWER7 es una excepción)  
Revisar en la transacción ST03 si todos los procesos de trabajo tuvieron el mismo problema. En caso afirmativo la causa del problema puede ser un cuello de botella en CPU, memoria o en semáforos. Para un registro estadístico puede existir varias razones por las cuales el tiempo de procesamiento es alto. Esta puede estar relacionada a RFC, sentencia WAIT en ABAP o bloqueos internos.



## Análisis de una transacción (ST12)

Como ya lo hemos mencionado en el pasado, esta transacción conjunta las transacciones para hacer trazas de ABAP (SAT/SE30), SQL, RFC, Búfer y enqueue (ST05). Esta transacción se actualiza con el Add-On ST-A/PI que es el componente de software que tiene las herramientas de servicio para Aplicación para los servicios de EarlyWatch y Going Live.

Antes de poder ejecutar trazas es bueno saber si hay consumo alto de CPU o Base de Datos, ya que en condiciones de inflexión del rendimiento las mediciones puede que no sean confiables.

La forma en cómo se debe ejecutar el análisis de transacciones se puede simplificar a la siguiente figura 7.4.1-9, donde tenemos el flujo de operación:

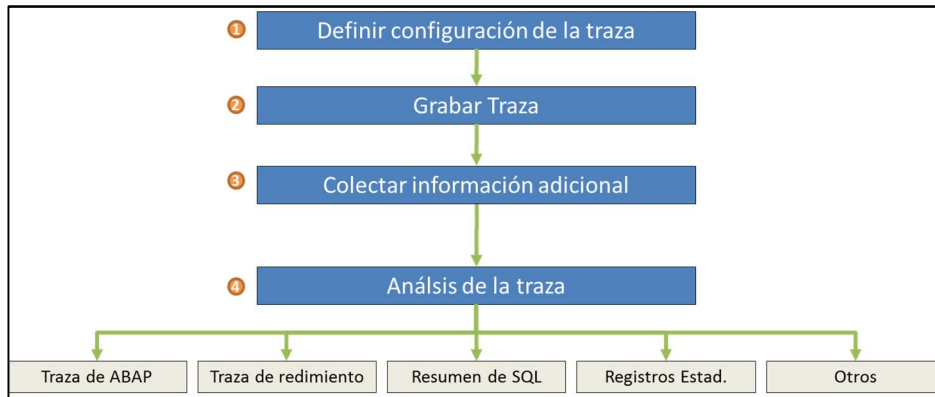


Fig. 7.4.1-9 Ejecución de una traza transaccional

1. Configuración. El primer paso es configurar técnicamente la traza que consta principalmente de poner la bandera para incluir tablas internas, aumentar el tamaño de la traza y solo si es necesario cambiar la resolución de medición a baja.

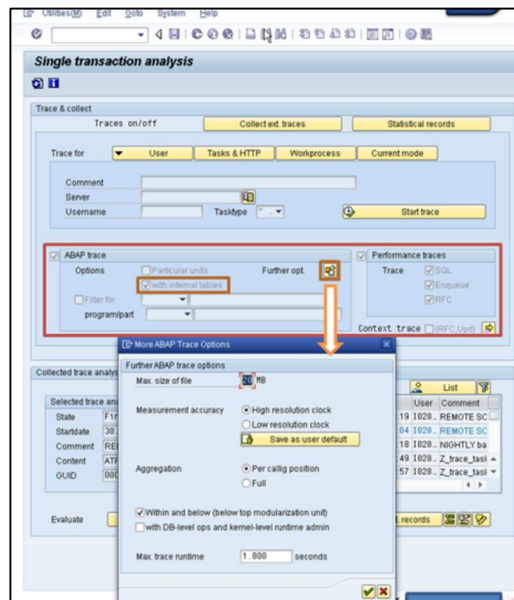


Fig. 7.4.1-10 Configuración transacción ST12

2. Grabación. Existen diferentes modos de ejecutar la traza y a continuación explicaremos cual se debe utilizar de acuerdo al escenario del cual queremos obtener información:
  - Modo Usuario. En este modo el usuario de negocios ejecuta uno o más pasos de diálogo en los cuales el consumo o rendimiento no es el adecuado o Si el usuario empieza o calendariza un trabajo de fondo sin ejecutar el trabajo en modo paralelo. En este modo el usuario debe

cerrar todas sus sesiones y solo dejar habilitada sobre la cual se tienen problemas de rendimiento. El problema de no ejecutar esta acción es que la traza puede capturar información no relacionada si el usuario está trabajando en algún otro proceso en paralelo. Se recomienda ya no utilizarlo.

- Modo por tareas y HTTP. En este modo se capturan las entradas de peticiones al proceso es por eso que se debe seleccionar un proceso de trabajo o todos los servidores de aplicación en caso de que se tengan varios servidores de aplicación así como peticiones de RFC y HTTP. Otro filtro útil es especificar el nombre de la tarea con HTTP, RFC o DIA y el nombre de usuario para filtrar peticiones.

Otro parámetro importante a especificar es el número de activaciones de traza por petición, la cual es útil si se está tratando con muchas peticiones de RFC, ABAP WebDynpro o BSP.

Los contenidos de ABAP y de SQL puede que no concuerden pues podemos tener varias trazas de ABAP ya que se activa por cada unidad lógica de trabajo (LUW) y la traza de SQL se inicia de inmediato.

- Modo por Proceso de trabajo. En este modo se captura información del tiempo de ejecución de una aplicación que ya está siendo ejecutada y por lo tanto solo la captura no va a incluir el tiempo desde que inició el programa (próxima acción después de la inicialización de la traza) hasta que iniciamos la grabación. Se utiliza para procesos que ya están siendo ejecutados y se tiene que seleccionar el servidor de aplicación donde está siendo ejecutado. Para iniciarlo se selecciona el proceso de trabajo en una ventana similar a la transacción SM50 y se presiona el botón con icono de cerillo. Los iconos amarillos indican que no ha empezado a grabar, mientras los verdes indican que la traza ya ha iniciado. Para terminarlo solo es necesario presionar de nuevo el botón con icono de cerillo apagado y regresar para coleccionar la información.

En caso de que se deje prendido, una vez que el proceso la traza de ABAP se termina y la de ST05 continúa con el filtro del proceso de trabajo.

Para la sentencias SQL puede que no capture la inicialización de una sentencia costosa (OPEN CURSOR) si se inicia la traza después de este comando así que hay que tener en consideración.

No.	ATRA	SQL	ENQ	RFC	Type	PID	Status	Reason	Time	User	Report	Action	Table	KB	Name
0					DIA	1316	Waiting							0	
1					DIA	3920	Waiting							0	
2					DIA	3284	Waiting							0	
3					DIA	5096	Waiting							0	
4					DIA	2708	Running			118 I028234_Z_ITAB_ALL				0	C:\usr
5					DIA	4512	Waiting							0	
6					DIA	4060	Waiting							0	
7					DIA	3096	Waiting							0	
8					DIA	2120	Waiting							0	
9					DIA	2008	Running			I028234 SAPLTHFB				0	
10					UPD	2096	Waiting							0	
11					ENQ	2916	Waiting							0	
12					BTC	4668	Waiting							0	
13					BTC	2924	Waiting							0	
14					BTC	3376	Waiting							0	
15					BTC	3848	Waiting							0	
16					SP0	332	Waiting							0	
17					UP2	2088	Waiting							0	

Fig. 7.4.1-11 Traza por proceso de trabajo

- Modo por ejecución propia. En este modo nosotros podemos mandar llamar o ejecutar la transacción o programa que deseamos analizar. Una vez que le presionamos el botón para empezar a grabar instantáneamente saltamos a la transacción que escogimos. Una vez terminada la ejecución solo hace falta regresar con a la transacción con el botón de regresar o F3 como atajo.

En este modo podemos utilizar la bandera de unidades particulares lo que nos permitirá utilizar los señalizadores en el OK-field /ron y /roff dentro de la transacción para grabar solo el paso de la transacción en el cual estamos interesados. Esta grabación está ligada a la transacción SAT y por tanto solo es relevante para la traza de ABAP.

Si lo que queremos es grabar la ejecución de una ventana emergente es necesario crear un atajo mediante el SAP GUI como se explica en el documento web publicado en SDN: <http://wiki.sdn.sap.com/wiki/display/ABAP/SAPGUI+shortcuts>

Un problema común con la transacción ST05/SAT es hacer una traza que incluya los módulos de función que se transmiten vía sRFC (tRFC/qRFC) no es siempre sencillo, pues se tienen que activar las trazas en varios sistemas. Con la ST12 se puede hacer de manera más sencilla utilizando la característica de "Context Trace" la cual también se utiliza para las trazas E2E y pone un tipo pasaporte al contexto para poder seguir sus movimientos a través de los procesos de trabajo. Esta funcionalidad está disponible desde el support Add-On ST-A/PI 01J\* y se activa con el parámetro rstr/accept\_remote\_trace = true que se puede modificar dinámicamente desde la misma transacción ST12. Es importante mencionar que se recomienda restaurar el valor a falso una vez se termine la traza.

Esta característica es útil para saber realmente en que parte de la llamada del RFC se consumió el tiempo de ejecución, pero como prerrequisito se tiene que buscar la traza colectando información del resto de los servidores de aplicación.

1. Colección de información adicional. Una vez que hemos terminado de grabar nuestra aplicación podemos recolectar información extra como trazas de sistemas remotos, registros estadísticos de la ejecución que realizamos (Ver siguiente sección) y un resumen de SQL que se crea asincrónicamente al finalizar la colección de información de ABAP.
2. Análisis de la traza. Finalmente queda el paso de analizar los resultados obtenidos de toda la información recolectada: traza de ABAP, de SQL, de rendimiento, registros estadísticos y otros. A continuación mencionaremos para cada uno los pasos a seguir para analizar la información sin ser objetivo explicar la optimización propia.
  - Traza de ABAP. El primer paso es ordenar la información por tiempo Gross que explicamos en la SAT, con eso podemos reconocer las unidades modulares que son costosas y podemos obtener una vista general que es lo que pasa por debajo de ellas. El segundo paso sería ordenar el resultado de la traza de ABAP por tiempo de NET en forma descendente lo cual nos muestra las líneas de ejecución más costosas en primer lugar. Se puede dividir el número de llamadas por el tiempo para obtener un promedio de tiempo por llamada. También podemos tener una vista jerárquica de las llamadas a unidades modulares para obtener un poco la lógica del programa.

En la figura vemos un ejemplo donde el 99.8% del tiempo está dentro del servidor de aplicación de ABAP. Notamos que tanto el tiempo de Gross como de Net la función de VALIDATIONS dentro del programa ZSAPF110V es la que más consume tiempo y por tanto es donde se debe centrar la investigación.

Call	No.	Gross	Net	Gross (%)	Net (%)	Program (called program)
Runtime Analysis On	0	261,901,102	0	100.0	0.0	
Perform VALIDATIONS	1	261,763,849	261,272,199	99.9	99.8	ZSAPF110V
Call Func. /BDL/_READ_JOB_STATUS	260	311,493	16,244	0.1	0.0	/BDL/SAPLBDL3
Select Single TBICO	261	296,412	296,412	0.1	0.1	/BDL/SAPLBDL3

Fig. 7.4.1-12 Resultado de la traza

- Traza de rendimiento. En este caso podemos proceder como se describe en la sección de la transacción ST05 ya que tiene las mismas características.
- Resumen de SQL. En este caso vemos un resumen con características similares al resumen de la transacción ST05 y podemos de igual manera saltar al código fuente y a la explicación de la sentencia de SQL donde se puede realizar un análisis más detallado.

- Registros Estadísticos. En esta podemos obtener información estadística recolectada por el kernel para nuestro proceso. Para el detalle ver la sección de STAD dentro de este capítulo.

## 7.4.2. Optimización en la capa de Base de Datos (Oracle)

En la arquitectura de SAP una parte vital del rendimiento es la base de datos. Este componente es difícilmente escalable por el tema de consistencia de información y generalmente representa el punto más delicado para asegurar un buen desempeño generalizado del sistema.

La base de datos es un punto centralizado de en los sistemas y es por eso que su rendimiento tiene que ser óptimo de lo contrario todo el sistema se ve afectado.

### 7.4.2.1. Introducción técnica

A continuación presentamos la arquitectura de la base de datos y conceptos esenciales para analizar un problema de rendimiento en una base de datos.

Si abstraemos la arquitectura de la mayoría de la Base de Datos moderna, no encontraremos con una arquitectura similar a la de la figura 7.4.2.1-1.

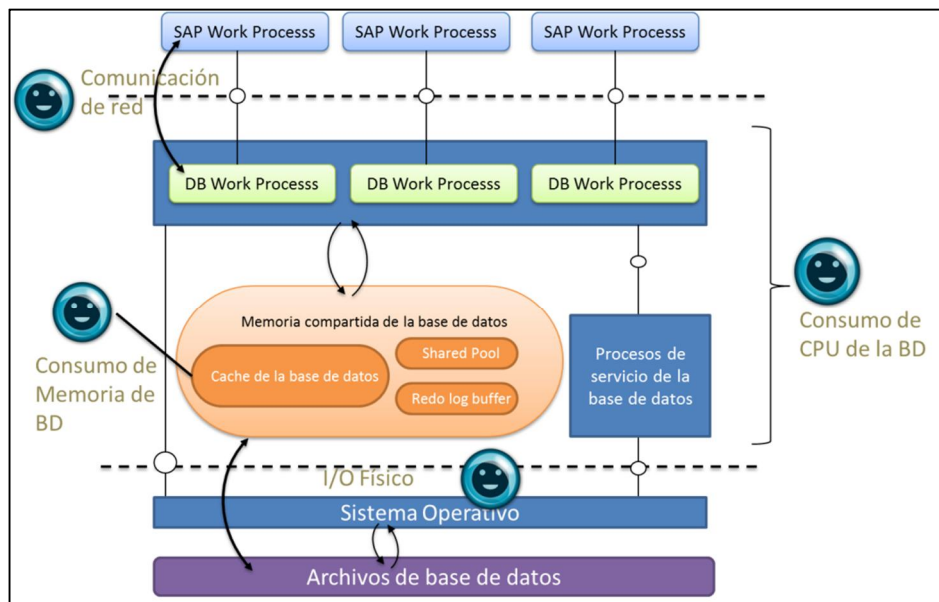


Fig. 7.4.2.1-1 Arquitectura de bases de datos modernas

Los recursos importantes en el servidor de Base de Datos que son limitados son la capacidad de CPU, la cantidad de memoria, el número de operaciones de I/O que se realizan de manera eficiente y la comunicación de red. A través de este capítulo iremos analizando cada una de las situaciones donde se ve disminuido el rendimiento de los recursos antes mencionados, pero antes como lo hemos venido trabajando daremos una pequeña introducción a los conceptos básicos necesarios para entender la optimización.

### Arquitectura de Oracle

Oracle se compone de su manejador de base de datos y la base de datos donde se almacena se información.

La base de datos básicamente consiste en archivos.

El manejador es el encargado de extraer, mantener, administrar y mantener esa información de los archivos en la base de datos de manera consistente, además de obtener esa información que se le solicita de forma eficiente.

En la siguiente sección se hablará concretamente del manejador de base de datos de Oracle debido a que una gran cantidad de sistemas ERP de SAP actualmente están instalados con este motor de base de datos.

Es importante también mencionar que algunos de los temas se extrapolan fácilmente a otros motores de base de datos como son DB2 LUW o Sybase.

### Base de Datos: Estructura Lógica y Física de la base de datos

Una base de datos es una colección de información en un o más archivos. La base de datos contiene estructuras físicas y lógicas. Las lógicas son las que se utilizan en general por las personas que consultan la información o los mismos desarrolladores de aplicaciones, es decir tablas o índices que guardan filas que ayudan a realizar búsquedas más rápidas. Estas se guardan en una estructura lógica llamada Tablespace (Oracle y DB2), esta estructura a su vez constan de uno o más segmentos (tablas/índices). Los segmentos están divididos en extents, que consisten en bloques continuos de Oracle. Un bloque de Oracle es la unidad lógica mínima de I/O usada por la base de datos. Para acceder a la información guardada en un bloque de Oracle varios bloques se tienen que leer o escribir, es decir se tiene que leer información para poder encontrar el lugar de ese bloque. Para los sistemas SAP, el tamaño del bloque que se recomienda es 8KB por el tamaño de datos que se manejan en la aplicación del ERP de SAP, y con esto el búfer de datos puede guardar más bloques con menos fragmentación interna. Las estructuras físicas son los archivos de datos llamados Datafiles que puede estar asociado a uno o más Tablespaces.

En la figura 7.4.2.1-2 de abajo se ve gráficamente la relación de los elementos mencionados anteriormente y como se conectan para formar un modelo básico de base de datos.

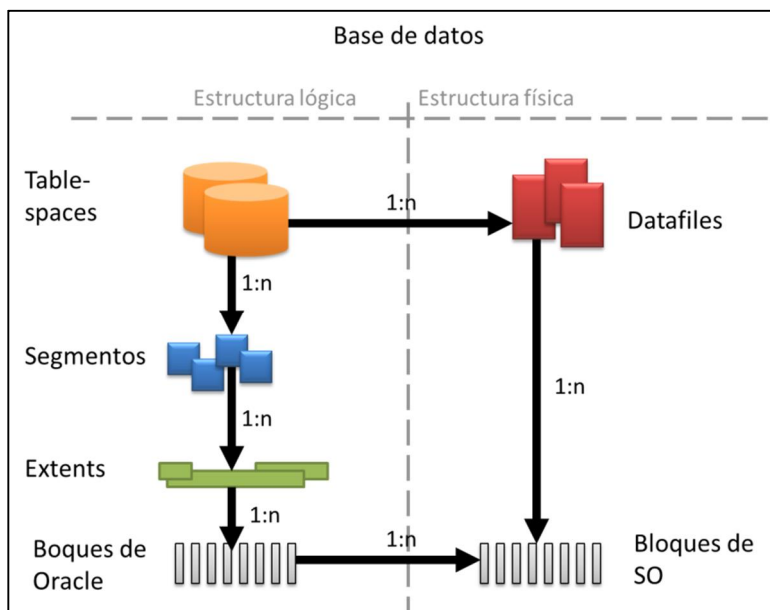


Fig. 7.4.2.1-2 Estructura Lógica y Física de la base de datos

Para obtener más información acerca de estas estructuras se tiene vistas en el tablespace del sistema:

dba\_tablespace, dba\_segments, dba\_tables, dba\_indexes, dba\_lobs, dba\_ind\_columns, dba\_objects, dba\_exents, dba\_data\_files

### Instancia de la base de datos en Oracle DB

Una instancia de base de datos consta de:

- Un área de memoria SGA, Área de Global del Sistema por sus siglas en inglés que sirve para evitar accesos a disco y operaciones redundantes
- Varios procesos de fondo que interactúan con el área SGA.
- Los archivos de la base de datos en el disco.

En Oracle DB existe una modalidad de alta disponibilidad llamada RAC en donde se tienen dos o más instancias de Oracle que usan la misma base de datos, cada una de estas instancias esta generalmente en servidores separados conectados por una conexión intranet de alta velocidad y que

tienen compartido los archivos en alguna SAN. El RAC de Oracle se utiliza para escalar también pues se tienen más servidores para realizar las actividades de la instancia, pero generalmente representa una buena carga de administración y conocimiento extra de la aplicación.

Los tres componentes principales en Oracle DB que comprende la arquitectura general como se muestra en la siguiente figura 7.4.2.1-3

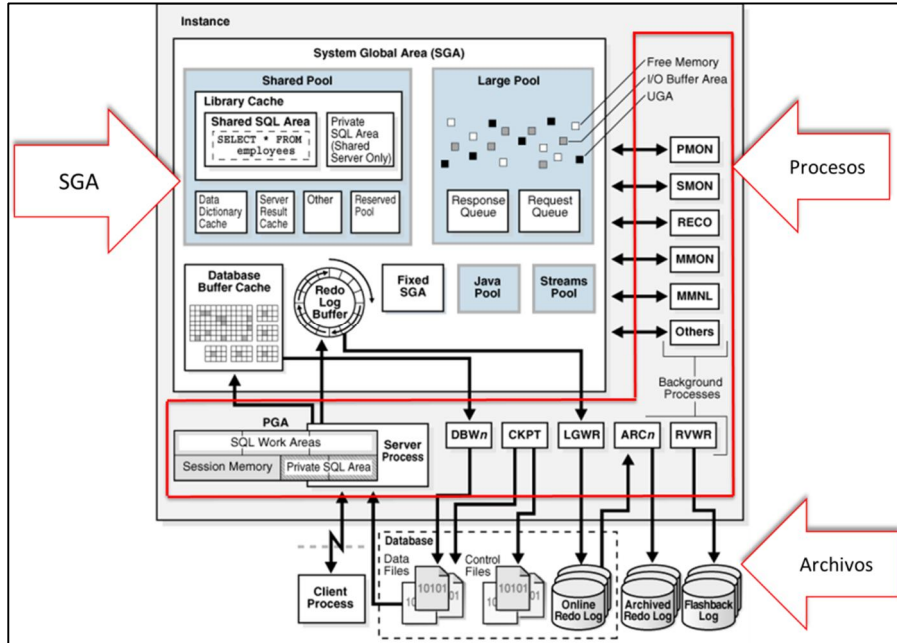


Fig. 7.4.2.1-3 Arquitectura de la base de datos Oracle

El problema principal de rendimiento en una base de datos relacional es poder obtener y mantener la información en ella de forma óptima, y es por eso que cuenta con algoritmos complejos para la estadística de la información y su forma de distribución dentro de sus contenedores. Los productores de las bases de datos relacionales no solo deben preocuparse por la persistencia y confiabilidad de la información, sino también de la forma de cómo pueden acceder estos datos almacenados con la velocidad necesaria.

Los recursos importantes en el servidor de base de datos que son limitados son la capacidad de CPU, la cantidad de memoria y el número de operaciones de I/O que se realizan de manera eficiente:

El uso inadecuado o innecesario de los buffers de la base de datos resulta en desplazamientos de otros bloques de datos y afecta el rendimiento del sistema en general.

Los desplazamientos de bloques de datos que se necesitan frecuentemente resultarán en una gran actividad de I/O físico y empeorará el rendimiento del disco. En la figura 7.4.2.1-4 podemos observar la distribución de recursos físicos como limitantes del desempeño con sus componentes lógicos a nivel de software.

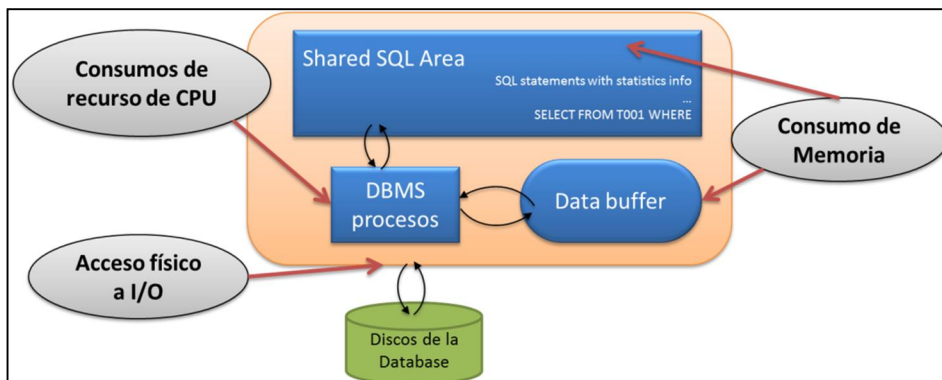


Fig. 7.4.2.1-4 Limitantes de recursos en las bases de datos

Mientras se procesan las peticiones al buffer los recursos de CPU también se utilizan.

La relación típica de llamadas de usuario del buffer de datos para un sistema ERP de SAP típico es de aproximadamente 1:15. Esto es el promedio de número de bloques que se tienen que leer para poder procesar una sentencia de SQL. En caso de que este ratio sea significativamente mayor, es un buen indicador que el buffer de Shared SQL se necesita analizar en busca de sentencias de SQL costosas.

En general hemos observado que llevar los problemas a un extremo siempre ayuda comprender mejor la situación ideal a la queremos llegar; Para esto imaginemos un ambiente perfecto, el cual no debe solicitar información fuera de la instancia de base de datos, es decir toda la información permanece en memoria todo el tiempo, en este ambiente no existe el tiempo de respuesta de I/O por acceso a disco que es en la mayoría de los casos es lo que más tarda porque el disco o la Flash de la SAN está lejos del CPU, entre 100,000 y 1,000,000 de ciclos de reloj, es decir el CPU está desocupado y no hace nada mientras espera para que se obtenga la información, es decir tiempo de espera de I/O, esto se traduce en tiempo libre desperdiciado por el procesador.

Sin embargo a menos de que trabajemos con una base de datos pequeña este escenario es muy poco común, aunque en la última parte de este trabajo incluiremos un concepto donde la reducción de costos de memorias en silicio hace que esta limitante pueda eliminarse en un futuro no tan lejano con la solución de SAP HANA que utiliza la compresión de datos, el almacenamiento en columnas y el almacenamiento en filas para crear un motor de base de datos especializado para las cargas de trabajo de los sistemas ERP de SAP. La realidad es que la mayoría de las instalaciones actuales cuenta con un sistema de discos magnéticos de donde debe obtener la información. Es decir, si la información no se encuentra en memoria, se debe realizar un acceso a disco, pero cuántos de los eventos de espera que dice que acceden a disco van realmente físicamente al disco, bueno pues eso depende de la infraestructura del centro de datos, si se usa cache en una SAN, tal vez se pueda evitar un porcentaje de esos accesos y aun así no sería tan rápido como si lo tuviéramos en memoria porque hay que tomar en cuenta también el tiempo de red y tiempo de recuperación de la información de cache dentro de la SAN.

Debido a las penalizaciones en rendimiento por el uso de I/O, es importante que el diseño para almacenar la estructura física de la base de datos Oracle (Datafiles) sea tomando en cuenta la siguiente recomendación:

- La distribución en el disco se debe hacer considerando capacidad de procesamiento de la unidad de almacenamiento o controlador y no el tamaño de los discos. Por lo tanto se considera una buena práctica el distribuir los tablespaces a través de todos los discos en donde sea posible. Para hacer esto, el tamaño de los archivos de datos para un tablespace debe de ser posible una constante de n gigabytes. Este tamaño puede depender de las restricciones del sistema operativo y la base de datos en combinación, sin embargo este tema es poco relevante cuando se hacen análisis a sistemas productivos debido a que en la mayoría de los casos cambiar esta estructura representa un proyecto grande con inversiones fuertes en equipo extra y tiempo de indisponibilidad por lo cual no voy a seguir profundizando, pero que en ocasiones resulta imposible darle la vuelta.

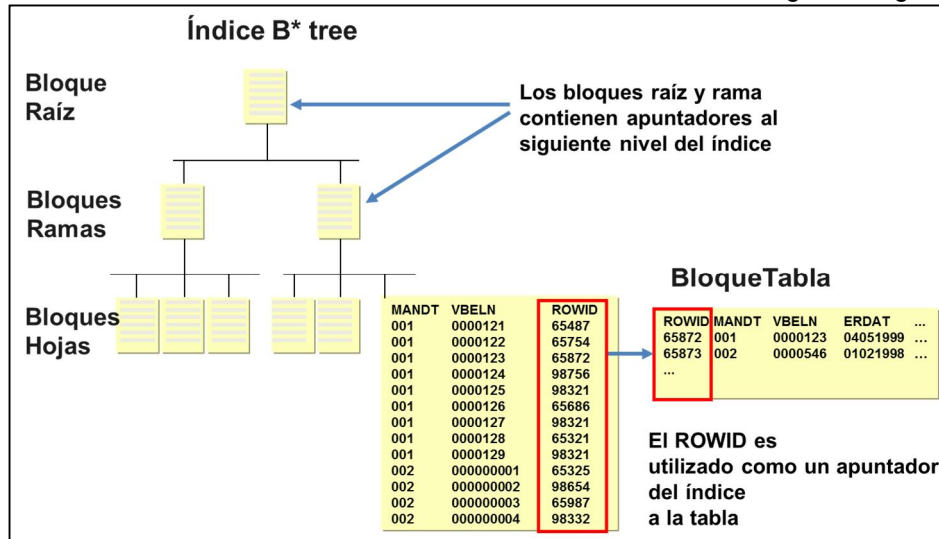
#### 7.4.2.1.1. Índices

Cuando se accede a la información de las tablas en Oracle, se puede hacer de dos maneras principalmente: leer todas las renglones de tabla (Full table Scan) y comparar la información para solo obtener la deseada, o acceder a un rango de filas ya ordenadas y comparar para obtener su número de identificación (ROWID).

Los índices son representaciones ordenadas de la información y es por eso que es más rápido encontrar la información, pero también más difícil de mantener.

Un índice se relaciona con una o más combinaciones de los campos de una tabla. A diferencia de las filas en una tabla, las filas de un índice están ordenadas y ese orden es definido cuando se crea el índice. Existe diferencia en tener un índice con las columnas C1, C2, C3 o con C3, C2, C1 si en la condición se especifican los valores de C2 y C3 pero no C1.

La mayoría de los índices que se define para el ERP de SAP en Oracle DB son B\*tree. Se construye una estructura arborescente usando los valores contenidos en los campos del índice. Los bloques Hoja del árbol contienen los valores de las columnas en las tablas indexadas y tiene un apuntador a las filas que coinciden en la tabla. Para poder encontrar una combinación de valores específica eficientemente los bloques Ramas ayudan a encontrar los bloques Hoja, esto reduce el número de pasos de búsqueda para leer un registro. Esta estructura definida en Oracle DB para los ambientes del ERP de SAP normalmente cuenta con 4 niveles como se muestra en la siguiente figura 7.4.2.1.1-1



**Fig. 7.4.2.1.1-1 Representación de un índice B-tree**

Los apuntadores en los bloques Hoja no son más que el ROWID que consiste del número de datafile de Oracle DB, el número de bloque en Oracle DB dentro del datafile y el número de fila dentro de ese bloque de datos.

Cuando se accede solo un pequeño porcentaje de una tabla, entonces se debe utilizar un índice. Por ejemplo para obtener información de solo el 10% de las filas en una tabla grande, si se utiliza un índice se requiere menos I/O para identificar los bloques donde se guarda la información. En caso de que no se use un índice se leerán todos los bloques de la tabla, incluso aquella que no se necesitan causando un uso innecesario de recursos.

El grado en que un índice puede ayudar al rendimiento está relacionado parcialmente con la selectividad de la información y la forma en cómo está distribuida en los bloques de la tabla. Si la información es lo suficientemente selectiva, entonces solo habrá pocas filas que cumplan con los valores del índice (como números de empleado, número de factura, o cualquier identificador numérico). Oracle DB es capaz de traer la información rápidamente consultando el identificador de fila (ROWID) correspondiente al valor del índice, y el pequeño número de bloques pueden ser seleccionados rápidamente. En cambio si la información no es muy selectiva, el resultado del índice pueden ser muchos identificadores de filas (ROWIDs), resultando en la búsqueda de muchos bloques separados en la tabla [VL912620].

Si la información es selectiva, pero aun así las filas relacionadas en información no están cerca una de las otras en la tabla, el beneficio del índice se reduce porque se tienen que consultar más bloques de la tabla, ya que la información se encuentra dispersa en ese conjunto de bloques. Hay casos en los que la información se encuentra tan esparcida en los bloques de la tabla que es mejor no utilizar el índice y hacer mejor un escaneo completo de la tabla. Cuando se hace un escaneo completo de la tabla Oracle DB usa una lectura de muchos bloques, permitiéndole leer la tabla completa de manera más rápida.

Las lecturas basadas en los índice son lectura de un solo bloque, entonces en conclusión es que cuando se utiliza un índice es porque se busca reducir el número de lecturas de un solo bloque.

Los índices en general aumentan el rendimiento de las consultas. SELECT UPDATE con la cláusula WHERE y el SELECT DELETE con igual la cláusula WHERE (cuando se acceden pocas filas), se puede beneficiar de los índices. En general la adición de índices va a tener una penalización en el rendimiento para las clausulas INSERT (ya que tanto la tabla como el índice se actualizan). El



UPDATE de columnas indexadas en general es más lento que el de las columnas sin índice, ya que de nuevo la base de datos tiene que actualizar tanto el índice como la tabla. El DELETE masivo de filas también se ve afectado por la presencia de índice en la tabla. Por ejemplo si se borra la mitad de las filas de una tabla, también se tienen que borrar la mitad del índice.

Entre más índices se tengan por tabla las cláusulas DML (Data Management Language) se ven más afectadas, es por eso que se tiene que hacer un balance de los beneficios de un índice contra el impacto que tiene en el rendimiento de la manipulación de la información.

El uso de índice también se puede dar por cláusulas que no tienen WHERE, pero que igual se benefician del índice, como pueden ser un MIN, MAX, COUNT en una columna indexada. Oracle DB en vez de traer el valor de la tabla lo obtiene directamente del índice, lo cual aumenta también el rendimiento de este tipo de cláusulas.

Con el siguiente comando se puede de obtener información de un índice en Oracle DB:

- `SELECT object_id FROM dba_objects WHERE object_name = '<index_name>';`
- `ALTER SESSION SET EVENTS 'immediate trace name treedump level <object_id>';`

Como resumen podemos tomar la siguiente síntesis:

Los índices pueden (pros):

- Ser un camino de acceso (acceso acelerado) para leer y cambiar información (UPDATE DELETE)
- Dar soporte para ordenar
- Dar soporte para agrupar
- Ser una proyección ordenada de la tabla original (solo se accede al índice)
- Implementar restricciones (por ejemplo candidatos únicos por llave)

Pero también pueden (contras):

- Incrementar el tiempo de insertar, borrar y actualizar la información
- Aumentar el consume de espacio
- Incrementar el tiempo de optimización
- Como regla general de rendimiento: Crear tantos como sean necesarios, pero los menos posibles.

El ciclo de vida de un índice en SAP dentro de un sistema OLTP (Online Transaction Processing), debe ser creado por los desarrolladores de SAP. Estos índices se incluyen y envían a cada cliente. Sin embargo algunos índices pueden estar inactivos y la activación depende de la manera en cómo se configura el proceso de negocio. Los índices pueden ser dependientes de la plataforma de base de datos.

El ciclo de vida de un índice en el sistema del cliente, es crear índices adicionales, los existentes se pueden borrar, excepto los índices únicos y en general son parte de una adaptación técnica del sistema.

#### **7.4.2.1.2. Optimizador**

El optimizador es una pieza del código del kernel del manejador de base de datos que determina el mejor camino de acceder la información solicitada. Decide si un índice debe ser tomado en cuenta o un acceso a toda la tabla es la mejor opción de hacerlo. En caso de que debe ser mediante un índice éste decide cual.

El optimizador considera lo siguiente [FM766349]:

- La cláusula SELECT
- La cláusula WHERE (con algunos valores)
- Índices en la tabla
- Las estadísticas en diferentes niveles de calidad

- Valores mínimos, máximos, densidad e histogramas
- 'Order by' casi nunca es considerado

El optimizador usa las estadísticas para obtener el mejor acceso a la información, basa sus decisiones principalmente en las estadísticas de los objetos a ser accedidos por ejemplo [FM766349]:

- Que tan grande son las tablas y los índices en bloques
- Cuantas filas tiene la tabla
- Cuantos valores distintos tiene una columna
- Que tanto se asemeja el orden de la tabla con el del índice

Las estadísticas se revisan constantemente y se recalculan si el nombre de filas cambia considerablemente, su función principal es guiar hacia el acceso con mejor rendimiento. Otro aspecto importante es que deben representar la situación real de las tablas y sus índices.

Los optimizadores basados en reglas ya no son utilizados por los motores de base de datos con SAP desde hace varios años porque son difíciles de mantener y requieren conocimiento amplio del esquema de datos que se maneja [FM772497].

Además de cómo se programa una sentencia de SQL, hay otro camino llamado **hints** para influenciar el plan de ejecución de un optimizador de la base de datos. Este recurso sólo se debe utilizar cuando han fallado todo, pues evita la fácil adaptación del software. En los hints se indicar cuál es el camino por el que se debe acceder a la información, la forma de un hint es como sigue:

```
SELECT * FROM MARA WHERE MATNR = '300-500' AND WERKS = '2100'
%_HINTS ORACLE 'INDEX("&TABLE&" "MARA~M" "MARA^M")'.
```

### 7.4.2.1.3. Procesamiento de sentencias de SQL

Antes de poder mejorar sentencias de SQL se necesita entender la forma en cómo se consulta esa información. SQL se puede decir que es Orientado a Conjuntos, porque si analizamos una sentencia de selección de datos SQL, podemos observar el universo de búsqueda, filtrado y consulta. En la sentencia siguiente, utilizamos los colores para que concuerden con la figura 7.4.2.1.3-1 y que sea más fácil identificar cada uno de los dominios dentro de la sentencia.

```
SELECT campo1, campo2, campo3
FROM tabla1
WHERE campo1 IN ('A', 'B') AND campo2 LIKE 'T%'
```

Teniendo un índice en: campo1 + campo2.

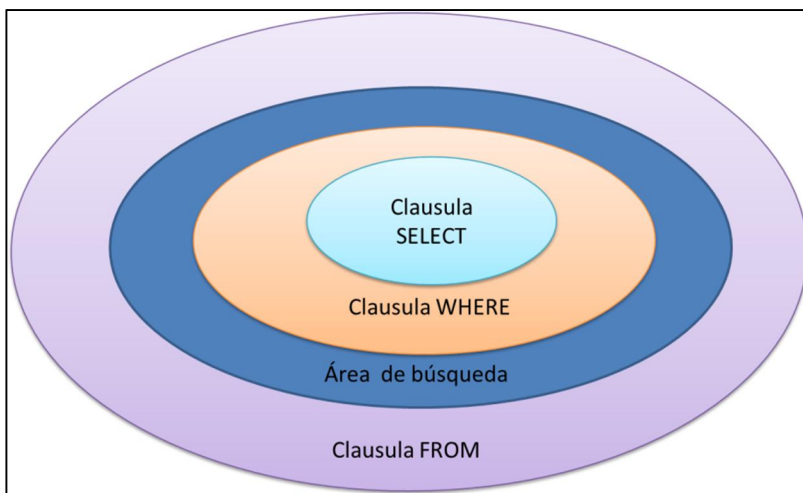


Fig. 7.4.2.1.3-1 Representación del universo de una sentencia SQL

Vemos que el primer conjunto del SELECT contiene la información que va a ser transferida (azul claro), el conjunto del WHERE es el conjunto con el resultado (naranja), el conjunto de búsqueda tiene

la información en la cual se debe buscar, limitada solo por el uso de índices (azul oscuro) y finalmente el conjunto FROM donde se encuentra el universo de tablas que pueden ser una o varias (lila).

En este orden de ideas, vemos porque los índices son un parte importante para ejecutar las búsquedas de datos en un conjunto de datos con menos elementos.

### Organización de las tablas y los índices

Cada una de las tablas tiene una llave primaria, ya sea definida por el usuario o por la base de datos y los índices utilizan esa llave como identificador de fila. En Oracle ese identificador es llamado ROWID y no solo identifica una fila sino que contiene un identificador de bloque que le indica en que bloque se encuentra esa información [FM766349].

Oracle DB procesa las sentencias SQL en dos fases como se muestra en la figura 7.4.2.1.3-2:

1. Fase de análisis de la sentencia (Parsing): El SQL es comparado con otros en el buffer de Shared SQL Area, si existe una coincidencia, la sentencia ya analizada se reusa.
2. Obtención de la información del buffer de datos (2) o de disco (3): Oracle DB localiza las tablas requeridas y carga los bloques apropiados al buffer de datos. Oracle DB entonces realiza las operaciones necesarias en la información. La fase de obtención de datos consisten en dos pasos: Abrir y Extraer (Open y Fetch). Una sentencia SQL puede generar múltiples operaciones de extracción dependiendo de la longitud de las filas y del número que se pidan.

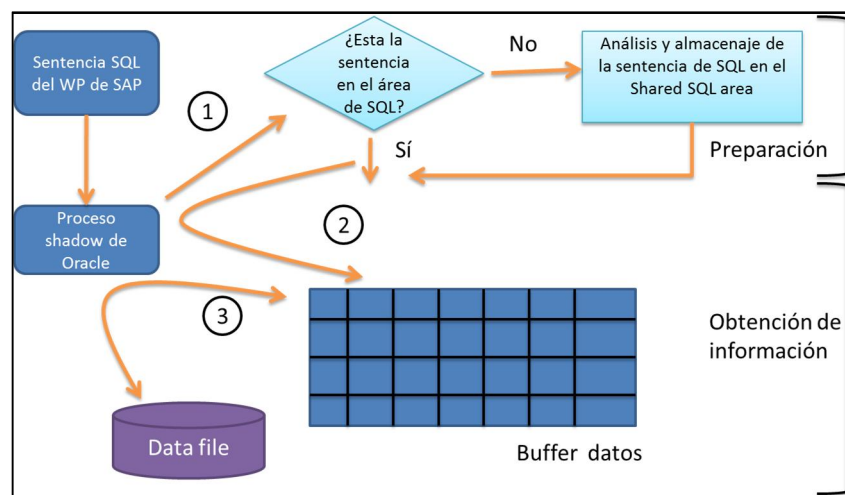


Fig. 7.4.2.1.3-2 Procesamiento de sentencias SQL

### Obtener Información para el análisis de performance.

Para poder obtener la información necesaria del comportamiento y arquitectura de Oracle DB, este cuenta con una serie de vistas con la cual ese puede consultar por medio de sentencias SQL. Estas vistas se les nombra V\$-Vistas que también se les denomina Vistas Dinámicas de Rendimiento. La información de las vistas proviene de las funciones de kernel de Oracle DB. Algunas de esas vistas son:

- V\$instance – Información referente a la instancia
  - V\$sppparameter – Consultar el archivo binario de configuración de parámetros de Oracle DB SPFILE.
  - V\$processes – Información relacionada con los procesos Shadow
  - V\$sessions – Información relacionada con los cliente conectados a los procesos Shadow
  - V\$sga – Información sobre la memoria compartida como el Shared SQL Area (V\$sql), Dictionary cache(V\$rowcache) y el Buffer cache (V\$bh).
- V\$database – Información sobre los archivos
  - V\$controlfile – Información sobre la organización de la base de datos está contenida en los Control Files

- V\$datafile – Información sobre los datafiles
- V\$log Información sobre los archivos de log

#### 7.4.2.1.4. Consumo de Memoria de Oracle DB

Las tres principales partes que consumen memoria en la arquitectura de Oracle DB en el ERP de SAP son el buffer cache, el shared pool y la memoria interna del proceso del servidor.

Existe una distinción entre las áreas de memoria compartidas que pueden ser llamadas por los procesos de Oracle DB y la memoria local de los procesos que solo puede ser asignada a un proceso.

System Global Area (SGA, Memoria compartida)

- Buffer Pool (Buffer de datos o cache): Buffer utilizado para guardar bloques de datos
- Shared Pool. Se compone de varios segmentos incluidos (“Shared SQL Area” o “Shared Cursor Cache” o “Library Cache”) los que tiene el pre-análisis de las sentencias SQL o información de DDIC (Cache del Diccionario o Cache de Registros)
- Java Pool: Buffer para java
- Large Pool: Buffer para información especial, por ejemplo cuando se utiliza un servidor Multi-threaded, RMAN utiliza varios procesos de I/O cuando se activa
- Redo Buffer: Es el buffer para la información de los redo log.

#### Program Global Area (PGA, memoria local del proceso)

- Este buffer se utiliza para ordenar, hacer hash joins, operaciones de bitmap y para requerimientos locales del proceso como puede ser el parsing de la sentencia.

La configuración de estos búferes varía en cada instalación y es por eso que es difícil dar un estimado que función en general, sin embargo analizaremos cuales son las recomendaciones generales para lo buffers que mencionamos arriba.

#### Buffer Pool:

Este es el buffer que más influye en el rendimiento de la base de datos. Un buffer de datos lo suficientemente grande significa que la base solo tiene que hacer pocos accesos costos en tiempo a disco.

El buffer de la base de datos esta implementado con el principio de Menos Usado Recientemente (LRU) por sus siglas en inglés, por lo cual los bloques menos utilizados son los que salen del buffer en casa de no haber espacio disponible.

Para medir que tantos datos obtenemos de este buffer, Oracle DB tiene un indicador llamado: “Buffer cache ratio” que indica porcentaje de numero de bloques que se pueden acceder desde el buffer de la instancia de Oracle DB sin necesidad de acceso al disco, sin embargo las estadísticas internas de Oracle DB no reflejan que esta métrica ayude siempre a clasificar a un sistema de base de datos como un sistema de buen desempeño. Este indicador se puede observar en la transacción ST04 (DBACockpit) de SAP y se recomienda que tenga un valor mayor iguala a 94%, sin embargo aun cuando un sistema tenga valores aún mayores a 94% no significa que no existan problemas de rendimiento en la base de datos, de hecho hay ocasiones que una sentencia SQL costosa que se ejecuta frecuentemente este causando Hit Ratio de 99%, pero el sistemas este en muy malas condiciones y el resto de sentencias tarden demasiado tiempo porque son desplazadas fuera del buffer y entonces tenemos mayor número de accesos a disco.

Existe una vista V\$DB\_CACHE\_ADVICE que estima la ganancia en tiempo perdido por leer registros de los discos por aumentar el tamaño de este buffer. Esta vista solo es informativa y se solo si se tienen los recursos físicos se debe aumentar el espacio del buffer.

Para aumentar el tamaño se usa el parámetro DBA\_CACHE\_SIZE.

#### Shared Pool:

Debe al menos ser de 400MB y menor a los valores de acuerdo a la siguiente fórmula (Mueller, 2007):

$$SPMAX = \min (( cpus / 4 * 350 MB + 300 MB ), 3 GB )$$

Donde el número de cpus se puede obtener de la vista v\$parameters con la sentencia:

```
select name, value from v$parameter where name in ('SHARED_POOL_SIZE',  
'CPU_COUNT')
```

#### **Redo buffer:**

Un buen valor para el Redo buffer es 1MB, el cual difícilmente debe cambiarse.

#### **PGA:**

La memoria requerida por todos los procesos se puede especificar con el parámetro PGA\_AGGREGATE\_TARGET que reserva el total de memoria para todos los procesos y va asignado de manera dinámica. En general se recomienda 20 % de toda la memoria para OLTP y 40 % para OLAP.

El resto de áreas de buffer son demasiado pequeñas si las comparamos y es por eso que en podemos decir que el consumo de memoria de Oracle DB es igual a la suma del Buffer Pool, Shared Pool, PGA y la memoria de los procesos.

El consumo del PGA se puede obtener de la vista V\$PGASTAT en el DBACOCKPIT y la memoria de los procesos de Oracle DB se puede obtener con la siguiente sentencia:

```
SELECT  
    MAX_UTILIZATION, MAX_PROCESSES,  
    <No. Processes> * MAX_UTILIZATION PROCESS_LOCAL_MEMORY_MB  
FROM  
    V$RESOURCE_LIMIT  
WHERE  
    RESOURCE_NAME = 'processes';
```

#### **7.4.2.2. Análisis del rendimiento de una sentencia SQL y método para la optimización de sentencias costosos [FM766349]**

Cuando se realiza la programación del proceso de aplicación, generalmente se crean sentencia de SQL con el solo propósito de obtener la información que se requiere procesar sin tomar en cuenta factores como el rendimiento y el acceso eficiente a esa información. Sin embargo es común encontrar que los sistemas están gobernados, en términos de rendimiento, por una decena de sentencias de SQL, las cuales producen hasta el 50% de la carga total.

No es parte de este capítulo el diseño de las aplicaciones per se; en vez de eso se revisará la forma de identificar, analizar y optimizar esas sentencias pre-existentes. Esto se vuelve relevante en sistemas que están en productivo y para los cuales el número de usuario aumenta.

En general podemos identificar dos tipos de sentencias costosas:

1. Sentencias SQL que no cuentan con acceso óptimo a la información
2. Sentencias SQL que obtienen un conjunto muy grande de información

Es a partir de esa clasificación simple como podemos proseguir y empezar un análisis más sofisticado.

#### **Identificación de sentencias**

Lo primero que debemos tener en cuenta es enfocarnos en sentencias que van a representar una ganancia significativa en rendimiento. Es importante evitar perder el tiempo en sentencias que no representarán una ganancia evidente en términos de rendimiento generalizado del sistema o que tiene una relevancia significativa para algún proceso de negocio.

Podemos utilizar principalmente una traza de SQL (transacción ST05) si es que nos interesa un proceso en específico o estadísticas de sentencias SQL en el cache si lo que buscamos es una mejora generalizada del sistema.

Las herramientas de SAP que podemos utilizar son las siguientes:

- SM66/SM50. Observamos procesos que duran mucho tiempo en una sola tabla. El siguiente paso es hacer una traza con la transacción ST05 para obtener las tablas y sentencias más costosas
- ST03. Podemos buscar información estadística histórica de las transacciones con tiempos de base de datos más altos y una vez más realizar trazas a esos procesos con la transacción
- ST05 para refinar la búsqueda de tablas y sentencias.
- ST04/DBACOCKPIT. Con esta transacción podemos obtener las estadísticas del motor de base de datos y podemos hacer estadísticas de las sentencias que causan más carga de trabajo al motor. Podemos organizar por el número de lecturas a disco, a buffer, el número de lecturas o ejecuciones entre otras.
- Sentencias de SQL para filtrar las sentencias más costosas como las que se encuentran en la nota de SAP 1438410 Note "SQL: Script collection for Oracle"

Una vez que se han identificado esas sentencias se debe decidir una estrategia para optimizarlas. Una forma es la modificación del código ABAP, otra es el agregar o modificar un índice para optimizar el acceso y otra puede ser la agregación de hint o parámetros para modificar el comportamiento del optimizador.

Para decidir esta estrategia primero hay que revisar la cantidad de información que se están enviando. Si esta cantidad es desproporcionalmente grande se tiene que revisar con el programador si la gran cantidad de información es necesaria. Es decir, si un reporte regresa millones de filas desde la base de datos, la persona encargada de ese reporte debe replantar una mejor estrategia de acceso a la información.

Si por el contrario lo que se quiere optimizar es una sentencia costosa de SQL que obtiene pocos registros, pero tiene que buscar la información en muchos bloques dentro de la base de datos, esto es un claro indicador de que la búsqueda de esa información es ineficiente. Este tipo de búsquedas debemos soportarlas con algún índice secundario nuevo o preexistente que no está siendo utilizado. Independientemente del tipo de optimización se debe obtener la siguiente información como prerequisite:

- Propósito del programa
- Las tablas, índices y sus contenidos. Tamaño de las tablas, si son de datos maestros, de configuración o de información transaccional. El tamaño de estas y sus características técnicas como factor de cluster y tamaño.
- El nombre del usuario dueño del programa y el último usuario en modificarlo.

Antes de comenzar un análisis más detallado se recomienda:

- Asegurarse que la consistencia entre el diccionario de ABAP y la base de datos sea consistente.
- Activar los índices secundarios en ABAP para que se creen en la base de datos
- Generar estadísticas cada vez que se genere un índice y periódicamente para tablas que sufren un cambio mayor al 10% de su tamaño.
- Reorganizar las tablas e índices que están fragmentados. Un índice se dice que está fragmentado si el radio de espacio ocupado y su contenido es muy bajo. Una tabla en la cual se borra mucha información puede fragmentarse con rapidez. Mediante la funcionalidad del análisis de tablas/índices en el DBACOCKPIT se puede encontrar aquellas tablas que necesitan ser reorganizadas.

Reglas para programar en Open SQL [GH2009]:

1. Mantener el conjunto de resultado reducido:

La única forma de lograr este objetivo es realizar reportes efectivos desde el punto de vista negocio y con rango adecuados de clientes, fechas, productos, etc. Es importante también que la cantidad de información que se conserva en la base de datos sea solo la necesaria y archivar el resto para evitar problemas de acumulación excesiva de información.

Mantener el conjunto de resultado reducido es la mejor estrategia para cumplir con este punto. Los puntos que hay que cumplir son:

- Reducir el número de columnas. Para reducir el volumen de datos, especifica solo las columnas que el programa realmente necesita. La figura 7.4.2.2-1 ilustra mejor esta mejora

<p><b>Select *</b> Runtime: 15.949 microseconds</p> <pre>SELECT * FROM DD01L INTO DD01L_WA WHERE DOMNAME LIKE 'CHAR%' AND AS4LOCAL = 'A'. ENDSELECT.</pre>	<p><b>Select with select list</b> Runtime: 4.087 microseconds</p> <pre>SELECT DOMNAME FROM DD01L INTO DD01L_WA-DOMNAME WHERE DOMNAME LIKE 'CHAR%' AND AS4LOCAL = 'A'. ENDSELECT.</pre>
--	--

Fig. 7.4.2.2-1 Reducir el número de columnas en la selección

- Reducir el número de filas. Siempre se debe restringir el volumen de información con una condición WHERE, especialmente para tablas de tamaño considerable. Las tablas de personalización y las que se tiene que procesar toda la tabla deben ser excepciones. El filtrado de la información se debe realizar antes de traer la información, de lo contrario este punto no se cumple.

<p><b>Select + Check statement</b> Runtime: 251 microseconds</p> <pre>SELECT * FROM SBOOK INTO SBOOK_WA. CHECK: SBOOK_WA-CARRID = 'LH' AND SBOOK_WA-CONNID = '0400'. ENDSELECT.</pre>	<p><b>Select with Where condition</b> Runtime: 260 microseconds</p> <pre>SELECT * FROM SBOOK INTO SBOOK_WA WHERE CARRID = 'LH' AND CONNID = '0400'. ENDSELECT.</pre>
---	--

Fig. 7.4.2.2-2 Reducir el número de filas antes de traer la información

- Agregar información. Utilizar las funciones de agregación propias de la base de datos ayuda a minimizar la transferencia de información si es lo que se pretende. En ocasiones observamos programas donde se obtienen múltiples registros con el único fin de ser agregados en ABAP, lo cual representa un desperdicio de recurso. Las funciones de agregación comunes son COUNT, SUM, MIN, MAX y AVG; La figura 7.4.2.2-3 muestra un ejemplo con los tiempos de ejecución.

<p><b>Select ... Where + Check</b> Runtime: 21.933 microseconds</p> <pre>DATA: MAX_MSGNR type t100-msgnr. MAX_MSGNR = '000'. SELECT * FROM T100 INTO T100_WA WHERE SPRSL = 'D' AND ARBGB = '00'. CHECK: T100_WA-MSGNR &gt; MAX_MSGNR. MAX_MSGNR = T100_WA-MSGNR. ENDSELECT.</pre>	<p><b>Select using an aggregate function</b> Runtime: 271 microseconds</p> <pre>DATA: MAX_MSGNR type t100-msgnr. SELECT MAX( MSGNR ) FROM T100 INTO max_msgnr WHERE SPRSL = 'D' AND ARBGB = '00'.</pre>
---	---

Fig. 7.4.2.2-3 Agregación de la información desde la base de datos

- Revisión de existencia. Otro ejemplo clásico para evitar la transferencia innecesaria de información es para saber si una tabla tiene registros para una condición particular. Es común que los programadores cuente el número de filas que cumple con la condición para ver si existe, lo cual es una forma ineficiente ya que no se necesita saber el número de filas, sino solo la existencia de ellas. Para evitar eso existe una sentencia UP TO 1 ROWS en ABAP para evitar la transferencia innecesaria de registros.

## 2. Reglas para reducir el número de lecturas de bloques [GH2009]:

Un índice por regla debe ayudar a reducir el número de lecturas de bloques.

Si se puede reducir los accesos a bloques de datos, se puede incrementar significativamente el rendimiento. Por cada bloque que tiene que ser leído, hay un riesgo de que no exista en buffer de la base de datos y por lo tanto se tiene que leer del disco.

Un buen índice debería reducir significativamente el área de búsqueda, reduciendo el número de bloques que se tienen que leer.

Un mal índice no reduce el área de búsqueda. La base de datos tiene que leer una gran cantidad de bloques y por lo tanto sacar bloques del buffer de datos. Adicional a este problema, se tienen que hacer chequeos en los bloques y solo después son las filas descartadas.

Idealmente el índice debe restringir el área de búsqueda a un tamaño cercano al conjunto de información del resultado.

Para reducir el área a ser escaneado secuencialmente por un índice se puede dividir en dos partes:

- La primera parte es una búsqueda binaria con una llave para empezar y terminar
- La segunda parte es una búsqueda secuencial con una llave como rango

Un buen índice por lo tanto permite a la base de datos reducir el área de búsqueda principalmente usando una estrategia de búsqueda binaria.

En un índice malo, una gran parte del índice tiene que ser escaneado secuencialmente, haciendo que se lean muchos bloques de datos y se deban ejecutar muchas validaciones en la información.

La experiencia nos enseñó que la correcta definición del orden de un índice se debe considerar tomando la selectividad de los campos y la utilización del índice para otras sentencias. Algunos ejemplos comunes en un ERP de SAP de campos son:

- No son selectivos: bukrs, gjahr, kokrs (incluir si están con =)
- Muy selectivos: belnr, matnr, kunnr, objnr (localizan rápido al optimizador al índice adecuado)

Los campos que se pueden especificar con un igual (=) o en una lista IN también son buenos al principio del índice, porque permite a la base de datos usar búsquedas binarias también en los siguientes campos.

No deben de existir brechas entre los campos de índice antes de que el rango a ser escaneado secuencialmente.

Poner el cliente (campo MANDT) en la primera posición es una buena práctica, porque permite hacer búsquedas binarias en los siguientes campos del índice y separa los mandantes en la base de datos reduciendo el número de bloques que se leen.

Algunos errores comunes son suprimir índices, lo cual se puede lograr si se utilizan las sentencias con:

- Operadores NOT EQUAL ('<>' '!='). Este tipo de operador no ayuda a filtrar las búsquedas. Un ejemplo muy simple es el directorio telefónico indexado por apellido. Si nosotros le decimos que queremos todos los números donde el APELLIDO <> 'Perez' y NOMBRE = 'JUAN' tendremos que buscar en todas las paginas estas condiciones.
- Operadores <, >, BETWEEN. Este tipo de operadores generalmente lleva a búsquedas de rangos. Estos operadores tienen la ventaja de tener un punto de inicio y fin, lo cual de alguna forma limita las lecturas en los bloques de las hojas de un índice. Sin embargo si el rango es demasiado amplio, también suelen ser muy costoso.
- Operadores LIKE. Este operado se utiliza para buscar patrones lo que puede generar problemas si usan sentencias del tipo APELLIDO LIKE '%P%', continuando con el ejemplo del directorio telefónico, se tendría que barrer todo el directorio para saber cuáles cumplen con esta condición. Cuando se tiene un comodín al principio tampoco es eficiente y su comportamiento en búsqueda es similar al operador "<>". Finalmente si se tiene el comodín al



final LIKE 'P%' es más eficiente y su comportamiento es similar a un BETWEEN ya que conoce el rango que tiene que buscar.

Una parte muy importante dentro de la optimización de sentencia SQL, es plan de ejecución que tiene la sentencia a ser analizada. Este plan de se puede obtener desde la traza de SQL o desde el SQL cache en la DBACOCKPIT.

El plan de ejecución es dependiente de la base de datos y cada una tiene diferentes formas de acceso, eso hace la optimización de sentencia única dependiendo de la base de datos con la que se trabaje.

En el plan de ejecución se puede encontrar el método de acceso y el índice que se utilizó. También nos da información sobre la cláusula WHERE y las columnas que se utilizaron para filtrar.

Lo que debemos realizar en este primer paso es identificar los campos de la cláusula WHERE con los índices de la columna que se están utilizando y ver si estas columnas tienen una buena selectividad.

En la figura 7.4.2.2-1 podemos obtener un buen ejemplo de unas sentencias de SQL y su plan de ejecución. El plan indica que se acceso mediante una búsqueda con índice en el punto tres (IXSCAN) y el número de columnas que se utilizaron para la búsqueda (Key Columns: 2 MAND+PERNR). La selectividad de la columna PERNR comparada con la cardinalidad de la tabla nos indica que se trata de un índice primario. La segunda acción de acuerdo al plan es el obtener el campo ENAME de la tabla directamente con la instrucción FECTH y finalmente en el tercer paso regresa el resultado al agente que lo solicito.

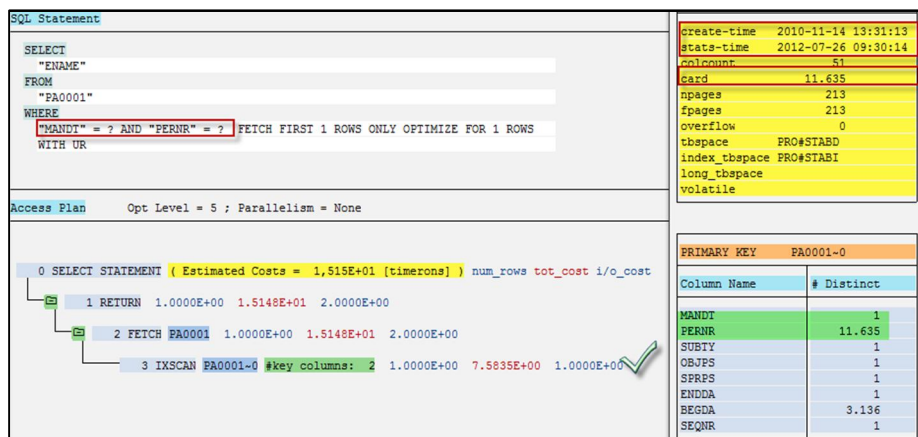


Fig. 7.4.2.2-1 Plan de ejecución de SQL

Para el análisis adecuado de selectividad del plan de acceso se necesita información adicional sobre las variables que se ocupan con el operador "=", si existe algún salto en los campos del índices que obligan a realizar index range scan, vale la pena revisar el diseño de ese índice.

Para eso contamos con herramientas como la transacción ST16 para navegar en las tablas y saber la cantidad de registros y la otra herramienta es la transacción DB05 para el análisis de tablas respecto a campos de los índices.

Un ejemplo de dicho análisis se muestra en la figura 7.4.2.2-2 donde se muestra el análisis para la tabla BSID con 5 millones de filas y las columnas propuestas para un índice que puede soportar una sentencia costosa. Las columnas más selectivas deben de actuar como último filtro del predicado.

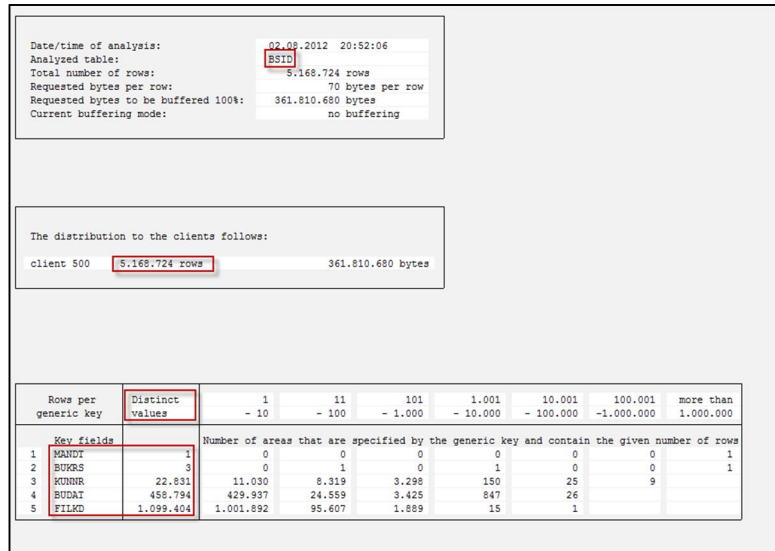


Fig. 7.4.2.2-2 Análisis de una tabla de acuerdo a sus columnas

### 7.4.3. Optimización en la capa de ABAP con programación eficiente

Utilizar estándares de programación e indicadores mejora la adherencia a las mejores prácticas de programación relativas a rendimiento. Estos estándares se pueden resumir en programación eficiente en la capa de persistencia, escalabilidad de las aplicaciones, red y dimensionamiento.

La programación eficiente se debe de aplicar como regla generalizada en los proyecto de implementación para el código que se desarrolle para el cliente. Esta debe brindar seguridad de un buen rendimiento y escalabilidad a la aplicación.

SAP sigue estos lineamientos con el fin de brindar software que sea fácilmente escalable y no necesite pruebas extensivas de regresión para las configuraciones estándares. Esto ayuda a minimizar las interrupciones y el costo de innovación con actualizaciones de software.

En la siguiente describiremos algunos de los indicadores que se pueden revisar manualmente para revisar que los estándares se están cumpliendo.

#### 7.4.3.1. Criterios de diseño para criterios de programación eficiente

Guías generales de programación [GH2009]:

- Capas de persistencia.
  1. Índices apropiados. Son indispensables para utilizar óptimamente el almacenamiento centralizado. Se necesita definir índices para soportar todos los accesos que se ejecutan con gran frecuencia en la capa de persistencia. Para poder verificar de manera estática que los índices adecuados se puede utilizar la transacción SCI (Inspector de código), ejecutar una traza con la transacción ST05.
  2. Clausular WHERE completas. Solo información que se necesita se lee de la capa de persistencia, por ejemplo la mayor parte de los accesos a la capa de persistencia tienen que estar completos con una cláusula WHERE que permita filtrar.
  3. Búferes y Caches. Para minimizar aún más los accesos a la capa de persistencia, se deben utilizar búferes y caches para datos de configuración, personalización y catálogos pequeños.
  4. Evitar accesos idénticos a la capa de persistencia. Mantener el mínimo número de accesos a la capa de persistencia, no deben existir accesos idénticos en el mismo paso de la transacción de negocio.
- Capa de aplicación.
  1. Procesamiento en paralelo. Para habilitar procesamiento paralelo, se deben emplear los mecanismos de bloqueo y de balanceo de cargas, además de saber crear paquetes bien

particionados con los tamaños adecuados. La forma de verificar que se está utilizando bloqueos se puede realizar mediante la transacción ST05 con una traza de colas y de SQL. La traza de colas nos muestra las peticiones y liberaciones de bloqueo, mientras que la traza de SQL no muestra las operaciones INSERT, DELETE y UPDATE seguidas por un COMMIT WORK.

2. Dependencia Lineal.
  - a. La memoria que requiere una aplicación depende linealmente en la cantidad de información procesada. No deben de existir fugas de memoria, las aplicaciones con fugas de memoria se van comportando más lentas con el tiempo y causan problemas con grandes cantidades de usuarios o con volúmenes altos, se puede utilizar la transacción `_S_MEMORY_INSPECTOR` para comparar el uso de memoria en instantes diferentes y ver si el existe algún componente como una tabla interna, objeto o instancias de ABAP Objects que está causando la fuga.
  - b. El tiempo de procesamiento depende linealmente de la cantidad de información procesada. Una forma de poder analizar si nuestro programa cumple con esta características se puede hacer utilización de la transacción SAT y comparar los resultados con diferentes volúmenes de información, si el resultado no es linealmente proporcional o  $n \cdot \log(n)$  que es aproximadamente  $2n$  se tiene que revisar las partes de la traza que contribuyen a este comportamiento no lineal

- Comunicación. Dos viajes redondos por paso de interacción con los usuarios.
  1. Para rendimiento adecuado en WAN, no deben de existir más de dos viajes redondos síncronos por la red entre la capa de aplicación y el cliente por cada paso de interacción con el usuario. Para obtener el número promedio se utiliza la transacción STAD en la columna de número de viajes promedio y sumarlos para todos los que tienen como tipo de tarea "Dialogo" y dividir esta suma de por el número de pasos, por ejemplo en la transacción VA01 se ejecutan 25 viajes redondos en total con 14 pasos de pasos de interacción lo cual resulta un valor de 1.8 que cumple con el requerimiento.
  2. Para rendimiento optimizado de servidor a servidor debe de haber un promedio de dos viajes entre los servidores o instancias por cada paso de interacción de usuario.
- Otros
  1. Tiempo de respuesta punto a punto menor a 2 segundos. Solo es aplicable si es para el tiempo de respuesta de dialogo.
  2. Procedimiento de dimensionamiento disponible. Existe un procedimiento de dimensionamiento de hardware para consumo en disco, CPU y memoria.
  3. Comentarios sobre las posibles desviaciones de los estándares. Si existe alguna excepción se debe de documentar la causa para saber de antemano las razones.

#### Principales indicadores

- Capas de persistencia. Numero de accesos a la capa de persistencia y valor neto de transferencia de información en KB. Se puede calcular de manera similar a los viajes redondos por paso, pero usando sumando las columna "Terminal-In" y "Terminal-Out" de la transacción STAD entre el número de pasos de las tareas de dialogo de la transacción.
- Capa de aplicación. Consumo pico de memoria y consumo de CPU de funciones y métodos (micro segundos)
- Comunicación. Número de viajes redondos entre el servidor de aplicación y el cliente, la cantidad de Kilobytes transmitidos en el cliente y la capa de aplicación.

#### 7.4.3.2. Búferes para de la información y manejo de memoria

La información que se necesita múltiples ocasiones debería como mejor práctica estar en memoria dentro de un búfer dentro de la arquitectura general de la solución. La arquitectura de memoria es tema complejo que se extiende más allá de la simple administración de la aplicación. Es conveniente que los desarrolladores de ABAP entiendan estas áreas de memoria para conocer las diferentes opciones de utilización de búferes de memoria dentro del sistema SAP [FHW2009].

El proceso de trabajo de SAP tiene una gran cantidad de áreas de memoria que se pueden acceder, las que se muestran en blanco en la figura 7.4.3.2-1 pueden ser manejadas directamente por un desarrollador de ABAP. El proceso de trabajo tiene acceso al área de memoria extendida vía el área de mapeo donde guarda los apuntadores a esos objetos y así evita el copiar esos objetos con datos. Cada usuario tiene una cuota para el uso de esta memoria extendida que por lo general es de bloques de 1MB a 4MB y se define con el parámetro *em/blocksize\_KB*.

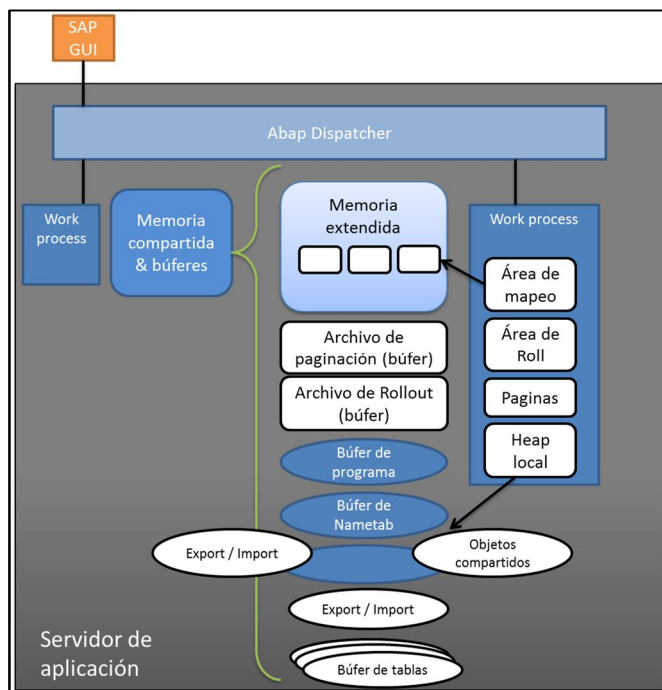


Fig. 7.4.3.2-1 Memoria en el servidor de aplicación ABAP

Fuera del control del programador existen otros búferes como el nametab, el cual contiene información del diccionario de datos, y el búfer de programa que contiene los programas ejecutables de SAP. Si por alguna razón estos búferes están muy pequeños, esto se puede detectar por medio de la transacción ST02 como se describió en la sección 7.3.1.3 de este trabajo, y por tal motivo no serán el detalle de esta sección.

Desde la perspectiva del desarrollador de ABAP existen áreas de memoria específicas de usuario y áreas comunes de usuario en las que se puede guardar información en memoria. Esto está estrechamente ligado con las sesiones del usuario.

Desarrolladores de aplicaciones directamente controlan el contenido del búfer export/import (SHM) y del búfer de tablas. El uso eficiente de estos dos recursos significa que el programa es robusto para la arquitectura cliente-servidor.

El búfer de export/import se usa con la palabra clave de ABAP *export to /import from shared buffer* y por el contexto.

El búfer de export/import SHM se usa con la palabra clave de ABAP *export to/ import from shared memory*.

El búfer de tablas mantiene información de las tablas de aplicación a las cuales se les ha configurado esta característica por el desarrollador de aplicación.

A continuación exploraremos los diferentes métodos de utilizar los búferes dentro de SAP.

#### **Búferes de usuario específico [GH2009]:**

En la sesión interna, la información se puede guardar en memoria (búfer) con variables. Dentro de la memoria específica de usuario, ninguna otra sesión puede leer esta información. Es por eso que la información esta automáticamente protegida de que alguna otra sesión la pueda consultar.

La información en este búfer se mantiene en una sesión interna. Este búfer se debe ocupar si no se puede utilizar tablas internas y si se quiere evitar múltiples accesos a la información (por ejemplo a la base de datos). La información en esta sesión interna se le puede llamar sin ninguna carga adicional por copiarla si se implemente adecuadamente. En la sección 7.4.3.4 se describe un poco más a detalle el acceso a tablas internas que están libres de una sobre carga por un proceso de copiado interno.

Aunque el utilizar el búfer de la sesión interna significa el acceso más veloz a la información, también significa que si múltiples usuarios trabajan con esta aplicación maximizan el uso de memoria. Es por eso que si la información la requiere más de un usuario, se necesita escoger otro nivel de búfer.

Para el intercambio de datos entre dos sesiones internas, se puede transferir información por medios de una interfaz (call parameter, selection screen, etc.) o usar la memoria de ABAP.

Si se utiliza la sentencia de ABAP *EXPORT TO/IMPORT FROM MEMORY* se puede exportar o importar información a la memoria de ABAP, localizada en el área de paginación de ABAP.

Esto tipo de memoria fue diseñado para la transferencia de conjuntos de datos pequeños ya que fue pensada solo para configuración específicas de los usuarios o valores de parámetros. Al requerir un proceso de copia para cada proceso que pide la memoria, el consumo de CPU y memoria se pueden disparar por su mal uso. Si se necesitan conjuntos de datos más grandes se debe utilizar un búfer a nivel del servidor de aplicación.

#### **Búferes comunes [GH2009]:**

Dentro de la memoria común, múltiples usuarios tienen accesos compartidos al mismo conjunto de datos, y se copia directamente al área de memoria del usuario o se trabaja directamente en el área común (Objetos compartidos). Para cambios dentro de esta área de memoria la aplicación tiene que asegurar la consistencia de información.

Los desarrolladores cuentan con los siguientes búferes:

##### 1. Búfer compartido (shared buffer)

Se utiliza para guardar información a nivel del servidor de aplicación utilizando el comando de ABAP *EXPORT TO/ IMPORT FROM SHARED BUFFER* se puede guardar u obtener información en este búfer. También cuenta con una sentencia de ABAP para borrar el contenido *DELETE FROM SHARED BUFFER*. Esta información es visible a todas las sesiones de usuarios de un servidor de aplicación. Este búfer se localiza en memoria virtual y se maneja por el algoritmo de menor uso (LRU), por lo cual si el servidor esta corto de memoria puede causar desplazamientos de información de la memoria que se pueden analizar en la ST02.

##### 2. Memoria compartida (shared memory)

También se utiliza para guardar información a nivel del servidor de aplicación. Se utilizan las sentencias *EXPORT TO/ IMPORT FROM / DELETE FROM SHARED MEMORY* para tener acceso a este búfer. Este búfer se localiza en el memoria virtual a nivel OS y no tiene un algoritmo de desplazamiento, por lo cual a falta de memoria se genera una excepción en ABAP "EXPORT\_NO\_SHARED\_MEMORY", por lo cual es muy importante la utilización de la sentencia de borrado para no dejar objetos innecesarios en memoria. No existe sincronización entre los servidores de aplicación por lo cual queda a carga del desarrollador de aplicación el realizar esta tarea.

##### 3. Objetos compartidos (shared object)

Solo se puede utilizar para guardar instancias de objetos de ABAP y se pueden leer por medio de referencias o apuntadores por lo cual no se necesita un proceso de copiado siendo esta una de la mayores ventajas. Este se reserva para objetos grandes y complejos, como regla general deben de estar en un rango de 200 a 500 KB. El área de objetos compartidos puede ser considerado como un modo interno adicional al cual la sesión de usuario se puede conectar directamente sin necesidad de copiar información, sin embargo se paga un precio que principalmente consisten en el tiempo necesario para establecer la conexión (primer "attach\_for\_read") y el incremento en tiempo de ejecución para tener acceso a los objetos compartidos.

##### 4. Búfer de SAP para tablas (SAP table buffer) analizado en la siguiente sub-sección

**Búfer de SAP para tablas [GH2009]:**

Existe otro tipo de búfer que se utiliza comúnmente en ABAP y que está relacionado con tablas de base de datos dentro de la aplicación de SAP para obtener un mejor rendimiento y disminuir la carga del recurso centralizado de la base de datos. Este se considera como un búfer de memoria compartido y es un mecanismo principalmente de lectura. Se tiene que habilitar individualmente para cada tabla.

Al estar en memoria compartida del servidor de aplicación se tienen múltiples copias de esta tabla si el número de servidores de aplicación es mayor igual a dos. La sincronización entre los servidores se hace a intervalos fijos de forma asíncrona. Se utiliza la tabla DDLOG para la sincronización de los cambios de las tablas en búfer.

El desarrollador necesita crear sentencias de Open SQL para poder tener acceso a este búfer. Si se realiza una sentencia de selección de información en una tabla con esta característica, se busca la información primero en este búfer y solo se accede a base de datos cuando no existe la información requerida.

Los registros de una tabla en este búfer siempre están ordenados por la llave primaria. Existen 3 tipos de configuración para las tablas son búfer completo, genérico y de un solo registro.

Se recomienda un búfer completo si la tabla cumple con:

- Tablas con tamaño de hasta 30 KB. Si la tabla se accede frecuentemente, y todos los accesos son de lectura el valor anterior se puede sobrepasar.
- Las tablas donde frecuentemente se consulta la existencia de un registro. En este caso se utiliza un búfer completo ya que se tienen todos los registros de información sin necesidad de ir a la base de datos.

Se recomienda un búfer genérico si la tabla:

- Es dependiente de cliente
- Es dependiente de lenguaje

Se recomienda el búfer de un solo registro si la tabla:

- Es una tabla grande en la cual solo unos pocos registros se consultan. El tamaño de los registros que se obtienen deben de estar entre 100 y 200 KB.

En un sistema productivo existen tablas que rara vez cambian como son las tablas de control, de configuración o tablas pequeñas de datos maestros. Se tiene que evaluar si la tabla es candidata a un búfer, por ejemplo no se debe poner esta característica si la información tiene que estar siempre actualizada ya que pueden existir problemas con la consistencia de información durante el periodo de sincronización y no se activar si la sobre carga de esta sincronización es mayor a 1% del total de los cambios. Estas invalidaciones son causadas por la sincronización del búfer y se pueden monitorear con la transacción ST10. Para las tablas con búfer genérico las columnas ABSENT, DISPLACED o ERROR indican que la tabla no está en el buffer en ese instante. Para tablas de búfer de un solo registro la columna ERROR señala una falta de espacio en el búfer. La columna de INVALID indica que el contenido de la tabla cambio y que existen transacciones abiertas referentes a este cambio.

Existen sentencias ABAP que puede evitar este búfer como son *SELECT ... BYPASSING BUFFER*, „*SELECT ... FOR UPDATE*, funciones de agregación (*COUNT*, *MIN*, *MAX*, *SUM*, *AVG*), *GROUP BY*, *HAVING* „*SELECT DISTINCT ...* , „*WHERE ... IS [NOT] NULL*, *WHERE ... COL1 <cond> COL2*, *ORDER BY* (diferente a la LLAVE PRIMARIA), „*Joins (SELECT ... JOIN ON ...)* y Sentencias nativas de SQL.

La herramienta de monitoreo de este búfer son:

- ST05 para grabar los accesos a la base de datos o al búfer
- ST10 para monitorear el estatus de la tablas en el búfer
- DB05 para hacer un análisis de las tablas candidatas a colocar en el búfer con el tamaño y la selectividad.

### 7.4.3.3. Tablas internas en SAP

SAP como servidor de aplicación sirve para ejecutar funciones o pasos de negocio con la información que se obtiene de la base de datos. Es una práctica común obtener la información necesaria de la base de datos y ponerla en tablas internas de SAP para poder manejar de una manera más segura y eficiente.

Las tablas internas son objetos internos de ABAP que permiten guardar conjuntos de datos dinámicos en memoria. Se pueden comparar de cierta forma con arreglos de estructuras y le ayudan al programador a evitarse el manejo de la memoria. La información de las tablas internas se maneja por medio de filas, donde cada fila tiene una estructura que es definida por el programador.

Mayormente se utilizan para almacenar y modificar conjuntos de datos de la capa persistente. Las tablas internas, al igual que las de una base de datos, presentan un problema de rendimiento si su acceso no es eficiente, sin embargo a diferencia del acceso a tablas de base de datos la afectación de rendimiento se refleja en sólo CPU o memoria que siempre se puede escalar con la capa de aplicación.

Una de las características más importantes de ABAP son las tablas internas. Las tablas internas sirven como medio para tomar información de una estructura fija y ponerla en un área de trabajo en memoria de ABAP. La información se guarda línea por línea en memoria, cada una teniendo la misma estructura. En ABAP se puede decir que las tablas internas cubren las funciones de los arreglos. Un uso importante de las tablas internas es para guardar y manejar la información de una tabla de la base de datos dentro de un programa [KHT2009].

Si las comparamos con otros lenguajes de programación sería lo equivalente a un vector de estructuras en C++ o un vector de objetos en java, con la diferencia de que ABAP tiene una serie de sentencias que permiten fácilmente acceder y manipular el contenido de las tablas internas.

La organización de las tablas internas en memoria, al igual que en una base de datos, está por bloques o páginas. Cuando declaramos una tabla interna en un programa de ABAP, el sistema solo crea una referencia en la memoria principal. Es cuando se le agregan registros es que el sistema crea la cabecera de la tabla y el cuerpo de la misma. La cabecera contiene una referencia a la primera página del cuerpo y una referencia al manejo de las páginas el cual se encarga de mantener las direcciones en la memoria principal.

El encabezado de la tabla incluye información importante de la tabla, por ejemplo el número de filas el cual se puede consultar con la función de ABAP `LINES <itab>`.

Cuando se declara la tabla se puede agregar el tamaño inicial para saber cuánta memoria se tiene que reservar si la tabla es muy pequeña y no hacerlo con el tamaño estándar que puede ser un desperdicio en tablas muy pequeñas.

Una vez que terminamos de utilizar estas tablas internas tenemos dos opciones para limpiar la memoria. La primera es `FREE itab` que resulta en la liberación completa del cuerpo de la tabla, es decir de todas las páginas y se agrega a una lista de bloques libres para su reuso. No obstante no se recomienda si la tabla va a ser utilizada nuevamente. La segunda opción es utilizar el comando `REFRESH` o `CLEAR` el cual evita la creación de la primera página una vez que se vuelve a utilizar esta tabla.

#### Atributos de las tablas internas

Las tablas internas siempre tienen 3 atributos principales que son: un tipo de línea, una llave y un tipo de tabla.

El tipo de línea puede estar basada en estructuras, objetos DDIC o campos simples de un objeto de datos.

La llave se define en el programa o en el diccionario de datos (DDIC). Esta identifica los componentes y su secuencia así como la unicidad de la llave.

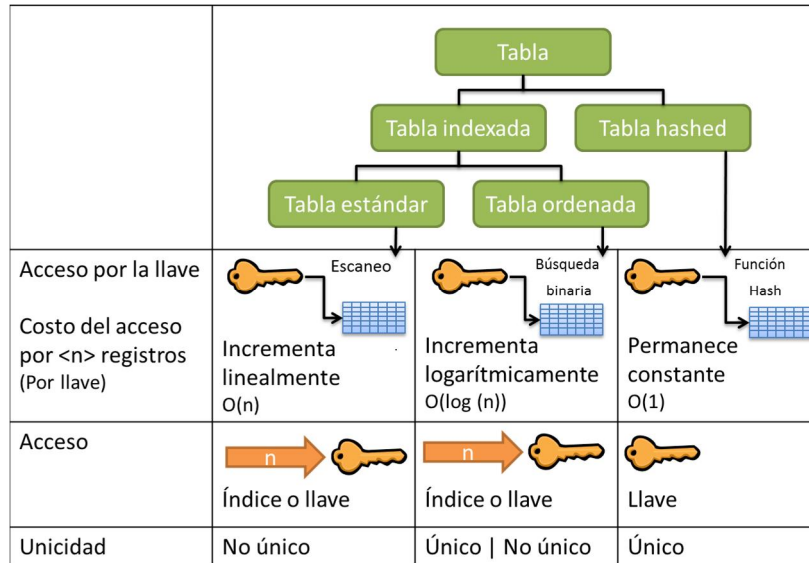
Los posibles tipos de tablas son tablas estándar, ordenadas y hashed. Cada tipo se describirá con más detalle en el siguiente párrafo [KHT2009].

**Tipos de tablas internas**

El tipo de tabla simplemente especifica cómo se puede acceder filas individuales de las tablas a través de ABAP.

Una tabla se puede acceder entonces por una llave o por un índice del registro. Las tablas indexadas se dividen en tablas estándar y ordenadas, que permiten el acceso vía índice o llave. Las tablas hashed sólo permiten el acceso por medio de una llave.

En la figura 7.4.3.3-1 podemos observar los tipos de tablas internas y sus características principales.



**Fig. 7.4.3.3-1 Tipos de tablas internas [GH2009]**

Las tablas estándar el acceso a cada uno de sus registros puede ser por un índice en la tabla o por medio de una llave. La llave de una tabla estándar no es única, por lo tanto si no se especifica una llave en la definición, se toma como default la combinación de todos los campos de tipo carácter. Para un acceso con llave el tiempo de respuesta depende linealmente del número de registro en la tabla interna al ser un escaneo lineal y por eso tiene una notación asintótica de O(n).

Las tablas ordenas siempre están ordenadas de acuerdo a una llave. El acceso a sus filas puede ser por medio de la llave o el índice. Sin embargo cuando se utiliza la llave, el tiempo de acceso es logarítmico dependiendo del número de entradas, ya que la búsqueda es binaria y eso hace que el rendimiento de acceso a estas tablas sea más eficiente. La llave para las tablas ordenas puede ser única o no única.

La lectura a tablas hashed solo es posible si se especifica completamente la llave, por lo cual el tiempo de acceso es constante y no depende del número de registros de la tabla, esto es porque se utiliza un algoritmo hash. Si la tabla se accede de algún otro campo que no sea la llave, seguramente la tabla se manejará como si fuera una tabla estándar y se tendrá que leer linealmente por todos los registros.

**Definición en ABAP de las tablas internas.**

Para declarar una tabla interna en ABAP se utiliza la siguiente línea:

```
DATA: i tab1 TYPE STANDARD|SORTED|HASHED TABLE OF sfl ight
      WITH [NON-]UNI QUE KEY carri d.
```

Se pueden utilizar tipos de tablas locales o globales. Dentro de la definición si se omite la declaración del tipo de tabla, siempre la tabla será creada como estándar. Si por otro lado se omite la llave, se crea con la llave default del tipo de línea o si no tiene con una llave vacía.



### Trabajar con tablas internas: Operaciones con índices.

Un índice se utiliza para mejorar el rendimiento de acceso en la secuencia lógica o para mejorar la validación de filas si la tabla tiene huecos debido a operaciones de borrado. Las operaciones que afectan esta relación de secuencia lógica se describe en la figura 7.4.3.3-2. En ella podemos observar su declaración de acuerdo al tipo de procesamiento que queremos manejar (un solo registro o masivamente) y el tipo de operación que deseamos realizar con la tabla interna como es leerla, modificarla, insertar nuevos registros, agregar nuevos registros al final o borrar registros. Estas operaciones están estrechamente ligadas con el manejo interno en memoria de árboles binarios para los índices.


	Procesamiento de un registro	Procesamiento Masivo
Leer	READ TABLE <itab> INDEX <n> [INTO <wa>   ASSIGNING <FS>]	LOOP AT <itab> [FROM <n1>]   [TO <n2>] [INTO <wa>   ASSIGNING <FS>] ENDLOOP
Agregar	APPEND [<wa> TO] <itab>	APPEND LINES OF <itab1> [FROM <n1>] [TO n2] TO <itab2>
Insertar	INSERT [<wa> INTO] <itab> [INDEX <n>]	INSERT LINES OF <itab1> [FROM <n1>] [TO <n2>] INTO <itab2> [INDEX <n3>]
Modificar	MODIFY <itab> [INDEX <n>] [FROM <wa>]	
Borrar	DELETE <itab> [INDEX <n>]	DELETE <itab> FROM <n1> TO <n2>

Fig. 7.4.3.3-2 Operaciones con índices

### Trabajar con tablas internas: Operaciones con llaves.

Las tablas hashed están basadas en una llave única. El valor de esta llave se asigna a un valor único que se obtiene con una función hash. Para cada uno de estos números existe una dirección de memoria que se guarda en una matriz hash [GH2009].

Si se borra o reorganiza la tabla hashed, se debe crear una doble lista ligada (apuntador siguiente y anterior al registro), para que el acceso secuencial con un loop sea posible de acuerdo a la secuencia de insertado. Estas listas ligadas exigen aproximadamente 50% más espacio para la administración de memoria.

En la figura 7.4.3.3-3 están esquematizados las operaciones definidas en ABAP con el uso de llaves.


	Procesamiento de un registro		Procesamiento masivo
	Implicito del <wa>	Explicito	
Leer	READ TABLE <itab> FROM <wa> [INTO <wa>   ASSIGNING <FS>].	READ TABLE <itab> [INTO <wa>   ASSIGNING <FS>] WITH [TABLE] KEY <k> = <v> ... .	LOOP AT <itab> [INTO <wa>   ASSIGNING <FS>] WHERE <condition>.
Insertar	INSERT [<wa> INTO] TABLE <itab>.		INSERT LINES OF <itab1> INTO TABLE <itab2>.
Insertar colectiv.	COLLECT [<wa> INTO] <itab>.		
Modificar	MODIFY TABLE <itab> [FROM <wa>].		MODIFY <itab> [FROM <wa>] WHERE <condition>.
Borrar	DELETE TABLE <itab> [FROM <wa>].	DELETE TABLE <itab> WITH TABLE KEY <k> = <v>.	DELETE <itab> WHERE <condition>.

Fig. 7.4.3.3-3 Operaciones con llave

### Aspectos de Rendimiento

A continuación analizaremos algunas sentencias que causan problemas de rendimiento y que están directamente ligadas con su comportamiento lineal [GH2009]:

READ TABLE itab WITH [TABLE] KEY ...

Es usualmente un problema con:

- En tablas Estándar si no se especifica la sentencia BINARY SEARCH.
- En tablas Ordenadas si la parte principal o inicial de la llave no se especifica.
- En tablas Hashed si la llave no se especifica completamente.

MODIFY / DELETE TABLE itab WHERE ...

LOOP AT itab WHERE ...

Es usualmente un problema con:

- En tablas Estándar
- En tablas Ordenadas si la parte principal o inicial de la llave no se especifica.
- En tablas Ordenadas si la condición WHERE encuentra mucho duplicados.
- En tablas Hashed si la llave no se especifica completamente.

LOOP AT itab WHERE ...

Es usualmente un problema con:

- En tablas Estándar
- En tablas Ordenadas
- En tablas Hashed

COLLECT wa INTO itab.

Es usualmente un problema con:

- En tablas Estándar

Todas estas sentencias causan problemas de rendimiento si se ejecutan con frecuencia o con grandes volúmenes, ya que hacen un escaneo completo de la tabla interna.

### 7.4.3.4. Notación asintótica o grande O para problemas de ABAP

La notación asintótica, es una notación matemática que nos ayuda a representar el comportamiento de límite de una función, cuando el argumento de la función tiende a un valor en específico o al

infinito, en general se usan funciones simples. Esta notación caracteriza las funciones de acuerdo a su grado de crecimiento, por ejemplo dos funciones con el mismo grado de crecimiento se pueden representar usando la misma notación asintótica.

Aunque fue desarrollada como parte de las matemáticas puras, esta notación se usa en la actualidad con frecuencia en el análisis de algoritmos para representar el consumo que realiza el algoritmos de recursos computacionales: el peor caso o el tiempo promedio de ejecución o el consumo de memoria de un algoritmo es comúnmente expresado como una función de la longitud de la entrada en la notación asintótica. De esta forma cuando se resuelve un problema con un algoritmo, podemos predecir el comportamiento de éste independientemente de la arquitectura o de la rapidez de la computadora [BS2007].

Debido a que la notación descarta las constantes multiplicativas en el tiempo de ejecución e ignora la eficiencia para cuando el argumento es poco numeroso, no siempre muestra el algoritmo más rápido para conjuntos de información reducidos, pero aun así es muy efectivo para comparar las escalabilidad de varios algoritmos cuando su entrada se procesa grandes cantidades de información. Se utilizan generalmente las siguientes reglas de simplificación:

- Si  $f(x)$  es una suma de varios términos, el términos con un mayor grado de crecimiento se conserva, y el resto se omiten.
- Si  $f(x)$  es un producto de varias funciones, cualquier constante (términos en el producto que no dependan de  $x$  se omiten.

Por ejemplo en una función  $f(x) = 7x^6 + 11x^4 + 12$ , considerando las dos simplificaciones antes mencionados, eliminaríamos los términos con menor crecimiento  $7x^6 + 11x^4 + 12$  y la constante que hace producto  $7x^6$ , quedando en la notación solo como  $O(x^6)$

Con lo anterior, la dependencia del tiempo de ejecución  $t$  con la cantidad de información  $n$ , utilizando la notación asintótica, tenemos las siguientes operaciones:

- $t = O(1)$ : Operaciones que no dependen en el cantidad de información.
- $t = O(n)$ : Operaciones con dependencia directa de la cantidad de información que procesan. Por ejemplo, para encontrar un registro en una tabla con  $n$  registros, necesitas leer, en el peor caso,  $n$  registros.
- $t = O(n^2)$ : Operaciones con una dependencia cuadrática de la cantidad de información que procesan. Por ejemplo, aparecen en loops anidados con tablas internas donde se procesa información de aplicación. Existen dependencias mayores  $O(n^2 * p^2 \dots)$ , pero con la cuadrática se ejemplifica bien y no es necesario incluirlas.
- $t = O(\log n)$ : Operaciones con una dependencia logarítmica y aparecen en búsquedas binarias
- $t = O(n \log n)$ : Operaciones con loops anidados, donde el loop inferior hace búsquedas binarias o en tablas ordenas.

La tabla 7.4.3.5-1 que se muestra a continuación contiene las operaciones más relevantes y comunes de ABAP para la programación de aplicaciones de negocios. La tabla incluye la notación antes descrita y el código que la ejemplifica.

Notación	Nombre	Ejemplo de código en ABAP
$O(1)$	Constante	APPEND wa INTO i tab.
$O(\log_2 n)$	Logarítmica	READ TABLE i t_tab INTO l s_tab WITH KEY i = ' 5' USING BI NARY SEARCH.
$O(n^c), c > 1$	Polinomial o algebraica	LOOP (anidados * c)  LOOP AT i t_tab. LOOP AT i t_tab2. LOOP AT i t_tab3.  ENDLOOP.

		ENDLOOP. ENDLOOP.
<b>O(n)</b>	Lineal	LOOP AT it_tab. ... ENDLOOP.
		READ TABLE it_tab INTO ls_tab WITH KEY i = '5'.
		SELECT (full table scan)
<b>O(n log<sub>2</sub> n)</b>	Lineal	SORT it_tab1.
		LOOP including single READ BSEARCH
<b>O(n<sup>2</sup>)</b>	Cuadrática	LOOP AT it_tab. LOOP AT it_tab2. ... ENDLOOP. ENDLOOP.
		LOOP AT it_tab. READ TABLE it_tab2 INTO ls_tab WITH KEY i = it_tab- i. ENDLOOP.
<b>O(C<sup>n</sup>)</b>	Exponencial geométrica	o Usar la búsqueda de fuerza bruta para encontrar una combinación. No es común encontrarla en problemas de negocios.

Tabla 7.4.3.5-1 Notación Asintótica para ABAP

#### 7.4.3.5. Búsqueda del consumo lineal

Durante un proyecto de implementación el tiempo de resolución de problemas es muy limitado lo que ocasiona que los programadores con menos experiencia no estimen bien la degradación del rendimiento de un sistema con el paso del tiempo y el aumento de datos. Un ejemplo que encontramos comúnmente es que un cliente reporta un problema con un programa que se había estado desempeñando bien hasta cierto momento en el tiempo, cuando el programa tiene que procesar facturas o cualquier otro documento con más de 30,000 posiciones. Un escenario común es ver como el cliente espera procesar su información en aproximadamente 2.5-3 horas por experiencia con documentos previos con menos posiciones, pero se sorprende cuando el programa se tarda en procesar la información más de 20 horas.

Si dentro del proyecto se realizaron pruebas de volumen, estas confirmaron que el programa se comportaba de manera adecuada con hasta 15,000 posiciones por lo que se asumió que el comportamiento del programa se podía extrapolar de manera lineal, lo cual por los resultados reales del programa no es del todo cierto. El cliente ante esta situación se muestra decepcionado e inconforme, porque él asumió que con las pruebas de volumen comprobaba que no tendría problemas de rendimiento en su operación.

Este problema de escalabilidad del software para manejar mayores cantidades de información está estrechamente relacionado con la forma en cómo se puede predecir el comportamiento del código de ABAP.

Para entender este problema de la extrapolación necesitamos entender el tiempo de ejecución de una mejor manera, se necesita entender la dependencia de un programa con la cantidad de información que va a procesar. Es por eso que es necesario conocer las operaciones de ABAP que son dependientes de la cantidad de información que procesan.

Para profundizar en el tema daremos una breve introducción de la notación asintótica a continuación.

El problema que explicamos en la primera parte se basa en el hecho de que, aún con pruebas de volumen, siempre es difícil en la vida real identificar cual es el comportamiento real de nuestro programa en forma exacta.

Hemos graficado algunas de las funciones de la Tabla Notación Asintótica para ABAP, para visualizar de mejor manera porque estas funciones causan problemas de tiempo de ejecución con grandes cantidades de información.

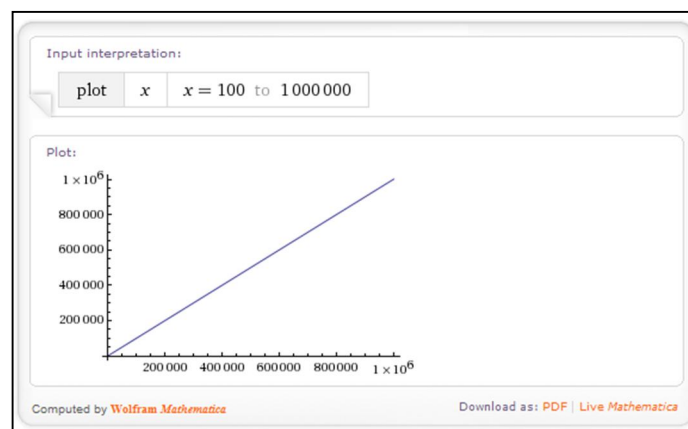


Fig. 7.4.3.6-1 Loop y Read s/BB

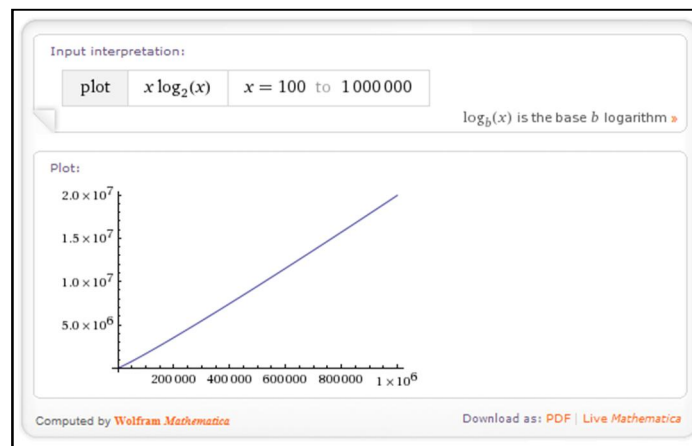
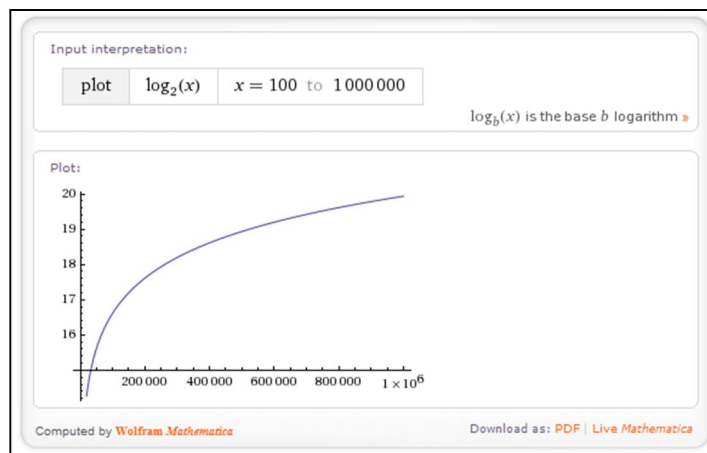


Fig. 7.4.3.6-2 Operación SORT it\_tab



**Fig. 7.4.3.6-3 Operación READ con BB**

En la práctica cuando nos enfrentamos a este tipo de problemas, donde el tiempo de ejecución es muy alto, se empieza por monitorear el proceso de trabajo del sistema en la transacción sm50, de ahí podemos determinar si la mayoría del tiempo se consume en la base de datos o en algún programa de ABAP. Podemos hacer un trace también desde la st12 a ese proceso de trabajo y de ahí obtener los porcentajes de consumo o también parar varias veces el programa con la opción de debug y si el 90% del tiempo está en una subrutina de ABAP, la probabilidad de detenernos en esa subrutina es también de 90%. Identificando una vez el código la corrección puede empezar.

Sin embargo el escenario ideal sería evitar estos problemas durante la fase de diseño donde el sistema de desarrollo o pruebas tiene poca información para poder correr pruebas de volumen.

Como conclusión podemos decir que es de cierta manera fácil identificar problemas no lineales en sistemas productivos, pero no es fácil hacerlo en sistemas de pruebas o desarrollo con poco volumen de información, donde las funciones lineales pasen por alto. Debemos por lo tanto ser capaces de poder modelarlo para esta fase el comportamiento o utilizar paquetes más pequeños de información.

Nosotros en este trabajo explicaremos como modelar el código en base a [BS2007] y poder de alguna manera mitigar este problema en la fase de diseño con algunas mejoras básicas.

Ahora que tenemos un mejor entendimiento del tiempo de ejecución de cada uno de los pasos podemos describir el tiempo de ejecución de un programa como la suma todos los contribuidores con estas dependencias de la cantidad de información:

$$t(n) = A O(1) + B O(n) + C O(n^*n) + D O(n \log n)$$

Si utilizamos la información de las trazas podemos ver la forma en como escala el tiempo neto por cada una de la llamadas de ABAP por ejecución. Las opciones que podemos identificar son:

- El tiempo neto por ejecución incrementa cuadráticamente.
- El número de ejecuciones incrementa cuadráticamente.
- Las ejecuciones y el tiempo incrementan linealmente, lo cual indica que seguramente se debe hacer un análisis de como optimizar el código para escalar aún mejor que linealmente.

Una vez que se identifica estos algoritmos, se debe crear una traza incluyendo las tablas internas o si la codificación incluye módulos demasiado pequeños solo es necesario buscar aquellas ejecuciones de funciones con comportamientos no lineales.

El código no lineal es comúnmente asociado con loop anidado en tablas internas y los programadores no están conscientes que algunas sentencias para tablas internas escalan linealmente con el número de registros por lo cual se debe tener atención en su uso. Estas sentencias se nombran en la siguiente sección con el método alternativo de optimización.

### 7.4.3.6. Algoritmos de búsqueda que ayudan a la mejora de problemas comunes de búsquedas en tablas internas.

A continuación mostramos algunos de los algoritmos de búsqueda que se han desarrollado con algunos ejemplos de optimización. Estos ejemplos lo que buscan es mostrar la utilización de la notación asintótica para la clasificación de los algoritmos comúnmente usados en la programación de ABAP.

El primer ejemplo es la búsqueda de un registro utilizando la tabla que contiene información de cabecera como puede ser la cabecera de un documento de ventas y manipular la información a nivel de posición del documento de venta. Si queremos el detalle de varias facturas hay que buscar esas facturas para un cliente y sus respectivas posiciones, el algoritmo comúnmente usado para resolver este problema es:

Tiempo de ejecución  $O(n^2)$

```
LOOP AT it_cabecera INTO is_cabecera.
  READ TABLE it_detalle WITH KEY i = is_cabecera-i INTO is_detalle.
ENDLOOP.
```

La optimización propuesta se basa en el ordenamiento de la información detallada para poder encontrar los registros utilizando una búsqueda binaria en vez de una búsqueda lineal.

Tiempo de ejecución con ejecución  $O(n \log_2 n)$

```
SORT it_detalle BY llave.
LOOP AT it_cabecera INTO is_cabecera.
  READ TABLE it_detalle WITH KEY llave = is_cabecera-llave
  INTO is_detalle BINARY SEARCH.
ENDLOOP.
```

El segundo ejemplo es la búsqueda de información dentro de una tabla interna que contiene el detalle de la información como lo vimos en el primer ejemplo, pero con la necesidad de agregar registros a esa tabla interna. Esto impide la implementación del mismo algoritmo debido a que se rompe el ordenamiento inicial al agregar un registro al final de la tabla.

Tiempo de ejecución  $O(n^2)$

```
LOOP AT it_cabecera INTO is_cabecera.
  READ TABLE it_detalle WITH KEY llave = is_cabecera-llave
  INTO is_detalle.
  APPEND is_so3 TO it_detalle.
ENDLOOP.
```

Un error común que podemos encontrar también en los códigos es el utilizar un BINARY SEARCH pero para asegurar la lógica se ordena la tabla cada vez que se agrega un registro, lo cual puede llegar a crear un problema de rendimiento aún peor.

Tiempo de ejecución  $O(n^2 \log_2 n)$

```
SORT it_detalle.
LOOP AT it_cabecera INTO is_cabecera.
  READ TABLE it_detalle WITH KEY llave = is_cabecera-llave
  INTO is_detalle BINARY SEARCH.
  APPEND is_so3 TO it_detalle.
  SORT it_detalle.
ENDLOOP.
```

La mejora recomendada es similar al primer caso, sólo que se tiene que adicionar lógica al programa para que las modificaciones en la tabla interna no alteren el orden de la misma. Para lograrlo se hace una búsqueda adicional para conocer el índice adecuado donde se debe insertar el registro dentro de la tabla interna y así evitar el reordenamiento de la misma.

Vale la pena mencionar que se utiliza la variable de sistema sy-tabix que se encuentra en la estructura del sistema SYST (Para verificar su estructura ver en transacción SE11 -> Data type -> SYST) y que define el índice en tiempo de ejecución de una tabla interna.

Tiempo de ejecución con optimización  $O(n \log_2 n)$

```
SORT it_detalle BY i.
LOOP AT it_cabecera INTO is_cabecera.
```

```

READ TABLE it_detalle WITH KEY llave= is_cabecera-llave
INTO is_detalle BINARY SEARCH.

READ TABLE it_detalle WITH KEY llave = is_detalleNuevo-llave BINARY SEARCH
TRANSPORTING NO FIELDS.
INSERT is_detalleNuevo-llave INTO it_detalle INDEX sy-tabix.
ENDLOOP.

```

El tercer ejemplo es un problema es muy similar al primer ejemplo donde buscamos un registro, pero en esta ocasión vamos llenando la estructura del segundo loop con los registros que cumplan la condición uno por uno. Para encontrar un registro en una tabla desordenada hay que recordar que el peor caso es la lectura de toda la tabla.

#### Tiempo de ejecución $O(n^2)$

```

LOOP AT it_vbak INTO wa_vbak.
  LOOP AT it_vbap INTO wa_vbap WHERE vbeln = wa_vbak-vbeln.
  ...
ENDLOOP.
ENDLOOP.

```

Para la optimización utilizamos el ordenamiento de la tabla y buscamos binariamente el índice inicial desde donde debemos empezar el segundo loop. Finalmente salimos del segundo loop una vez que la información que tenemos no concuerda con la primera búsqueda.

#### Tiempo de ejecución con optimización $O(n \log_2 n)$

```

SORT it_vbap BY vbeln.
LOOP AT it_vbak INTO wa_vbak.
  READ TABLE it_vbap WITH KEY vbeln = wa_vbak-vbeln
  BINARY SEARCH TRANSPORTING NO FIELDS.
  LOOP AT it_vbap INTO wa_vbap FROM sy-tabix.
    IF wa_vbap-vbeln = wa_vbak-vbeln.
    ...
    ELSE.
      EXIT.
    ENDIF.
  ENDLOOP.
ENDLOOP.

```

El cuarto ejemplo es un problema muy similar al segundo ejemplo donde buscamos un registro dentro de la tabla interna de detalles, pero en esta ocasión vamos a eliminar registros de esta tabla interna y para encontrar los registros que queremos borrar necesitamos buscar por cada iteración esos registros.

#### Tiempo de ejecución $O(n^2)$

```

SORT it_vbap BY matrnr.

LOOP AT it_mara INTO wa_mara.
  READ TABLE it_vbap INTO wa_vbap WITH KEY matrnr = wa_mara-matrnr BINARY SEARCH.
  DELETE it_vbap WHERE matrnr = wa_mara-matrnr.
  DELETE it_vbak WHERE vbeln = wa_vbap-vbeln.
ENDLOOP.

```

Para la optimización utilizamos el ordenamiento de la tabla y buscamos binariamente el índice inicial desde donde debemos empezar el segundo loop que se termina cuando se recorrieron los valores buscados. La cláusula DELETE WHERE se reemplaza por una búsqueda binaria para encontrar el índice seguido de un LOOP AT FROM INDEX. Dentro del LOOP existe una sentencia para borrar en base al índice DELETE INDEX, finalmente se borran todos los registros que cumplen la condición y de lo contrario termina el LOOP.

#### Tiempo de ejecución con optimización $O(n \log_2 n)$

```

LOOP AT it_mara INTO wa_mara.
  READ TABLE it_vbap INTO wa_vbap1 WITH KEY matrnr = wa_mara-matrnr BINARY SEARCH.
  IF sy-subrc = 0.
    LOOP AT it_vbap INTO wa_vbap FROM sy-tabix.
      IF wa_vbap-matrnr = wa_mara-matrnr.
        DELETE it_vbap INDEX sy-tabix.
      ELSE.

```



```

EXIT.
ENDIF.
ENDLOOP.
ENDIF.
READ TABLE it_vbak INTO wa_vbak WITH KEY vbeln = wa_vbap1-vbeln BINARY SEARCH.
IF sy-subrc = 0.
    LOOP AT it_vbak INTO wa_vbak FROM sy-tabix.
        IF wa_vbak-vbeln = wa_vbap1-vbeln.
            DELETE it_vbak INDEX sy-tabix.
        ELSE.
            EXIT.
        ENDIF.
    ENDLOOP.
ENDIF.

ENDLOOP.
    
```

Para ayudar a entender mejor la optimización lograda vamos a mostrar el tiempo de ejecución de este último ejemplo con los siguientes valores:

Sin Optimización	Con Optimización
it_vbap had 16.253 líneas al principio	it_vbap had 16.253 líneas al principio
it_vbak had 8.447 líneas al principio	it_vbak had 8.447 líneas al principio
it_vbap had 255 líneas al final	it_vbap had 255 líneas al final
it_vbak had 8.047 líneas al final	it_vbak had 8.047 líneas al final
Duración: 40.097.786 microsegundos	Duración: 4.476.945 microsegundos

Se puede ver una mejora sustantiva de aproximadamente 90% en el tiempo de ejecución y la mayoría del tiempo se utiliza obteniendo la información de la Base de Datos y no manipulándola.

Call	No.	Gross	Net	Gross (%)	Net (%)	Program (called program)	Type
Delete IT_VBAP	15.001	22.108.07...	22.108.07...	55,1	55,1	ZEX_DELETE_WHERE_NBS	
Delete IT_VBAP	15.001	13.322.22...	13.322.22...	33,2	33,2	ZEX_DELETE_WHERE_NBS	
Select VBAP	1	2.032.169	2.032.169	5,1	5,1	ZEX_DELETE_WHERE_NBS	DB->
Select MARA	1	1.611.444	1.611.444	4,0	4,0	ZEX_DELETE_WHERE_NBS	DB->
Select VBAK	1	569.436	569.436	1,4	1,4	ZEX_DELETE_WHERE_NBS	DB->
Loop at IT_MARA	1	35.858.86...	294.590	89,4	0,7	ZEX_DELETE_WHERE_NBS	
Read Table IT_VBAP	15.001	133.981	133.981	0,3	0,3	ZEX_DELETE_WHERE_NBS	
Runtime analysis	1	40.097.78...	0	100,0	0,0		Sys.
Submit Report ZEX_DELETE_WHERE_NBS	1	40.097.73...	7.096	100,0	0,0	Testframe	
Program ZEX_DELETE_WHERE_NBS	1	40.089.37...	4.307	100,0	0,0		

Fig. 7.4.3.7-1 Trazas con los tiempos de repuesta sin búsqueda binaria

Call	No.	Gross	Net	Gross (%)	Net (%)	Program (called program)	Type
Select VBAP	1	1.997.98...	1.997.98...	45,1	45,1	ZEX_DELETE_WHERE_BS	DB->
Select MARA	1	1.709.36...	1.709.36...	38,6	38,6	ZEX_DELETE_WHERE_BS	DB->
Select VBAK	1	551.260	551.260	12,5	12,5	ZEX_DELETE_WHERE_BS	DB->
Loop at IT_MARA	1	144.207	42.751	3,3	1,0	ZEX_DELETE_WHERE_BS	
Loop at IT_VBAP	522	54.755	41.459	1,2	0,9	ZEX_DELETE_WHERE_BS	
Read Table IT_VBAP	15.001	25.065	25.065	0,6	0,6	ZEX_DELETE_WHERE_BS	
Read Table IT_VBAP	15.001	20.106	20.106	0,5	0,5	ZEX_DELETE_WHERE_BS	
Delete IT_VBAP	15.998	13.296	13.296	0,3	0,3	ZEX_DELETE_WHERE_BS	
Sort IT_VBAP	1	8.023	8.023	0,2	0,2	ZEX_DELETE_WHERE_BS	
Program ZEX_DELETE_WHERE_BS	1	4.423.80...	3.806	99,9	0,1		
Sort IT_VBAK	1	2.769	2.769	0,1	0,1	ZEX_DELETE_WHERE_BS	

Fig. 7.4.3.7-2 Trazas con los tiempos de repuesta con búsqueda binaria

### 7.4.3.7. Líneas directivas para la programación en ABAP

Independientemente de la experiencia que se tenga con el lenguaje de programación ABAP, siempre es común encontrar código que sólo cumple con su objetivo inicial y el programador rara vez es lo suficientemente cuidadoso o metódico para buscar alternativas de mejora o nuevas funciones dentro de ABAP que le ayuden a su tarea.

Estas simples líneas directivas pueden ayudar a los programadores a remontar ese camino de experimentación y mejora en sus nuevos o viejos programas manteniendo el soporte adecuado en el desarrollo de programas:

- Mantenibles
- Bien estructurados
- Con buen rendimiento
- Legibles
- Actualizables
- Robustos

La mayoría de estas guías de programación vienen directamente del grupo encargado del lenguaje ABAP que se encarga principalmente del desarrollo del lenguaje, el compilador, las tablas internas, etc. Es por eso que la guía se basa más en el marco de trabajo de ABAP como lenguaje más que una guía de programación para aplicaciones de negocio, que queda por arriba para los arquitectos de soluciones.

En esta parte de la guía de recomendaciones de ABAP que excluye guía sobre el rendimiento de código ABAP, pero que desde el mero punto de vista del lenguaje es necesario.

Adicionalmente cada grupo de desarrollo tiene que crear sus propios estándares de producto y desarrollo de software para cumplir con los requisitos de calidad, legalidad y seguridad de los productos que se desarrollen [KHT2009].

#### Reglas fundamentales basadas en la experiencia

- El modelo de programación orientada a objetos que provee ABAP Objects es superior al modelo basado en procedimientos ya que ofrece mejores características de encapsulamiento de información, creación de instancias explícitas de objetos, aumenta la reutilización de código a través de la herencia, interfaces independientes y manejo explícito de los eventos. Adicionalmente ABAP Objects provee mejor acceso a nueva tecnologías. Por lo tanto se recomienda utilizar ABAP Objects siempre que sea posible, construyendo todo el código en clases globales y locales. La excepción es el uso de servicios que se implementaron utilizando módulos de función
- Basándonos en que principalmente se recomienda utilizar ABAP Objects, la ejecución o invocación de un programa consiste esencialmente en llamar a un método. Las transacciones están conectadas a un método principal de una clase global o local.
- La regla básica para la programación de interfaces de usuario (UI) es separar la capa de presentación de la de aplicación. No mezclar el código que maneja el flujo de pantallas o la apariencia de la pantalla con código de negocio o de aplicación independiente de la pantalla. Los motivos principales son:
  - Acceder a la lógica de negocio utilizando diferentes UIs (“User Interfaces”)
  - Acceder a la lógica de negocio utilizando diferentes clientes automatizados (usuarios no humanos)
  - Esta llamada separación de preocupaciones es un modelo básico que soporta la estabilidad y mantenimiento del software porque permite la reutilización de servicios y la pruebas de la lógica de negocio usando casos aislados de pruebas unitarias.
  - Se recomienda la utilización de la tecnología Web Dynpro que está basada en el modelo MVC (Model-View-Controller) y con eso se logra la separación de la lógica de negocio de la de presentación.

#### Mejores Prácticas aprendidas en los proyectos de implementación:

- Trabaja solo con programas Unicode, aun cuando el sistema no vaya a ser ejecutado en un sistema Unicode. Las reglas de Unicode automáticamente aumentan la calidad y mantenimiento de los programas.
- Implementa las sentencias operacionales de tus programas solo en métodos. Muchos elementos obsoletos y potencialmente peligrosos son prohibidos sintácticamente en las clases y por tanto se evita la posibilidad de acarrear eso a tus programas de forma automática.
- Organiza los programas pulcramente, especialmente a la secuencia de las partes del programa, la secuencia de declaraciones y la secuencia de las sentencias en procedimientos.
- Ordena el código fuente de programas grandes con programas INCLUDE, pero no con macros. No reutilices los programas INCLUDE en muchos programas.
- Mantener el código moderno, se recomienda esta alerta de las nuevas noticias de ABAP y aplicar las nuevas construcciones del código tan pronto como están disponibles. De hecho puede valer la pena re-escribir algunas partes con estas nuevas funcionalidades.
- Mantener el código de cualquier advertencia sintáctica y reemplaza el código que muestre errores, advertencias, y mensajes que regrese la herramienta de Revisión extendida de programa.
- Mantener los métodos simples y utilizables
- Evita estructuras de control que son demasiado complejas y que pueden causar comportamientos inesperados.
- Pensar claramente sobre el mejor lugar para declarar tipos de datos y objetos, y como usarlos apropiadamente.
- Aproveche los conceptos de programación dinámica de acuerdo a sus necesidades, no trate de resolver problemas simples con complicaciones.
- Distinguir con claridad cuándo se debe utilizar el significado del manejo de errores - los valores de retorno, las excepciones, las afirmaciones o mensajes.
- Pon a prueba todas las partes de sus programas con las herramientas adecuadas
- Documente sus programas
- Aproveche el concepto completo de paquetes

Somos conscientes de que este conjunto de recomendaciones representa nuestra visión personal de cómo deben hacerse las cosas. También somos conscientes de que nuestra colección puede ser demasiado heterogénea y es casi seguramente incompleta. Por lo tanto, si usted tiene más ideas no dude en incluirlas en su guía personal de recomendaciones de ABAP.

Estamos convencidos de que sus programas se convertirá en código robusto, fácil de mantener comprensible, y mejor, si usted sigue estas directrices.

## 7.5. Dimensionamiento del Hardware

En el mundo empresarial actual existe una gran variedad de escenarios de negocio que crean diferentes demandas de rendimiento y procesamiento en productos de software estandarizados. Algunos ejemplos pueden ser el correr una nómina de más de 3 millones de empleados y pensionados en menos de 3 horas, almacenar documento por más de 10 años lo cual crea tablas y bases de datos enormes o tal vez crea rutas de distribución con mercancía para el reparto de bebidas en un periodo no mayor de 4 horas.

Todos los anteriores son ejemplos de procesos de negocio que demandan tener un tiempo de respuesta predecible para cuando se trabaja con el sistema de manera interactiva y para los procesos de fondo como el pago de nómina lo más importante es la cantidad de pasos por unidad de tiempo.

Sin embargo no solo las grandes empresas esperan tener un sistema que se comporte de manera predecible cuando aumenta la carga de trabajo. Todas las empresas esperan un software que sea

escalable, lo cual significa que el sistema siempre tiene que ejecutar las tareas de la manera más óptima y no solo cuando se tiene grandes cargas de trabajo.

Un requisito para cumplir con esto es que tanto el hardware como el software son elemento escalables. Algunas de las dimensiones que se pueden mencionar de la escalabilidad son:

- A nivel de usuario final, la cantidad de usuarios puede crecer y aumentar
  - El número de objetos de negocio o la cantidad de elementos dentro del objeto de negocio es también escalable.
- La cantidad de trabajos de fondo que se pueden procesar es escalable
- La cantidad de servidores de aplicación que se puede agregar al sistema
- La base de datos permite el aumento de memoria, CPUs, cores o threads.

Entonces cuando un sistema de TI es escalable, permite hacer dimensionamientos adecuados. Estos dimensionamientos de hardware permiten estimar la cantidad necesaria de recursos para poder ejecutar nuestros procesos de negocio de forma efectiva. La escalabilidad y la determinación del dimensionamiento son pilares esenciales para utilizar los benchmarks de SAP, los cuales verifican esta escalabilidad de los sistemas SAP en diferentes plataformas de hardware y de base de datos.

### **Benchmarks Standard de SAP**

Existen diferentes métodos para analizar el rendimiento de un sistema, sin embargo estos básicamente se pueden sub-dividir en métodos de mediciones y técnicas de modelado como el modelo de servicios de colas. Sin embargo para medir, comparar y normalizar los tiempos de servicio para diferentes plataformas se necesita una táctica con un estándar independiente de la plataforma de hardware [SAPBM2012].

Las siguientes dos tácticas se utilizan para definir las plataformas de hardware con un estándar de rendimiento independiente y que ayudan a generar un benchmark [SAPBM2012]:

1. Una Máquina real de referencia. Todos los resultados se comparan con los resultados obtenidos de la máquina real de referencia tal como se maneja la definición de MIPS (Million instructions per second) por sus siglas en inglés. Este modelo es simple el rápido avance de la tecnología en hardware y los problemas de portabilidad de problemas son desventajas comunes en este modelo.

2. Una Máquina teórica de referencia. Una máquina abstracta de referencia es simplemente definida por sus características de rendimiento. Al ser esta una máquina abstracta su rendimiento teórico siempre puede ser adaptado a la tecnología de punta. Haas y Zorn sugieren esta máquina abstracta con simulaciones de procesos reales para la determinación de los factores de carga. Es con esto que el rendimiento se define con el procesamiento por unidad de tiempo y el tiempo de respuesta de un número específico de programas.

En SAP se optó por el segundo enfoque como un mejor análisis unificado para las aplicaciones de SAP. Los programas teóricamente referenciados son entonces llamados los Benchmarks estándares de aplicación de SAP.

Estos estándares simulan entonces cargas de fondo y de usuarios finales para diferentes aplicaciones, creando el procesamiento por unidad de tiempo, definiendo así el poder de procesamiento de un sistema o configuración.

Finalmente con fines de comparación, SAP creó una unidad independiente de hardware llamada SAPS (SAP Application Performance Standard) que expresa el poder de una cierta máquina que se puede comparar analógicamente con los caballos de fuerza de un auto.

### **Herramientas para el cálculo del dimensionamiento.**

Existen diferentes formas de ejecutar el dimensionamiento y básicamente las podemos agregar en cinco:

La primera es un cálculo inicial basado principalmente en la experiencia de implementaciones similares. Este método es generalmente poco asertivo para el largo plazo y no se recomienda para proyectos complejos.

El segundo es un método llamado dimensionamiento T-shirt, el cual utiliza algoritmos muy simples con muchas suposiciones. Este método es común para productos más simples como MDM o con proyectos pequeño donde los procesos de negocio son simplificados.

El tercero son fórmulas que se utilizan para extrapolar siempre y cuando se cuente con información previa como promedios de memoria, disco o red. Un ejemplo es para calcular el ancho de banda necesario, la cantidad de disco y factores del dimensionamiento con menores fluctuaciones.

El cuarto son los cuestionarios estructurados con preguntas que ayudan a identificar mejor las necesidades del negocio en términos de creación de documentos y cantidad de usuarios. También ayudan a identificar tiempos críticos que tiene que ser soportados por el dimensionamiento que se realice.

Finalmente en quinto es la herramienta en línea de SAP llamada quicksizer. Esta herramienta se basa en cuestionarios que toman como referencia los usuarios o la cantidad de documento que se tiene que procesas por unidad de tiempo como entradas para el cálculo del dimensionamiento. En versiones más recientes también se toma en cuenta principalmente para las aplicaciones con uso intensivo de web la velocidad de optimizada de los hilos de los procesadores.

### **La herramienta en línea de SAP: Quick Sizer [SAPSZ2012]**

SAP mantiene en conjunto con sus socios de Hardware una herramienta disponible desde internet. Esta herramienta provee a los clientes con estimados basados en la información que se llena, categoría generales para un dimensionamiento inicial, es decir, para una implementación nueva. Así los clientes cuenta con una base de dimensionamiento independiente de los proveedores de hardware.

El Quick Sizer cuenta con dos modelos básicos de dimensionamiento que son independientes uno del otro: dimensionamiento por usuario y dimensionamiento por rendimiento. El tener los dos métodos permite al cliente llenar el cuestionario de manera más flexible, sin embargo cada uno tiene sus inconvenientes, por lo que siempre es recomendable hacer el ejercicio con ambos métodos y comparar los resultados. Esto es importante para verificar la confiabilidad de los parámetros de entrada del cuestionario como los usuarios muy activos o la cantidad real de documentos que se crean actualmente por unidad de tiempo.

A partir de este resultado es de donde los proveedores de hardware pueden empezar a hacer sus propuestas. Es muy importante que durante el avance del proyecto se revise y modifique el cuestionario con la información más reciente del proyecto para tener el dimensionamiento adecuado para la salida en productivo.

El Quick Sizer no debe ser utilizado para proyectos de re-dimensionamiento ni para casos de actualizaciones, ya que la información actual del sistema da información más confiable.

### **Dimensionamiento Experto**

En el ciclo de vida de las aplicaciones de software después del go-live, en una actualización, en una migración, en el aumento de unidades de negocio o en cambios funcionales fuertes un re-dimensionamiento es necesario y en estos casos se recomienda hacer un dimensionamiento con los monitores de SAP y con nota o documentos de SAP que contengan los deltas.

El procedimiento que se detalla lo realizamos como parte de nuestra especialidades técnicas y es como sigue:

1. Monitorear la utilización de CPU, el crecimiento de tablas, la frecuencia de creación de documentos y el consumo de memoria.
2. Existen diferentes procedimientos de acuerdo a los objetivos.
  - Re-dimensionamiento: Agregar la carga proveniente de más usuarios o proyectos con la misma estructura.
  - Dimensionamiento Delta: Tratar el delta como un nuevo dimensionamiento y sumar la carga calculada.
  - Dimensionamiento de Actualización: Determinar los requerimientos adicionales y sumar la carga calculada.

Al final se tiene que juzgar si se cuenta con suficiente hardware o si se necesita más capacidad.

Monitores Disponibles:

Análisis de disco - DB02, Monitor de propio de la BD

- „(Rendimiento de la BD en tablas e índices)

Análisis de CPU- ST06, ST03N, STAD, ST03G

- „(Análisis de carga de trabajo, Registros estadísticos)

Análisis de Usuarios ST07, STAD, ST03G

- (Monitoreo de la aplicación, Registros estadísticos)

Análisis de Memoria - SM04, STAD, GCLOG

- (Lista de Usuarios, Registros estadísticos)

Carga de red en la capa frontal STAD, ST03N, ST03G, httplog

- (Registros estadísticos, Análisis de carga de trabajo)

Como regla siempre el 20% de los procesos causan el 80% de la carga, es por eso que hay que analizar:

- Crecimiento de las 20 tablas más grandes
- Promedio y picos de CPU
- Promedio y picos de Memoria

### A continuación mostramos un ejemplo de un dimensionamiento experto realizado

Fórmula utilizada:

$$CPU_{FUT} = CPU_{TOD} * (1 - B_{LOD}) * USERS_{FUT} / USERS_{TOD} * (1 + FLUCT) * (1 + FUNC_{NEW}) + CPU_{TOD} * B_{LOD}$$

$B_{LOD}$  La carga base es la carga que se genera independiente de la cantidad de usuarios en % (aun cuando no se tenga ningún proceso el simple hecho de tener una aplicación corriendo consume recursos)

$FLUCT$  Describe en % las fluctuaciones estadísticas de los picos en consumo de recursos (diario, semanales o por temporadas)

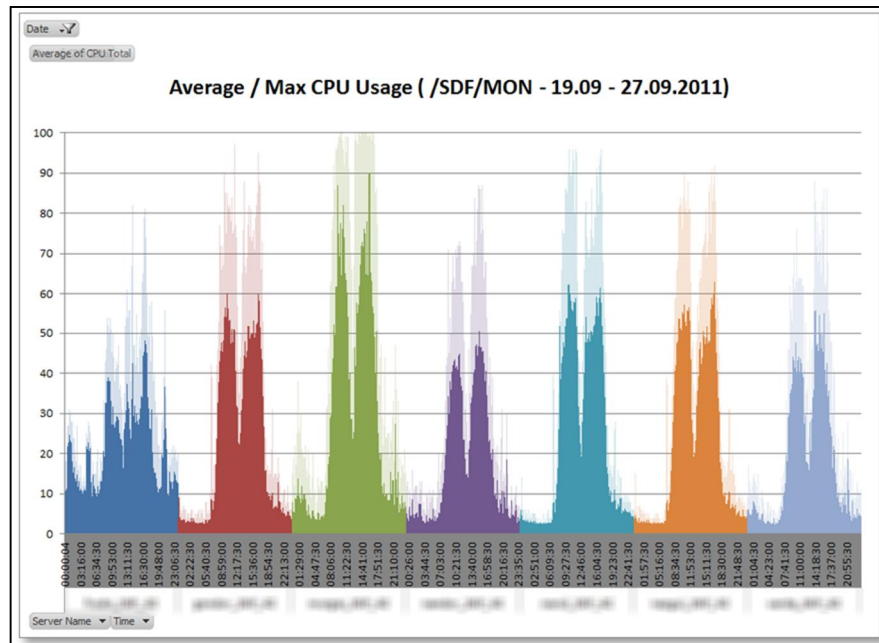
$FUNC_{NEW}$  Es un factor de escala que puede obtenerse a partir del quicksizer si se compara el consumo de recursos actual y el que agregaría la nueva funcionalidad.

$USERS_{FUT}$  Cantidad de usuarios que se planean tener a futuro

$USERS_{TOD}$  Cantidad de usuarios en sistema hoy en día

$CPU_{TOD}$  Consumo promedio pico de CPU

La grafica de a continuación muestra el consumo de CPU en promedio y en picos con un intervalo de muestreo de 1 minuto.



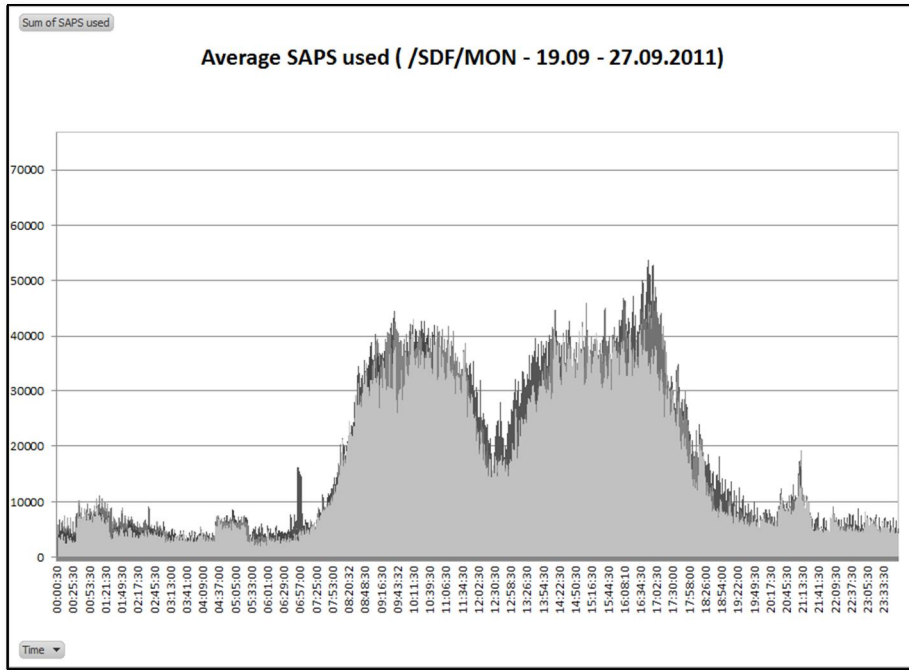
**Fig. 7.7-1 Monitoreo de CPU con alta frecuencia en varios servidores**

Si se considera que los servidores tienen una cantidad determinada de SAPS de acuerdo a los benchmarks de SAP

CPU Type	No. of CPU Cores	Frequency MHz	Memory in GB	SAPS	SAPS / CPU
US-IV+	40	1950	138	25028	626
US-IV	24	1350	54	8886	370
US-IV	24	1350	48	8886	370
US-IV	24	1350	48	8886	370
US-IV	8	1350	16	2962	370
US-IV+	16	1500	36	10132	633
US-IV+	16	1950	64	12038	752

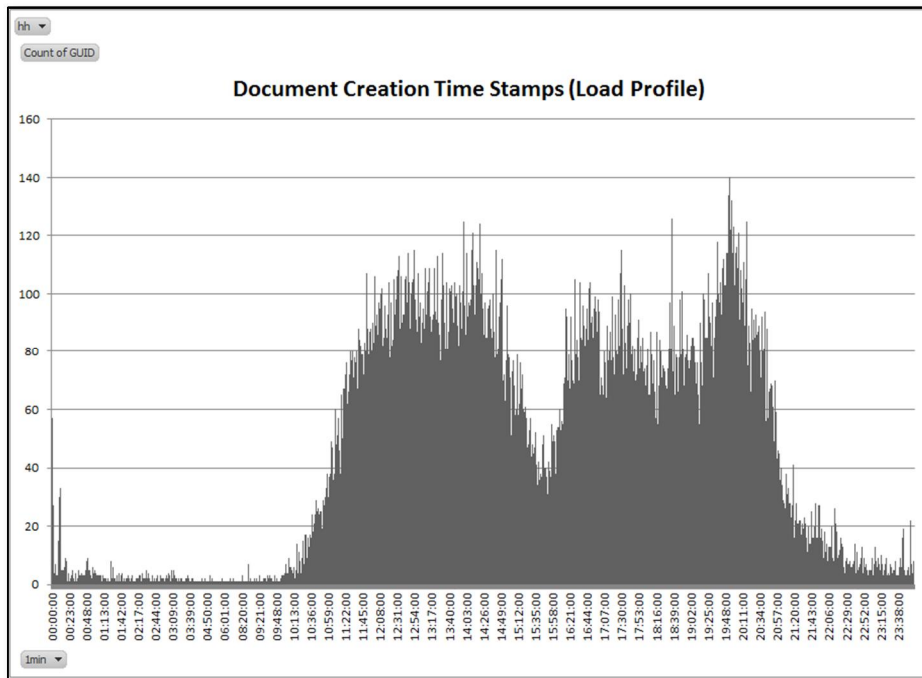
**Tabla 7.7-1 SAPS definidos por cada servidor**

Podemos acumular el consumo total de SAPS por todos los servidores y la aplicación.



**Fig. 7.7-2 Promedio de SAPS utilizados**

Una vez que tenemos el consumo total en SAPS, es importante también obtener la cantidad de documentos que se crean por unidad de tiempo y su distribución.



**Fig. 7.7-3 Número de documentos por unidad de tiempo**

Y finalmente la distribución de usuarios dentro del sistema trabajando activamente



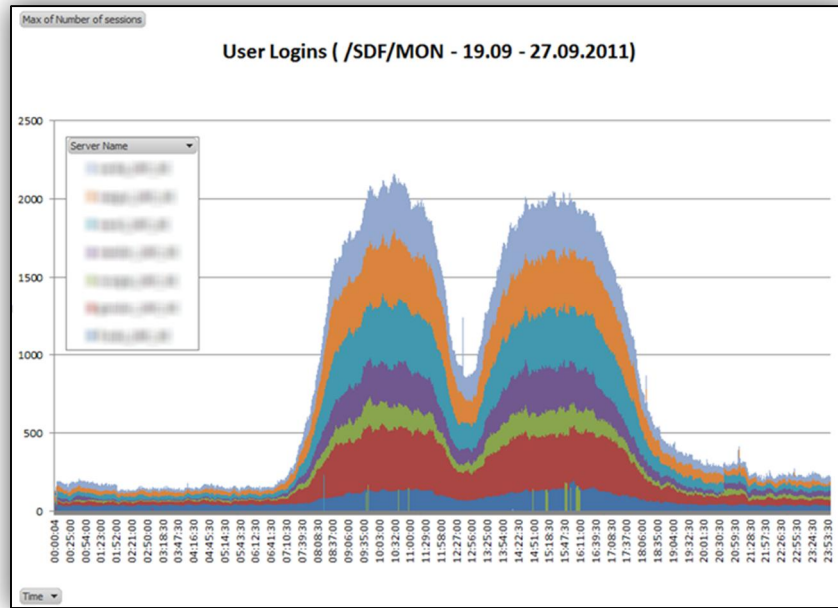


Fig. 7.7-4 Número de usuarios activos

Obtenemos las fluctuaciones con base en la cantidad de creación de documentos por día, y en base a un análisis de histograma podemos obtener entre un 20-30% de fluctuaciones en un periodo de nueve meses.

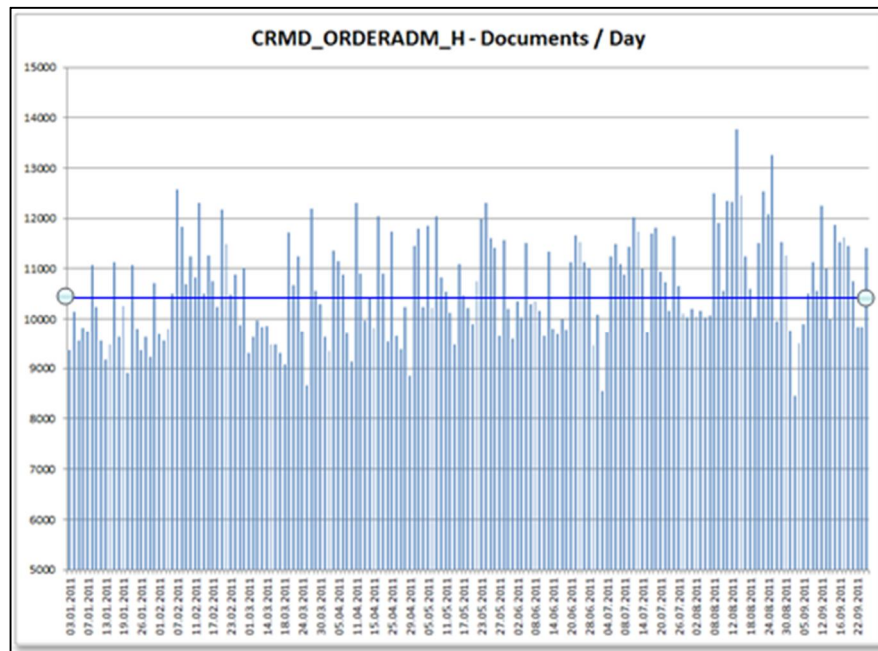


Fig. 7.7-5 Fluctuaciones en cantidad del número de documentos

Basados en la información anterior y considerando que el cliente planea agregar 1000 usuarios nuevos utilizamos la formula y obtenemos el resultado por día y la cantidad de SAPS que se requieren y lo graficamos como se muestra en la figura 7.7-6 donde conjuntamos la utilización de SAPS en base a la información recolectada, adicionando los búferes de seguridad para las fluctuaciones y comparando los recursos del servidor actual con el servidor que se va a utilizar.

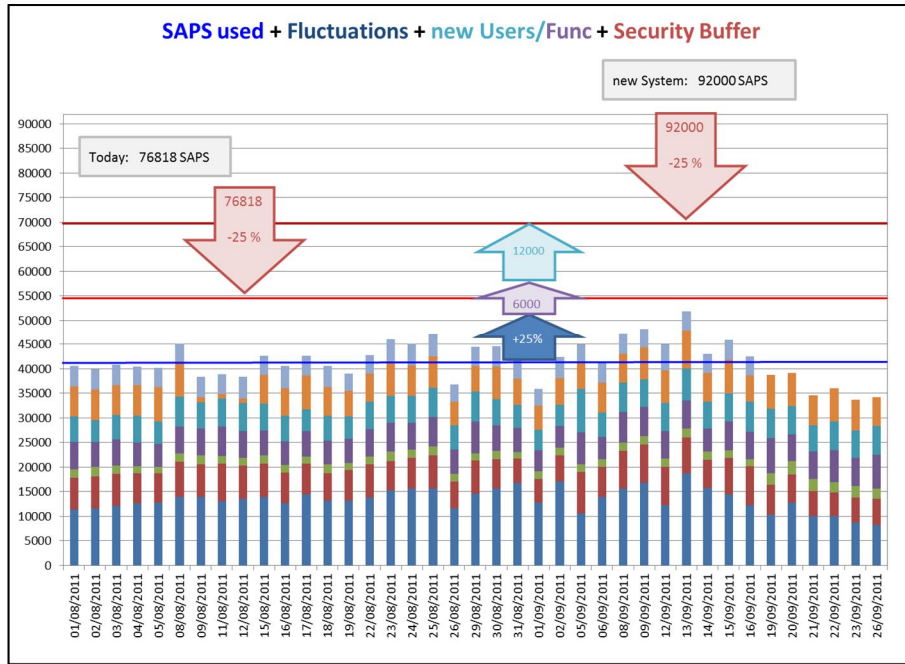


Fig. 7.7-6 Análisis completo de SAPS requeridos.

Y finalmente reemplazamos los valores de la fórmula que se describen en la siguiente tabla para saber si el futuro servidor cumple con los requerimientos del cliente.

		existing Server	future Server
<b>available SAPS</b>		<b>76818</b>	<b>92000</b>
- Security Buffer	25%	-19000	-19000
<b>Target Limit SAPS</b>		<b>57818</b>	<b>73000</b>
<b>SAPS used</b>		<b>42000</b>	<b>42000</b>
- Baseload	5%	-2100	-2100
+ Fluctuations	25%	9975	9975
+ add. Users	30%	11970	11970
+ new functionality	15%	5985	5985
+ Baseload		2100	2100
<b>Total SAPS</b>		<b>69930</b>	<b>69930</b>

Tabla 7.7-2 Análisis completo de SAPS requeridos.

## 8. Tendencias

Para finalizar este trabajo de investigación, en este capítulo se exploran las nuevas tecnologías de las que dispondrán las empresas para aumentar la agilidad en la toma de decisiones, reducir costos en el mantenimiento de sus sistemas y tener acceso a los datos en cualquier lugar en donde lo requieran.

Las nuevas tecnologías enfocadas al área de cómputo en memoria y movilidad, permitirán a las grandes empresas, el hecho de tener la información cada vez más accesible y en línea, así como también agilizar la ejecución de procesos de negocio que anteriormente tardaban horas a un par de minutos. Esta tendencia en el desarrollo tecnológico, hará posible optimizar las operaciones de las grandes empresas, y ayudar con esto a realizar la toma de decisiones de una manera más eficiente, con el hecho de tener la posibilidad de acceder reportes e información crítica en línea, de una forma simple y desde un dispositivo móvil a la mano [VA2011].

### 8.1. Base de datos en memoria

El volumen de datos que las empresas deben gestionar aumenta exponencialmente, pero disponen de menos tiempo para tomar decisiones en función de los datos relevantes.

Los procesadores multinúcleos y la disponibilidad de grandes cantidades de memoria a costos bajos están creando nuevos avances en la industria del software. Se ha alcanzado un punto en donde es posible almacenar conjuntos de datos enteros de una compañía en memoria, lo que ofrece un desempeño que es mucho más rápido comparado con sistemas tradicionales basados en discos duros. Los discos duros se volverán obsoletos. En un mundo de silicio la única parte mecánica que quedará, será aquella que se ocupe para respaldar la información. Con la computación “in-memory” y las bases de datos usando almacenamiento orientado a renglones y columnas, el procesamiento transaccional y analítico puede ser unificado. Existen técnicas de compresión que ayudan a evitar la cantidad de memoria física necesaria y con las características de los datos empresariales compresiones de hasta 80% son posibles [PHZA2011].

Con la tecnología “in-memory”, las compañías pueden aprovechar al máximo las grandes cantidades de datos para crear una ventaja estratégica. Los datos operacionales del negocio pueden ser analizados y consultados de forma interactiva sin apoyo del departamento de TI, abriendo de esa forma nuevos escenarios y oportunidades.

Consideremos el departamento de finanzas, en donde los datos tienen que ser agregados para crear los reportes, de forma diaria, semanal, mensual o anual. Con el manejo de datos “in-memory”, el agregado de datos y la filtración de los mismos se pueden llevar a cabo en tiempo real. La contabilidad se puede hacer en cualquier momento y de una forma ad-hoc. Las aplicaciones financieras no sólo serán significativamente más rápidas, sino que serán menos complejas, y fáciles de usar. La nueva información estará disponible para su análisis tan pronto como sea introducida en el sistema del negocio [PHZA2011].

En resumen, la tecnología “in-memory” podría impulsar los siguientes cambios:

- Reducción en el Costo total de Propiedad: Con los conceptos de manejo de datos en memoria, las capacidades de análisis requeridas están directamente incorporadas en los sistemas del negocio. Los sistemas dedicados a análisis ya no serán requeridos. Los sistemas empresariales se volverán menos complejos y más fáciles de mantener, teniendo como resultado, menos mantenimiento de hardware y menos recursos de TI necesarios.
- Aplicaciones Innovadoras: El manejo de datos en memoria combina transacciones de alto volumen con análisis en el sistema operacional. Planeación, predicción, optimización de precios, y otros procesos pueden ser mejorados dramáticamente y soportados por nuevas aplicaciones que no eran posibles antes.
- Mejores Decisiones y Más Rápidas: Sistemas empresariales basados en “in-memory” permiten el fácil y rápido acceso a la información que los tomadores de decisiones necesitan, aportándoles nuevas formas de ver el negocio. Simulaciones, análisis “¿qué tal si?”, y planeación se puede llevar a cabo de forma interactiva en los datos operacionales. La

información relevante es accesible de forma instantánea y la dependencia de recursos de TI se reduce.

## 8.2. Clientes móviles

Cerca de 500 millones de dispositivos móviles fueron vendidos en el último cuarto de 2011, con un crecimiento de 7.9% respecto al tercer cuarto del mismo año y un 5.4% respecto del mismo periodo del 2010. Los teléfonos inteligentes crecieron 29% en el tercer cuarto del 2011 y un 47% comparado con el tercer cuarto del 2010. Con sólidas ventas del iPhone 4S, Apple se convirtió en la compañía que más vende teléfonos inteligentes en el mundo. En Estados Unidos, RIM ahora tiene menos del 7.0% del mercado de teléfonos inteligentes, y Alcatel ingresó a los primeros diez lugares de las compañías que venden dispositivos móviles. – Gartner – Febrero 2012.

Según IDC (“International Data Corporation”, por sus siglas en inglés), en 2013, más de 1,190 millones de empleados de todo el mundo utilizarán la tecnología móvil, lo que representa un 34,9% de la población activa.

El impacto de la movilidad en las empresas es evidente. Cada vez más, se espera que los trabajadores de una empresa gestionen tareas de vital importancia y tomen decisiones en tiempo real, independientemente de donde se encuentren.

### Retos del negocio

- Se espera que los usuarios de soluciones empresariales manejen tareas críticas y tomen decisiones en tiempo real, no importando en dónde se encuentren.
- La información es lo que mueve a las compañías modernas. Al mismo tiempo que los trabajadores se vuelven más móviles, las compañías deben proporcionar información a esos trabajadores “al vuelo” y viceversa para una mejor colaboración.
- Los trabajadores tienen diferentes alternativas de obtener la información que necesitan. El acceso a sistemas “back-end” (tales como CRM, ERP, reportes financieros, y bases de datos) es crítico y la mayoría de los trabajadores, hoy en día, confían en la relación con colegas o sistemas dentro de la organización para que los provean de la información requerida.
- Diversidad de dispositivos y Sistemas Operativos.
- Las aplicaciones móviles son distantemente diferentes de las clásicas aplicaciones de escritorio.
- Las aplicaciones móviles no son mini versiones de las aplicaciones de escritorio existentes.
- La creación de buenas aplicaciones móviles requiere de una manera de pensar totalmente diferente (pensamiento móvil).

Las aplicaciones móviles están diseñadas tanto para usuarios que se desplazan como para usuarios fijos de la empresa que desean poder acceder a la información en cualquier momento. Este sistema conecta los procesos de la empresa con cualquier dispositivo para obtener acceso desde cualquier ubicación y siempre que se desee, incluso en modo desconectado.

Las empresas han comenzado a preocuparse en cómo resolver dichos retos y es así como surge un nuevo conjunto de herramientas conocidas como Plataforma para Aplicaciones Móviles Empresariales (MEAP, por sus siglas en inglés).

A continuación, se describen las características principales de estas plataformas.

### Plataforma para Aplicaciones Móviles Empresariales (PAME)

#### Propósito

La PAME ayuda a resolver las dificultades que se presentan en el desarrollo de software móvil, al manejar la diversidad de dispositivos, redes y grupos de usuarios al momento de la distribución y durante el ciclo de vida de las soluciones móviles.

Una ventaja importante de las PAMEs son las consideraciones multiplataforma que hay detrás de ellas. Por ejemplo, una compañía puede utilizar una PAME para desarrollar una aplicación móvil una vez y distribuirla a una variedad de dispositivos móviles (incluyendo teléfonos inteligentes, tabletas,

computadoras portátiles y dispositivos de mano de uso rudo) sin ningún cambio en la lógica del negocio.

### **Estructura**

Una PAME está generalmente compuesta de dos partes: un servidor de software intermedio móvil (mobile middleware) y un cliente de aplicaciones móviles. Un servidor de software intermedio es el componente que maneja la integración, seguridad, comunicación, escalabilidad, soporte multiplataforma, etc. La información no se guarda en el servidor de software intermedio (éste simplemente gestiona la información del sistema “back-end” hacia el dispositivo móvil y de regreso). La mayoría de las PAMEs también incorpora las herramientas que permiten a las compañías crear y ajustar las soluciones móviles.

Las aplicaciones móviles son el software que se conecta con el servidor de software intermedio y maneja tanto la interfaz del usuario como la lógica del negocio en el dispositivo. Las aplicaciones móviles pueden ser distribuidas como aplicaciones que se instalan en el dispositivo o presentadas en el navegador del dispositivo usando tecnologías tales como HTML5. El método de distribución depende en la complejidad de la aplicación, el dispositivo, los requerimientos de la experiencia del usuario y en la necesidad de la disponibilidad de la aplicación en caso de ausencia de cobertura de red [WIME2012].

### **Características y capacidades**

- Las PAMEs pueden soportar más de un tipo de dispositivo móvil y sistema operativo sin tener que mantener códigos fuente diferentes.
- Una PAME, generalmente, contiene un servidor de software intermedio móvil (mobile middleware) en dónde se administra la integración de la conectividad, seguridad y las aplicaciones.
- Se integra con múltiples fuentes de datos.
- Mejora las plataformas de negocios que existen actualmente al hacerlas accesibles a los usuarios en cualquier parte y en cualquier momento.

### **8.3. SAP en sistemas virtualizados (cloud computing)**

Las aplicaciones de negocios han traído consigo grandes beneficios en la operación de las grandes organizaciones, pero su vez también, han generado el reto de administrar una infraestructura de TI cada vez más compleja y costosa, la cual involucra la gestión de una variedad importante de plataformas tanto de hardware como de software, aunado a esto, se ha tenido la necesidad de contratar personal altamente especializado que sea capaz de instalar, configurar, probar, actualizar y en términos generales administrar la plataforma que servirá como base para la operación del negocio [CLRE2011].

Por este y otros motivos, a lo largo de los años las principales empresas de tecnología a nivel mundial han empezado a desarrollar una nueva tecnología conocida como “cloud computing” o traducido al idioma español “computación en la nube”.

Este concepto es nuevo un modelo de aprovisionamiento de servicios de TI, basado en Internet y centros de datos remotos, que permite gestionar la información y aplicaciones empresariales de una manera consolidada y eficiente, buscando realizar un mejor uso de la capacidad y recursos de hardware, así como también tratar de reducir en todo momento, los costos de operación y administración de la infraestructura computacional.

El término “nube” se utiliza de manera metafórica haciendo alusión al internet, en donde en este caso, no es necesario conocer la infraestructura de hardware que existe detrás de la nube para hacer uso de los servicios proporcionados, como se muestra en la figura 8.3-1:



**Fig. 8.3-1 Infraestructura en la nube.**

Al estar en la nube, los servicios deberán ser fácilmente escalables, deberán tener un tiempo de respuesta bueno de acuerdo al nivel de servicio acordado o SLA, así como contar con la característica de la alta disponibilidad. El término escalable, básicamente se refiere a que si crece la demanda de clientes hacia el aplicativo, existe la capacidad de incrementar los recursos en el “back-end” de una manera simple y eficiente en términos de tiempo y costos.

Una de las principales características por las que se define una arquitectura de computación de nube, es porque los aplicativos son del tipo virtualizados, esto se refiere a que múltiples servidores virtuales pueden ser creados a partir de un solo servidor físico, con recursos tanto de hardware como de software independientes entre sí, permitiendo en este caso que múltiples aplicaciones subsistan de manera virtual en un equipo físico, esta configuración permite que la administración y modificación de recursos tanto de hardware como de software, se pueda realizar en línea, requiriendo a lo más reinicios programados de los servidores virtuales para que los cambios surtan efecto [CLRE2011].

Adicionalmente esta arquitectura cuenta con la característica de ser multipropósito, básicamente este término se refiere a que la “nube” fue creada de tal manera, que permite el acceso a múltiples clientes desde diferentes tipos de dispositivos, sin preocuparse de ello y sin comprometer la seguridad y privacidad asociada entre los aplicativos.

### **Componentes de la computación en nube**

Los componentes más significativos de una arquitectura de computación en la nube son básicamente los siguientes:

- **Front-end:** Es el componente utilizado por parte del cliente, básicamente el dispositivo utilizado para conectarse a los servicios proporcionados por parte de la nube, llámese la computadora personal o cualquier dispositivo móvil que consumirá servicios en el “back-end”, así como la infraestructura de red del lado del usuario y la aplicación utilizada para acceder a la nube.
- **Back-end:** El back-end se compone por los dispositivos que comprenden la nube como tal, la cual incluye los servidores físicos, redes de almacenamiento masivo, balanceadores de cargas, ruteadores así como toda la infraestructura de comunicación e integración interna para operar y administrar la nube.

La implementación de un modelo de computación en la nube, se puede descomponer en tres segmentos fundamentales, software, plataforma e infraestructura, cada uno de estos segmentos tiene un propósito distinto y como tal, ofrece diferentes servicios a los usuarios finales, por tanto podemos contemplar los siguientes tipos de implementaciones [WICE2012]:

1. SaaS (Software as a Service) o el Software como un servicio.

En este caso el software es instalado en un ambiente de servidores centralizados que proveen acceso a la aplicación por medio de una intranet o internet, este tipo de implementación se conoce como “software bajo demanda”, ya que en lugar de comprar e instalar el software para que un usuario trabaje de manera local en su equipo, lo que se hace es rentar un servicio que proporcione acceso al aplicativo en todo momento desde un cliente externo, lo cual trae grandes ventajas desde el punto de vista de costos.

## 2. PaaS (Platform as a Service) o plataforma como un servicio.

Partiendo de la implementación de SaaS, la industria se está moviendo ahora a la implementación de PaaS, la cual ofrece una combinación de una plataforma de desarrollo en la que los programadores, pueden desarrollar nuevas aplicaciones, extender la funcionalidad de soluciones existentes, así como montarlas en una misma plataforma de integración, que proporciona el servicio a los usuarios finales.

## 3. IaaS (Infrastructure as a Service) o infraestructura como un servicio.

La implementación de IaaS conlleva a cabo un modelo completo de subcontratación, no solo de los recursos humanos responsables de administrar la operación de los sistemas tanto a nivel aplicación, base de datos como sistema operativo, sino una subcontratación también de los recursos de hardware e infraestructura completa, que soportará la operación del negocio.

Este modelo permite que las empresas, realicen la adquisición de infraestructura de hardware bajo demanda, en donde la adquisición de recursos se realiza dependiendo de los requerimientos que se tengan en un punto determinado en el tiempo, y no se tiene la necesidad de adquirir infraestructura adicional, que no es requerida en términos de capacidad de recursos de hardware.

En este caso, la empresa paga al proveedor del servicio únicamente la capacidad que está utilizando, y no subutiliza los recursos de hardware que no requiera para su operación diaria, con esto se logran reducir los costos por adquisición de infraestructura de cómputo de una manera significativa. Esta implementación presenta ventajas importantes no solo en la reducción de costos, sino también en la facilidad de escalar la infraestructura de una manera más rápida, dinámica y eficiente.

Esta implementación trabaja generalmente bajo modelos de virtualización y es conocida también como HaaS (Hardware as a Service), por el hecho de ofrecer y administrar el hardware como un servicio mismo.

### **Ventajas de la computación en nube.**

Las principales ventajas que presenta un modelo de implementación en la nube son las siguientes [CLRE2011]:

- Integración probada de servicios web. Por su naturaleza, la tecnología de “cloud computing” se puede integrar con mucha mayor facilidad y rapidez con el resto de sus aplicaciones empresariales (tanto software tradicional como “cloud computing” basado en infraestructuras), ya sean desarrolladas de manera interna o externa.
- Prestación de servicios a nivel mundial. Las infraestructuras de “cloud computing” proporcionan mayor capacidad de adaptación, recuperación de desastres completa y reducción al mínimo de los tiempos de inactividad.
- Una infraestructura 100% de “cloud computing” no necesita instalar ningún tipo de hardware. La belleza de la tecnología de “cloud computing” es su simplicidad y el hecho de que requiera mucha menor inversión para empezar a trabajar.
- Actualizaciones automáticas que no afectan negativamente a los recursos de TI. Si actualizamos a la última versión de la aplicación, nos veremos obligados a dedicar tiempo y recursos para volver a crear nuestras personalizaciones e integraciones. La tecnología de “cloud computing” no obliga a decidir entre actualizar y conservar su trabajo, porque esas personalizaciones e integraciones se conservan automáticamente durante la actualización.

### **Desventajas de la computación en nube.**

Las principales desventajas que presenta el hecho de implementar un modelo de computación en la nube son las siguientes [CCVD2011]:

- Seguridad de la información y privacidad de los usuarios

- Usar un servicio de cómputo en la nube para almacenar información, puede exponer al usuario a la violación de su privacidad, lo cual conlleva a un riesgo de exposición de datos sensibles o confidenciales. La información personal de un usuario es confiada a un proveedor que puede residir en un país diferente al del usuario. En caso de un comportamiento malicioso por parte del proveedor, podría acceder a la información con el objetivo de realizar estudios de mercado.
  - En el caso de industrias o corporaciones, toda la información guardada en memorias externas, está seriamente expuesta a posibles casos de ciber espionaje.
  - En el caso de utilizar servicios de cómputo en la nube en redes inalámbricas el riesgo en la seguridad se incrementa debido a la reducida seguridad ofrecida por las redes inalámbricas.
- Continuidad del servicio.
- Al delegar el manejo y procesamiento de la información a un proveedor externo a la empresa, se corre el riesgo de que los empleados se vean severamente afectados cuando se presenta una interrupción en el servicio, ya que cuando se presenta un incidente, existe el riesgo de detener operaciones críticas que pueden impactar al negocio. En algunos casos, el servicio se basa en una conexión de internet a alta velocidad tanto para subir información como para descargarla, y en el caso de un corte por ejemplo del enlace por parte del proveedor de internet, existe el riesgo de que los usuarios se vean totalmente imposibilitados a continuar con sus actividades laborales.
- Problemas de migración de datos al cambiar de proveedor de nube
- Cuando un usuario o compañía quiera cambiar su proveedor de nube, puede enfrentar problemas al migrar la información, ya que no hay un estándar definido entre los operadores, y un cambio como tal puede ser realmente complejo. Un caso de bancarrota de la compañía proveedora del servicio en la nube, podría ser extremadamente peligrosa para la empresa que contrato el servicio.



## Conclusiones

Al inicio de esta investigación se planteó que uno de los objetivos primordiales de este trabajo de tesis, consiste en realizar una exposición clara de los aspectos fundamentales para llevar a cabo un proyecto de implementación de ERP de una manera óptima, eficiente y exitosa.

La experiencia profesional nos ha permitido constatar que la falta de conocimientos y fundamentos técnicos sólidos al iniciar un proyecto, tendrán consecuencias negativas tales como retrasos en el proyecto, problemas técnicos que impactan la operación productiva del sistema y los costos relacionados. Por tal motivo, esta investigación se presenta como una base general para aquellos estudiantes de ingeniería en computación que decidan enfocar su carrera profesional a esta área de especialización.

En los primeros temas de la tesis se expusieron los puntos que se deben conocer al llevar a cabo la implementación de un ERP de SAP. Se planteó desde un inicio los fundamentos históricos y la evolución que han tenido los sistemas ERP para atender las necesidades de negocio de las empresas. Se revisan las metodologías recomendadas para ejecutar el proyecto de implementación.

La administración del sistema es una tarea primordial en la que, el profesional que toma dicha responsabilidad, debe tener un conocimiento extensivo y profundo en los siguientes campos: la arquitectura de la solución; las tareas de mantenimiento necesarias para garantizar la disponibilidad y el rendimiento eficiente del sistema. Este conocimiento se ofrece en los capítulos 5, 6 y 7.

Durante el desarrollo de este trabajo se mostraron los componentes técnicos que un ingeniero debe conocer para realizar una administración eficiente del sistema. Se describieron las herramientas de monitoreo que son utilizadas diariamente para identificar el comportamiento del sistema, conocer el rendimiento actual e histórico de la operación, así como los puntos principales en los cuales hay que poner atención para mantener y optimizar el rendimiento general de la solución.

El último capítulo tiene la intención de interesar a los ingenieros en computación en las nuevas tecnologías que están adoptando las empresas a nivel global, ya que difícilmente se tiene permeada esta información hacia las universidades. El compartir este conocimiento con los estudiantes recién egresados, representa una ventaja competitiva porque les permite tener un acercamiento a dichas tecnologías antes de enfrentarse a ellas en el mundo laboral.

Finalmente consideramos que el trabajo de investigación cumple con la tarea de transferir el conocimiento adquirido durante nuestra vida profesional y se entrega un documento que servirá como referencia a los alumnos que deseen incorporarse a la administración de sistemas ERP de SAP.

## Bibliografía

- [VK2005] Andreas Vogel y Ian Kimbell, mySAP ERP For Dummies, For Dummies(CA), 2005
- [CZ1990] Cooper, R.B., Zmud, Information technology implementation research: a technological diffusion approach. Management Science (1990)
- [MCZ2005] José Martínez, Martí Casadesús, Ibon Zamanillo, Evolución histórica de los sistemas ERP(2005). [Fecha de consulta: 20 Octubre 2012]. Disponible en: <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r19487.PDF>
- [DK2012] Derek Singleton, Manufacturing and ERP Software, 10.2012. [Fecha de consulta: 11 Noviembre 2012]. Disponible en: [Historyhttp://www.softwareadvice.com/manufacturing/#timeline](http://www.softwareadvice.com/manufacturing/#timeline)
- [EHT2012] Enterprise-Resource-Planning-History-Timeline, 2012. [Fecha de consulta: 07 Noviembre 2012]. Disponible en: <http://www.xtimeline.com/timeline/Enterprise-Resource-Planning-History-Timeline>
- [CCVD2011] Cloud Computing. Ventajas y desventajas. [Fecha de consulta: 23 Mayo 2011]. Disponible en: <http://bcnbinaryblog.com/cloud-computing-ventajas-y-desventajas/>
- [PI2012] Project Inspiration, Celebrating 40 year of SAP 2012. [Fecha de consulta: 23 Abril 2012]. Disponible en: <http://sap-project-inspiration.com/de/past>
- [LRW2012] SAP Feiert 40 Jahre Innovation, Christoph Liedtke, Daniel Reinhardt, Christoph Weissthauer, Walldorf 2012. [Fecha de consulta: 23 Abril 2012]. Disponible en: <http://www.sap.com/germany/about/press/index.epx.epx?pressid=18582>
- [VB2009] Enterprise Software Top 10, Balder Verberne, 2009. [Fecha de consulta: 7 Marzo 2012]. Disponible en: <http://www.softwaretop100.org/enterprise-software-top-10>
- [SAP01\_2009] SAP, Material de Entrenamiento SAP01, SAP Overview, 2009
- [SOA200\_2009] SAP, SOA200 SAP Enterprise Architecture Framework Level, 2009
- [FHW2009] Frank Föse, Sigrid Hagemann, Liane Will. SAP Netweaver AS ABAP System Administration. SAP System administration fundamentals. Galileo Press. Bostón (MA), 2009.
- [GH2009] Hermann Gahm, ABAP Performance Tunning, Galileo Press, 2009
- [SS2011] Sebastian Schreckenbach, SAP Administration, Galileo Press, 2011
- [KHT2009] Horst Keller, Wolf Hagen Thümmel, Official ABAP Programming guidelines, Galileo Press, 2009
- [AS2004] Andreas Schneider. SAP System Landscape Optimization. Galileo Press, 2004.
- [ST2010] Thomas Schneider, SAP Performance Optimization Guide, Galileo Press, 2010
- [GO2010]. Gerhard Oswald. SAP Enterprise Support - ASAP to Run SAP (2nd Edition). Galileo Press, 2010.
- [PHZA2011] Hasso Plattner, Alexander Zeier. In-Memory Data Management: An Inflection Point for Enterprise Applications. Editorial Springer. 2011.
- [TADM10\_12005] SAP. TADM10\_1 SAP Web AS Implementation & Operation I. SAP AG. 2005.

- [TADM10\_22005] SAP. TADM10\_2 SAP Web AS Implementation & Operation I. SAP AG. 2005.
- [TADM12\_12005] SAP. TADM12\_1 SAP Web AS Implementation & Operation II. SAP AG. 2005.
- [TADM12\_22005] SAP. TADM12\_2 SAP Web AS Implementation & Operation II. SAP AG. 2005.
- Andreas Blumenthal, Horst Keller, State-of-the-Art ABAP: Guidelines for Writing Robust, Understandable, and Maintainable ABAP Programs, 2009
- [VA2011] Vasco Alexander. Collaboration in the cloud. SAP Spectrum the customer magazine, 2011
- [VM2012] Michael van Kooten, Global software top 100 edition 2011: the highlights. [Fecha de consulta: 10 enero 2012]. Disponible en:  
<http://www.softwaretop100.org/global-software-top-100-edition-2011-the-highlights>
- [SCSO2011] SAP community network. Service-Oriented Architecture (SOA) and SAP. [Fecha de consulta: 8 Diciembre 2011]. Disponible en:  
<http://scn.sap.com/community/soa>
- [CLRE2011] Cloud computing. Retos y oportunidades. [ Fecha de consulta: 10 Agosto 2011]. Disponible en:  
[http://www.fundacionideas.es/sites/default/files/pdf/DT-Cloud\\_Computing-Ec.pdf](http://www.fundacionideas.es/sites/default/files/pdf/DT-Cloud_Computing-Ec.pdf)
- [SHAR2011] SAP help portal. ASAP Roadmap. [Fecha de consulta 25 octubre: 2011]. Disponible en:  
[http://help.sap.com/saphelp\\_47x200/helpdata/en/48/623972d55a11d2bbf700105a5e5b3c/content.htm](http://help.sap.com/saphelp_47x200/helpdata/en/48/623972d55a11d2bbf700105a5e5b3c/content.htm)
- [SCR2011] SAP community network. Roadmap and phases. [Fecha de consulta: 20 de Septiembre 2011]. Disponible en:  
<http://scn.sap.com/docs/DOC-8032>
- [SCAM2011] SAP community network. ASAP Methodology for implementation. [Fecha de consulta: 25 Septiembre 2011]. Disponible en:  
<http://scn.sap.com/community/asap-methodology>
- [SPMR2011] SAP portal. Metodología RunSAP. [Fecha de consulta: 15 Noviembre 2011]. Disponible en:  
<http://www.sap.com/mexico/services/support/pdf/Run%20SAP%20Methodology%20%20How%20to%20Implement%20End-to-End%20Solution%20Operations%20%20.pdf>
- [SPJS2011] SAP help portal. Job scheduling management. [Fecha de consulta: 10 Noviembre 2011]. Disponible en:  
[http://help.sap.com/saphelp\\_smehp1/helpdata/en/c5/7edcbd58ce4e29bc1baf1c4761d71f/content.htm](http://help.sap.com/saphelp_smehp1/helpdata/en/c5/7edcbd58ce4e29bc1baf1c4761d71f/content.htm)
- [WISP2011] Wikipedia. Sistemas de planificación de recursos empresariales. [Fecha de consulta: 20 Mayo 2011] Disponible en:  
[http://es.wikipedia.org/wiki/Sistema\\_de\\_planificaci%C3%B3n\\_de\\_recursos\\_empresariales](http://es.wikipedia.org/wiki/Sistema_de_planificaci%C3%B3n_de_recursos_empresariales)
- [SHCW2012] SAP help portal. Configuration wizard. [Fecha de consulta: 20 Junio 2012]. Disponible en:  
[http://help.sap.com/saphelp\\_nw70/helpdata/en/98/fa9b6ec510404dbcfefbf35cb107c04/content.htm](http://help.sap.com/saphelp_nw70/helpdata/en/98/fa9b6ec510404dbcfefbf35cb107c04/content.htm)
- [SHT2011] SAP community network. How to design a SAP Netweaver based system landscape. [Fecha de consulta: 20 Septiembre 2011]. Disponible en:

---

<http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/50a9952d-15cc-2a10-84a9-fd9184f35366?QuickLink=index&overridelayout=true&26164940770445>

- [WICS2011] Wikipedia. Cliente-servidor. [Fecha de consulta: 16 Junio 2011]. Disponible en: <http://es.wikipedia.org/wiki/Cliente-servidor>
- [WISM2011] Wikipedia. Solution Manager [Fecha de consulta: 10 Noviembre 2011]. Disponible en: [http://en.wikipedia.org/wiki/SAP\\_Solution\\_Manager](http://en.wikipedia.org/wiki/SAP_Solution_Manager)
- [SHSM2012] SAP help portal. Solution Manager. [Fecha de consulta: 10 Junio 2012]. Disponible en: <https://websmp110.sap-ag.de/solutionmanager>
- [SCSM2011] SAP community network. SAP Solution Manager ITIL support. [Fecha de consulta: 8 de Octubre 2011]. Disponible en: [http://rbe-plus.ru/pdf/solman\\_ITIL.pdf](http://rbe-plus.ru/pdf/solman_ITIL.pdf)
- [SHJC2011] SAP help portal. J2EE cluster architecture. [Fecha de consulta: 20 Febrero 2011]. Disponible en: [http://help.sap.com/saphelp\\_webas630/helpdata/en/2e/611724f410254ca12a3f396ec5ae85/frame.htm](http://help.sap.com/saphelp_webas630/helpdata/en/2e/611724f410254ca12a3f396ec5ae85/frame.htm)
- [WIIT2012] Wikipedia. ITIL. [Fecha de consulta: 10 de Octubre 2012]. Disponible en: [http://es.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://es.wikipedia.org/wiki/Information_Technology_Infrastructure_Library)
- [SHCS2011] SAP help portal. Central services instance. [Fecha de consulta: 11 Noviembre 2011]. Disponible en: [http://help.sap.com/saphelp\\_nwpi711/helpdata/en/48/0728f74c6a3837e10000000a42189b/content.htm](http://help.sap.com/saphelp_nwpi711/helpdata/en/48/0728f74c6a3837e10000000a42189b/content.htm)
- [SCTI2012] SAP community network. Technical implementation. [Fecha de consulta: 10 Enero 2012]. Disponible en: <http://scn.sap.com/docs/DOC-8848>
- [SHJA2011] SAP help portal. Java cluster architecture. [Fecha de consulta: 20 Julio 2011]. Disponible en: [https://help.sap.com/saphelp\\_srm40/helpdata/fr/2e/611724f410254ca12a3f396ec5ae85/content.htm](https://help.sap.com/saphelp_srm40/helpdata/fr/2e/611724f410254ca12a3f396ec5ae85/content.htm)
- [SHSE2011] SAP help portal. Standalone enqueue server. [Fecha de consulta: 10 Julio 2011]. Disponible en: [http://help.sap.com/saphelp\\_webas630/helpdata/en/36/67973c3f5aff39e10000000a114084/content.htm](http://help.sap.com/saphelp_webas630/helpdata/en/36/67973c3f5aff39e10000000a114084/content.htm)
- [WIJL2011] Wikipedia. Java (lenguaje de programación). [Fecha de consulta: 23 Abril 2011]. Disponible en: [http://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
- [WIEJ2011] Wikipedia. Enterprise JavaBeans. [Fecha de consulta: 11 Mayo 2011]. Disponible en: [http://es.wikipedia.org/wiki/Enterprise\\_JavaBeans](http://es.wikipedia.org/wiki/Enterprise_JavaBeans)
- [BS2011] Bernhard Steigleider. HA Setups for SAP Netweaver 04. [Fecha de consulta: 20 Agosto 2011]. Disponible en: <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/c029ec90-0201-0010-b192-8306f27295c2?QuickLink=events&overridelayout=true&5003637670574>
- [SPSI2012] SAP portal. SAP in-memory computing. [Fecha de consulta: 20 Febrero 2012]. Disponible en: <http://www.sap.com/mexico/platform/in-memory-computing/index.epx>

- [WICE2012] Wikipedia. Computación en la nube. [Fecha de consulta: 1 Octubre 2012]. Disponible en: [http://en.wikipedia.org/wiki/Cloud\\_computing](http://en.wikipedia.org/wiki/Cloud_computing)
- [IBMPTG2011] IBM, Performance tuning guidelines [Fecha de consulta: 20 Septiembre 2012]. Disponible en: <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/admin/c0005418.htm>
- [SHOSC2011] SAP Help Portal, Operating System Collector SAPOSCOL [Fecha de consulta: 12 Noviembre 2011]. Disponible en: [http://help.sap.com/saphelp\\_nw04/helpdata/en/c4/3a6bff505211d189550000e829fbbd/content.htm](http://help.sap.com/saphelp_nw04/helpdata/en/c4/3a6bff505211d189550000e829fbbd/content.htm)
- [TT2012] Tony De Thomasis, Take off with the DBACOCKPIT, 2009. [Fecha de consulta: 10 Marzo 2012]. Disponible en: <http://scn.sap.com/people/tony.dethomasis2/blog/2009/05/11/take-off-with-the-dbacockpit>
- [FM2009] Martin Frauendorfer, Oracle 10g Performance Case Studies, 2009. [Fecha de consulta: 10 Marzo 2012]. Disponible en: <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/d0161109-410c-2a10-4592-afd1586b66c1?quicklink=index&overridelayout=true>
- [FM619788] Martin Frauendorfer, SAP Note 619188 - FAQ: Oracle wait events. [Fecha de consulta: 11 Marzo 2012]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=619188>
- [VL912620] Lutz Vornefeld, SAP Note 912620 - FAQ: Oracle indexes. [Fecha de consulta: 18 Marzo 2012]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=912620>
- [FM766349] Martin Frauendorfer, SAP Note 766349 - FAQ: Oracle SQL optimization. [Fecha de consulta: 18 Marzo 2012]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=766349>
- [FM766349] Martin Frauendorfer, Note 766349 - FAQ: Oracle SQL optimizationts. [Fecha de consulta: 18 Marzo 2012]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=766349>
- [FM772497] Martin Frauendorfer, SAP Note 772497 FAQ: Oracle Hints. [Fecha de consulta: 18 Marzo 2012]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=772497>
- [SA723909] SAP, SAP Note 723909 Java VM settings for J2EE 6.40/7.0. [Fecha de consulta: 25 Junio 2011]. Disponible en (Usuario SAP requerido): <https://service.sap.com/bc/bsp/spn/sapnotes/index2.htm?numm=723909>
- [BS2007] Siegfried Boes, Performance Problems Caused by Nonlinear Coding, 2007. [Fecha de consulta: 30 Enero 2012]. Disponible en: <http://scn.sap.com/community/abap/testing-and-troubleshooting/blog/2007/02/12/performance-problems-caused-by-nonlinear-coding>
- [SAPSZ2012] SAP, Información sobre sizing. [Fecha de consulta: 29 Junio 2012]. Disponible en: <http://service.sap.com/sizing>
- [SAPBM2012] SAP, Información sobre los benchmarks de SAP. [Fecha de consulta: 29 Junio 2012]. Disponible en: <http://www.sap.com/campaigns/benchmark/index.epx>
- [IBCC2012] IBM portal. Cloud computing: reducir costos, mejorar costos, permitir la innovación empresarial [Fecha de consulta: 5 Junio 2012]. Disponible en: <http://www.ibm.com/mx/services/cloud.phtml>

- [SAPCI2012] Información de Clientes Móviles en [Fecha de consulta: 17 Junio 2012]. Disponible en: <http://www.sap.com/mexico/solutions/mobility/index.epx>
- [WIME2012] Wikipedia. Mobile enterprise application platform. [Fecha de consulta: 26 Febrero 2012]. Disponible en: [http://en.wikipedia.org/wiki/Mobile\\_enterprise\\_application\\_platform](http://en.wikipedia.org/wiki/Mobile_enterprise_application_platform)
- [GA2012] Gartner. Market Share Analysis: Mobile Devices, Worldwide, 4Q11 and 2011. [Fecha de consulta: 10 mayo 2012]. Disponible en: <http://www.gartner.com/id=1923315>
- [FM2011] Finnegan, Michael. El desarrollo de aplicaciones móviles requiere nuevo enfoque. [Fecha de consulta: 18 octubre 2011]. Disponible en: <http://www.informationweek.com.mx/movilidad/el-desarrollo-de-aplicaciones-moviles-requiere-nuevo-enfoque/>

## Glosario

### **ABAP**

“Advanced Business Application Programming”. Lenguaje de programación propietario de SAP para el desarrollo de programas de aplicación.

### **ABAP Workbench**

Ambiente gráfico de desarrollo integrado en el sistema SAP. El ABAP Workbench le permite desarrollar, modificar, probar y manejar aplicaciones cliente/servidor que están escritas en ABAP.

### **Ad-hoc**

Expresión para referirse a lo que se dice o hace sólo para un fin determinado.

### **Applet**

Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El applet debe ejecutarse en un contenedor, que lo proporciona un programa anfitrión, mediante un plugin, o en aplicaciones como teléfonos móviles que soportan el modelo de programación por "applets". A diferencia de un programa, un applet no puede ejecutarse de manera independiente.

### **ASAP**

Es una metodología diseñada para soportar los proyectos de implementación de soluciones SAP, la cual se basa en 5 fases.

### **Back-end**

Por lo general este concepto hace referencia al servidor de base de datos y servidor de aplicaciones, en el contexto de sistemas, el back-end se define en contraste con el front-end el cual suele representar la interfaz de comunicación con el usuario final.

### **BAPI**

“Business Application Programming Interface”. Es una interface de programación estandarizada que facilita el acceso tanto interno como externo a los procesos de negocio y a la información en los sistemas SAP.

### **BI**

“Business Intelligence” Inteligencia de Negocios

### **Business blueprint**

Es un documento utilizado para documentar los procesos de negocio de la empresa que serán implementados en el sistema, aquí es donde se describe de manera clara los escenarios y pasos asociados a un proceso de negocio de manera clara y estructurada.

### **Business Server Page**

Una “Business Server Page” (BSP) es una unidad de almacenamiento (HTML, XML) que puede contener una secuencia de comandos que se deben de ejecutar en el lado del servidor. Una BSP es, por lo tanto, parte de una aplicación BSP y contiene la estructura, todos los manejadores de eventos, los atributos de la página y la definición de los tipos.

### **BW**

“Business Warehouse” Se refiere al sistema encargado de almacenar datos históricos para su análisis.

### **Ciente**

Componente de software que utiliza los servicios que le provee un servidor (definición desde una perspectiva de software) o estación de trabajo a la cual un servidor suministra de recursos (definición desde una perspectiva de hardware).

### **Clúster**

El término clúster, se aplica a los conjuntos o conglomerados de computadoras contruidos mediante la utilización de hardware o software comunes y que se comportan como si fuesen una entidad única e independiente.

**Cutover**

Transferencia de datos desde el ambiente de calidad y sistemas legados del sistema anterior hacia el nuevo ambiente productivo. Las actividades de cutover incluyen la inicialización del ambiente productivo, el cierre de los sistemas legados así como el ingreso manual de información hacia el nuevo sistema.

**Despachador**

Proceso central en una instancia. Es responsable, entre otras cosas, de iniciar los procesos de trabajo y distribuir la carga de trabajo entre los procesos de trabajo.

**Dynpro**

“DYNAmic PROgram”. Un dynpro consiste de una pantalla y del lógico subyacente a dicha pantalla, programada en ABAP.

**EJB**

“Enterprise Java Beans”. Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB.

**ERP**

Planeación de Recursos de la Empresa “Enterprise Resource Planning”.

**Failover**

Es el cambio automático ya sea de un servidor, un sistema o una red al momento de detectar una falla o terminación anormal de una aplicación que se encontraba activa anteriormente, este cambio se realiza sin la intervención humana al momento que se presenta la falla con el objetivo de restablecer el servicio en el menor tiempo posible.

**Firewall**

Es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas, se compone de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

**Front-end**

Por lo general, el “front-end” hace referencia al dispositivo o entidad que establece la comunicación con el sistema de “back-end” para procesar una petición, en contraste el “back-end” representa el dispositivo que da atención a las peticiones provenientes del “front-end”.

**Gateway**

Interfaz que convierte un protocolo de comunicación en otro protocolo de comunicación.

**HTML**

“HyperText Markup Language”. Es el lenguaje predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

**HTTP**

Protocolo de aplicación “World Wide Web” (WWW). HyperText Transfer Protocol, en inglés. El Protocolo de Transferencia de HiperTexto controla la comunicación entre el navegador web (cliente HTTP) y el servidor web (servidor HTTP).

**ITIL**

Es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma en general, ITIL da descripciones detalladas de un extenso conjunto de procedimientos de gestión ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI.



### **ICM**

“Internet Communication Manager”. El Gestor de Comunicación de Internet es el componente en la arquitectura SAP que permite al sistema SAP comunicarse directamente con Internet. Es iniciado por el despachador de la instancia.

### **IDC**

“International Data Corporation”. Es una firma de investigación y análisis de mercado que se especializa en tecnologías de la información, telecomunicaciones y tecnología de consumo. IDC es una subsidiaria de “International Data Group”. IDC tiene sus oficinas centrales en Framingham, Massachussets y afirma tener más de 1000 analistas en 50 países.

### **Java**

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

### **JSP**

“Java Server Pages”. Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten la utilización de código Java mediante scripts.

### **Kernel**

Es el módulo central de un sistema que carga primero y permanece en la memoria principal, es importante que el “kernel” también conocido como núcleo sea tan pequeño como sea posible, el núcleo es responsable además de proporcionar todos los servicios esenciales requeridos por otras partes del sistema operativo y las aplicaciones, tales como la administración de la memoria, los procesos y tareas centrales, así como la gestión de tareas a disco.

### **Landscape**

Se refiere a la infraestructura técnica de servidores que soporta la operación de las soluciones SAP.

### **Log**

Un log es un registro oficial de eventos durante un rango de tiempo en particular.

### **Mandante**

Un mandante generalmente representa a una compañía en un sistema SAP. Esto significa que si un sistema SAP tiene varios mandantes, entonces, varias compañías pueden ser representadas y simultáneamente activas en ese sistema. El mandante tiene una llave correspondiente en los campos en las tablas de la base de datos para ese sistema SAP. Si usted se firma en un mandante, sólo podrá visualizar información de ese mandante. Los mandantes, por lo tanto, corresponden a entidades de negocio independientes.

### **SAP MDM**

SAP Master Data Management o simplemente Administración de Datos Maestros. Es una herramienta más de la familia Netweaver diseñada para tener un solo un registro único de información que alimenta a toda la organización

### **SAP PI**

SAP Process Integration. Es una herramienta más de la familia Netweaver para poder integrar los sistemas y procesos con estándares de comunicación. Es el punto central de SOA.

### **Netweaver**

Es la plataforma tecnológica desarrollada por SAP para dar soporte a la mayor parte de las soluciones desarrolladas por esta empresa de software, es conocida como una aplicación orientada a servicios y a la integración. Provee al usuario de un vínculo entre lenguajes y aplicaciones, está construido usando estándares abiertos de la industria por lo que es posible negociar transacciones de información con desarrollos de Microsoft .NET, Sun Java EE, e IBM WebSphere.

### **Notas de SAP**

Una nota de SAP es un texto informativo acerca de un problema particular o de un mensaje del sistema que puede surgir como parte del trabajo que se realiza con el sistema. Todas las notas de SAP están almacenadas en SAP en una base de datos que está disponible en línea, y puede ser consultada por los clientes.

**Proceso de Segundo Plano (background job)**

Un proceso de segundo plano describe los pasos que serán ejecutados en el sistema sin la interacción del usuario (en segundo plano). Los procesos de segundo plano son procesados por procesos de trabajo de segundo plano.

**Roadmap**

Es una hoja de planificación del desarrollo de un software con los objetivos a corto y largo plazo incluyendo unos plazos aproximados de consecución de cada uno de estos objetivos.

**SAProuter**

Es un programa utilizado para controlar el acceso remoto hacia los ambientes de SAP, generalmente fuera del segmento de red donde se alojan los sistemas, es posible configurar las direcciones de red así como los puertos asociados a las soluciones sobre las cuales se dará acceso por medio de reglas de ruteo.

**SDM**

“Software Deployment Manager”. El Gestor de Distribución de Software es la herramienta estándar con la que se instalan los componentes J2EE en un Servidor de Aplicaciones Java. Es parte del ambiente de ejecución de Java y siempre se instala en la parte de Java de la instancia central.

**Servlet**

La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor.

El uso más común de los servlets es generar todas páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

**Solution Manager**

Es una solución centralizada y robusta compuesta por un conjunto de herramientas utilizadas para la administración de soluciones, permite a las compañías implementar, operar, monitorear y dar soporte a sus soluciones instaladas sobre la plataforma de SAP.

**Support Package**

Transporte que contiene corrección de errores y nuevas/mejoradas funcionalidades para un componente de un sistema.

**Stack de ABAP/Java**

Se refiere a la utilización del servidor de aplicación con tecnología fundamentada en una pila de la máquina virtual de ABAP o Java respectivamente

**Trace**

Es un tipo especial de archivo que guarda información detallada de la ejecución de un programa. Esta información es generalmente usada para diagnosticar problemas.

**Tunning**

Afinar la configuración de hardware o software para optimizar su rendimiento.

**UNIX**

Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

**URL**

Uniform Resource Locator. Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en internet para su localización o identificación.

**Webdispatcher**

Es un balanceador de cargas desarrollador por SAP que funge como punto único de acceso para los sistemas SAP, ya sea para conexiones desde los sistemas satélite hacia un ambiente central o de los usuarios finales que ingresan al sistema.

### **Web de integración**

Permite a los usuarios tener información y funciones a través de un navegador de un servidor de aplicaciones.

### **XML**

eXtensible Markup Language. Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos.