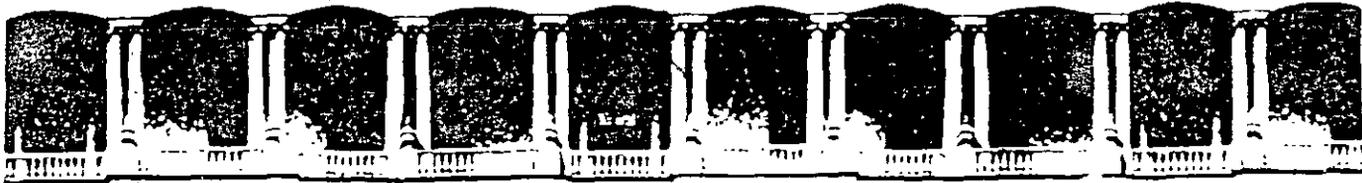


CI-173
2000



**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

CURSOS INSTITUCIONALES

**DIRECCIÓN GENERAL DE SERVICIOS
URBANOS
GOBIERNO DEL DISTRITO FEDERAL**

REINGENIERÍA DE PROCESOS.

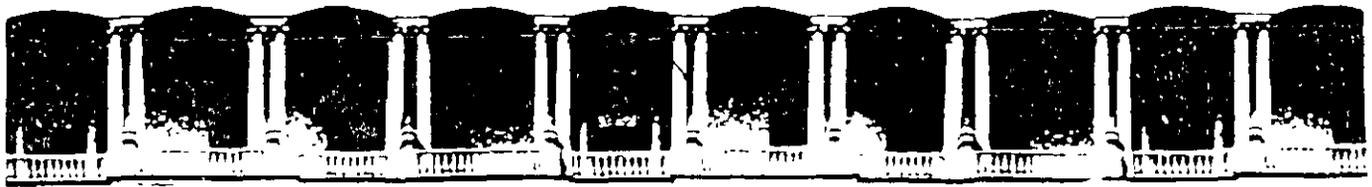
Del 25 de Septiembre al 06 de Octubre del 2000.

Apuntes Generales.

Mtro. Rómulo Mejías Ruiz
Gobierno del D.F.
Septiembre - Octubre/ 2000







FACULTAD DE INGENIERIA U.N.A.M. DIVISION DE EDUCACION CONTINUA

REINGENIERIA DE PROCESOS

OBJETIVO:

Que el participante esté en capacidad de eficientar procesos de trabajo para que se realicen en mucho menos tiempo, a menor costo, con menos errores y con mejor calidad de servicio a usuarios internos y externos del Gobierno del Distrito Federal.

DIRIGIDO A:

Mandos medios y superiores, servidores públicos, dirigentes políticos y personas interesadas en el tema.

DURACION: 20 horas

EXPOSITORES:

- * M. en I. Rómulo Mejías Ruiz
- * Ing. Gabriel Clavel Carmona

METODOLOGIA:

Exposiciones interactivas, ejercicios de aplicación de la Reingeniería a procesos de trabajo en la Administración Pública.

TEMAS:

1. Origen, razón de ser, principios, beneficios, costos y retos de la Reingeniería.
2. Visión en los procesos. Ejemplos y ejercicios de identificación de procesos y de tipificación de pasos.
3. Determinación de la eficiencia y deficiencia de procesos de trabajo.
4. Diseño técnico de innovaciones aplicando los principios básicos de la Reingeniería.
5. Aplicación de la Reingeniería de procesos en diferentes áreas.
6. Evaluación y selección de opciones de innovación
7. Estrategias y tácticas de negociación para hacer viable el éxito en la implantación de las innovaciones.

TEMA 1: ORIGEN, CONCEPTO, RAZON DE SER, PRINCIPIOS, BENEFICIOS, COSTOS Y RETOS DE LA REINGENIERIA

- 1. ORIGEN DE LA REINGENIERIA:** A mediados de los años ochenta algunas compañías norteamericanas decidieron mejorar espectacularmente su rendimiento, cambiando radicalmente las formas en que trabajaban.

Para lograr estas mejoras, se preguntaban:

- ¿ Por qué hacemos esto ?
- ¿ Por qué no hacemos otra cosa que nos produzca grandes resultados ?

También se preguntaban:

- ¿ Lo que estamos haciendo, a quién satisface más, al cliente o a nuestra empresa?
- ¿ Quién es primero, el cliente o la empresa ?

Al investigar bien cómo funcionaban, encontraron que a los trabajadores les importaba más quedar bien con sus jefes que con los clientes. Entonces, comenzaron a preguntarse:

- ¿ Quiénes mantienen a la empresa, los jefes o los clientes ?

La respuesta siempre era obvia, por lo tanto, decidieron invertir el enfoque de trabajo: "del cliente hacia el interior de la empresa". Al hacer este cambio, los resultados comenzaron a ser impresionantes.

Como este cambio los llevaba a invertir los procedimientos, se les ocurrió bautizarlo con el nombre de:

"Ingeniería Inversa" y después "Reingeniería de Procesos"

- 2. CONCEPTO DE REINGENIERIA:** Es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras espectaculares en cuanto a:
 - Reducir costos
 - Mejorar calidad
 - Mejorar servicio
 - Aumentar rapidez de cambio y de respuesta a las necesidades del mercado.

¿ POR QUÉ REVISION FUNDAMENTAL ?

- Porque debemos hacernos preguntas básicas, tales como:
 - ¿ Por qué hacemos lo que estamos haciendo ?, ¿ por qué ?
 - ¿ Qué tan eficaces son las normas, reglas y supuestos sobre los que se basa la administración de nuestra empresa ?
 - ¿ No habrán otras reglas y supuestos más eficaces ?
 - ¿ Qué actividades cuestan más de lo que aportan ?
 - ¿ Qué actividades impiden satisfacer al cliente ?

¿ POR QUÉ REDISEÑO RADICAL ?

Porque se trata de responder a las siguientes preguntas:

- ¿ Qué pasa si eliminamos o reducimos los procedimientos existentes e inventamos nuevas maneras de hacer el trabajo ?

¿ POR QUÉ MEJORAS ESPECTACULARES ?

- Porque se trata de dar saltos gigantescos en rendimiento y no hacer mejoras graduales. Estas se pueden lograr con programas de Calidad Total.

¿ POR QUÉ UN PROCESO ?

- Porque se trata de rediseñar un conjunto de actividades que reciben uno o más insumos y crean un resultado de valor para el cliente.

Por consiguiente, según la Reingeniería, conviene eliminar actividades que no crean valor, tales como:

Controlar, supervisar, revisar, autorizar, dar órdenes, dar indicaciones, evaluar y seleccionar proveedores, rehacer, mover, almacenar, esperar, apilar, descargar, levantar, empujar, devolver, etc.

Además, bajo el paradigma de Adam Smith, el dividir el trabajo en sus tareas más simples, y asignar cada una de ellas a un especialista, hace que haya muchos especialistas concentrados en tareas individuales del proceso, perdiendo de vista el objetivo principal: el satisfacer al cliente.

- Por consiguiente, el paradigma de Smith propicia el no satisfacer al cliente en todos sus requerimientos, por lo que no es eficaz para la nueva era.

3. RAZON DE SER DE LA REINGENIERIA

La Reingeniería surge:

- . Por apertura y globalización de la economía.
- . Por avances impresionantes de Japón en el mundo occidental, con armas de alto poder, como la Calidad Total y el Justo a Tiempo.
- . Por competencia cada vez más intensa.
- . Por clientes cada vez más exigentes en cuanto a calidad, variedad, buen servicio, buen precio.
- . Por tendencias democratizadoras que provocan en los trabajadores un mayor deseo de ser tomados en cuenta.
- . Por necesidad de que las empresas sean cada vez más ágiles, flexibles, competitivas, enfocadas al cliente y rentables.

Según Michael Hammer, Reingeniería significa "empezar de cero" y esto requiere:

- Empezar sin ninguna lógica previa.
- Preguntándose lo que es estrictamente necesario, para dejar de lado todo aquello que no lo es.
- Dejar de lado gran parte de los conocimientos acumulados durante los últimos doscientos años en materia de Administración de Empresas e Ingeniería Industrial.
- Dejar de ver a los trabajadores como simples empleados cuya obligación es cumplir órdenes, y comenzar a verlos como seres pensantes, con potencial creativo y como socios de la empresa.
- Desarrollar a los trabajadores para que encuentren nuevas formas de hacer mejor el trabajo.

4. BENEFICIOS DE LA REINGENIERIA:

- Rapidez, agilidad, flexibilidad, versatilidad, menos costos, menos precios, mayor competitividad.
- Mayor satisfacción del cliente.
- Mayor lealtad de los clientes.
- Mayor clientela por recomendaciones de clientes satisfechos.
- Mayor prestigio, mayor participación en el mercado.
- Organizaciones planas y livianas.
- Jefes no, facilitadores, entrenadores, asesores, líderes, si.
- Más equipo, menos individualismo.
- Más educación y desarrollo, además de capacitación y adiestramiento.
- Más libertad con responsabilidad.
- Más satisfacción y bienestar para los trabajadores.

- Reducción de desperdicios.
- Menos devoluciones, menos quejas, menos reparaciones.

5. COSTOS DE LA REINGENIERIA:

- Cambios en la planta física.
- Traslados de personal y su equipo.
- Reeducación y capacitación del personal.
- Salarios del personal capacitado y más responsabilizado.
- Sistemas de computación.
- Adaptación o reposición de equipos.

6. RETOS A VENCER:

- Cambiar paradigmas.
- Cambiar enfoque: de trabajar para los jefes a trabajar para los clientes.
- Vencer resistencia al cambio de las unidades de trabajo: de departamentos funcionales a equipos de procesos.
- Aceptar el cambio de organizaciones jerárquicas a planas.
- Vencer la actitud hacia apoyarse en bases de poder.
- Hacer que los gerentes y supervisores cambien a líderes.
- Superar el principio de la división del trabajo.
- No limitarse a su especialidad ni a su tarea.
- Aceptar el cambio de papel del trabajador: de controlado a facultado.
- Concientizar trabajadores.

TEMA 2: VISION EN LOS PROCESOS. EJEMPLOS Y EJERCICIOS DE IDENTIFICACION DE PROCESOS Y DE TIPIFICACION DE PASOS

¿ QUE ES UN PROCESO ?

Es un conjunto de actividades que transforman insumos en resultados de valor para el usuario/cliente (interno o externo).

Los insumos pueden ser:

- . Personas
- . Materiales
- . Equipo
- . Información
- . Tiempo
- . Dinero

Los resultados pueden ser:

- . Una resolución
- . Un producto terminado
- . Una visa concedida
- . Un permiso de construcción
- . Acuerdos derivados de la solución de un conflicto
- . Una forma llena
- . Una obra terminada

Los procesos pueden ser:

- . Proporcionar un servicio
- . Elaborar un producto
- . Procesar una solicitud de visa

- . Tramitar un permiso
- . Manejar un conflicto
- . Llenar una forma
- . Construir una obra

Ejemplos de procesos comunes:

- . Un paciente recibe tratamiento en un hospital.
- . Una póliza de seguros es procesada.
- . Se publica un periódico.
- . Se procesa un trámite de adquisición de vivienda.
- . Una pareja es atendida en un restaurante mientras cena.
- . Se fabrica un componente de computadora.
- . Se repara una máquina.
- . Se da servicio de agua potable a una comunidad.
- . Se atiende un conflicto de tenencia de la tierra.
- . Se estudian varias propuestas de construcción de una obra.
- . Se elabora un programa de trabajo.
- . Se sacan fotocopias.

Ejercicio:

. Considere usted un proceso de trabajo en el que esté involucrado, y después responda a las siguientes preguntas:

1. ¿ Qué nombre le daría usted al proceso ?
.....
2. ¿ Cuáles son algunos insumos del proceso ?
.....
3. ¿ Cuáles son los resultados ?
.....

4. ¿ Representan los resultados un servicio, un producto, la conclusión de una tarea ó alguna combinación de lo anterior ?

.....

5. ¿ Quién recibe los resultados ? ¿ quiénes son los clientes ?

.....

6. ¿ Son los clientes internos o externos ? ¿ o ambos ?

.....

7. ¿ Quiénes son algunos de los proveedores de insumos ?

.....

RELACION ENTRE PROCESO Y FUNCIONES DE UNA ORGANIZACION

La mayoría de las instituciones y empresas están organizadas en unidades o líneas departamentales o funcionales. Por ejemplo, una organización típica tiene un departamento de personal, uno de finanzas, uno de servicios, uno de capacitación, uno de difusión, etc.

La organización en departamentos o funciones separadas crea una jerarquía funcional. Sin embargo, los procesos no saben de jerarquías funcionales. Atraviesan los límites de departamentos y funciones para entregar un resultado al usuario.

Los procesos son horizontales y las organizaciones están formadas por funciones verticales.

Las funciones en sí están separadas y los procesos se encargan de interconectarlas.

En esta interconexión que hace el proceso con las funciones se observan muchos problemas, tales como pugnas internas, mala comunicación, competencia entre áreas y mala coordinación. Asimismo, provoca situaciones en las que nadie parece tener el control. Todos poseen parte del pastel, pero nadie es dueño del total.

Para evitar tal confusión, muchas empresas comienzan a organizarse en función a los procesos. Están aprendiendo a administrarse en forma multidisciplinaria.

Cuando las empresas se organizan por procesos, empiezan a ocurrir cosas buenas, tales como: mejoran la comunicación, la coordinación y la calidad. Además, las actividades se hacen mas rápido y en forma más barata.

TIPOS DE PASOS DE UN PROCESO

Existen seis pasos básicos de un proceso:

- 1) Operación
- 2) Transporte
- 3) Inspección
- 4) Demora
- 5) Almacenaje
- 6) Retrabajo

La operación (O) es un tipo de actividad que modifica la situación inicial. Hace avanzar el proceso hacia el resultado que espera el cliente. Por lo tanto, agrega valor al proceso.

El transporte (T) es cualquier acción que desplaza información, objetos o personas.

Demora (D): retraso de materiales, partes o productos y cualquier tiempo de espera de las personas:

Inspección (I): incluye inspecciones de calidad y cantidad, revisiones y autorizaciones.

Almacenaje (A): retraso programado de materiales, partes o productos.

Retrabajo (R): cualquier paso de repetición o corrección evitable.

Ejercicio:

Indicar el tipo correcto de paso (O, T, D, I, A, R):

1. Buscar información : _____
2. Ensamblar dos componentes: _____
3. Repetir un paso en un proceso: _____
4. Mover materiales: _____
5. Revisar un informe: _____
6. Esperar el inicio de una reunión: _____
7. Registrar datos por segunda vez: _____
8. Caminar hacia la camioneta de servicio: _____
9. Enviar información por fax: _____
10. Guardar material en un depósito: _____
11. Captar los datos una sola vez en su origen: _____
12. Efectuar una inspección de control de calidad: _____
13. Esperar por un listado de computadora: _____
14. Revisar y autorizar una solicitud: _____
15. Atender una llamada telefónica: _____
16. Repetir una carta para corregir un error: _____
17. Revisar un trabajo elaborado por otra persona: _____
18. Dejar un formato en una charola: _____
19. Llenar forma de requisición: _____
20. Elaborar una factura: _____
21. Se formula un programa: _____
22. Se somete un programa a la consideración de los superiores: _____

TEMA 3: DETERMINACION DE LA EFICIENCIA Y DEFICIENCIA DE PROCESOS DE TRABAJO

INTRODUCCION

El tiempo que dedicamos a un trabajo puede dividirse en dos componentes: trabajo y desperdicio. El objetivo de la Reingeniería de Procesos es eliminar todo el desperdicio existente en un proceso de trabajo.

Ahora, ¿ qué significa la palabra "trabajo" ?.

Si lo buscamos en un diccionario, se encuentra que "trabajo" se refiere a:

"Esfuerzo o actividad física o mental que se dirige hacia la producción o logro de algo". Con base en esta definición, sólo es posible lograr una mayor productividad a partir de un mayor esfuerzo físico o mental, es decir, trabajando más duro, pero no necesariamente en forma más inteligente.

En el contexto de la Reingeniería de Procesos, la palabra trabajo tiene un significado diferente. Se utilizará esta palabra sólo cuando una determinada actividad desplace un proceso hacia adelante o, lo que es lo mismo, le agregue valor en forma directa.

Por ejemplo, en el proceso de tramitar una solicitud, el hecho de "analizar la solicitud", "tomar una decisión" y "responder al solicitante" representan actividades que agregan valor al proceso. Sin embargo, si alguien tiene que "esperar a que otro analice la solicitud" para pasarla o otro a que tome la decisión, o este último tiene que esperar a que otro la analice, estas "esperas" no agregan valor al proceso. Al contrario, le agregan demoras y costos. No agregan valor porque impiden un avance **rápido** al trámite (tal como lo desea el solicitante).

Asimismo, si la solicitud tiene que estarse llevando y trayendo para diferentes fines, estas actividades de "lleva y trae" tampoco agregan valor al proceso. Al contrario, le agregan esfuerzos, demoras y costos.

Igualmente, si la solicitud tiene que pasarse a otro formato o a otro departamento u oficina para procesarla o responderla, o tiene que dejarse esperando en una charola hasta que otro la recoja para seguirla procesando, o hay que esperar a que alguien firme su resolución, o hay que dirigirse nuevamente al solicitante porque hubo un error en su llenado, o hay que hacer alguna corrección o un retrabajo durante el trámite porque hubo un error en su lectura o en la transcripción de algún dato, todas estas actividades no agregan valor al proceso. Al contrario, estarán agregando nuevamente esfuerzos, materiales, movimientos, demoras, costos y hasta molestias.

Entonces en la Reingeniería de Procesos, el trabajo agrega valor y el desperdicio agrega demoras y costos.

Así, el desperdicio representa las actividades que no agregan valor al proceso. Incluye el esfuerzo, tiempo, materiales, movimientos y costos que se desperdician.

Ahora bien, por lo general el trabajo y el desperdicio requieren la misma cantidad de esfuerzo físico. En el ejemplo del trámite de la solicitud, el pasarla a otro formato, requiere la misma cantidad de trabajo físico que el llenarla por primera vez, o el tener que dirigirse nuevamente al solicitante para corregir un error, requiere la misma cantidad de trabajo (o más) que si se revisa exitosamente la solicitud al momento de llenarla, para que, si hay algún error, detectarlo en el momento y corregirlo.

¿ COMO IDENTIFICAR TRABAJO Y DESPERDICIO ?

Como se observó en el ejemplo precedente, el desperdicio aparece en muchas formas: demoras, transportes, inspecciones, retrabajos, etc. Todas éstas son actividades que "no agregan valor" al proceso, sino demoras y costos.

Para identificar el desperdicio, nos podemos hacer las siguientes preguntas:

- 1) Si se elimina o reduce al mínimo esta actividad en particular del proceso, ¿ se afectará la calidad del rendimiento o del resultado del proceso ?
- 2) Como usuario, ¿ Desea usted pagar por esta actividad en particular ? ¿ Le es de valor para usted ?.

Si la respuesta es no, tal vez la actividad sea un desperdicio. Es necesario eliminar el desperdicio del proceso o al menos, reducirlo al mínimo.

Ejercicio:

Colocar una T o una D a cada una de las siguientes actividades según sean Trabajo o Desperdicio:

1. Llevar información: _____
2. Ensamblar dos componentes: _____
3. Repetir un paso en un proceso: _____
4. Detener un trabajo por esperar que otro firme: _____
5. Transcribir información: _____
6. Corregir datos ya registrados: _____
7. Revisar un trabajo hecho por otro: _____
8. Revisar un trabajo hecho por uno mismo: _____
9. Dar órdenes: _____
10. Esperar a que comience una reunión: _____
11. Pasar datos: _____
12. Ir a buscar información: _____
13. Almacenar materiales en un depósito: _____
14. Capturar los datos una sola vez en su origen: _____
15. Realizar cualquier retrabajo: _____

¿ QUE ES EFICIENCIA Y DEFICIENCIA DEL TRABAJO ?

En forma ideal, todos los procesos contienen sólo trabajo y cero desperdicio. De manera realista, eso es difícil de alcanzar. En vez de eso, es preciso aumentar al máximo el trabajo y reducir al mínimo el desperdicio en el proceso. La eficiencia del trabajo es un indicador de qué tanto valor agrega ese trabajo, y la deficiencia es un indicador de qué tanto valor desagrega ese trabajo o de qué tanto es su desperdicio.

La Eficiencia del trabajo se calcula dividiendo la cantidad de trabajo entre la suma del trabajo y el desperdicio de un proceso.

Matemáticamente, se expresa de la siguiente manera:

$$E = (\text{TRABAJO} / \text{TRABAJO} + \text{DESPERDICIO}) \times 100 \%$$

La Deficiencia del trabajo se calcula dividiendo la cantidad de desperdicio entre la suma del trabajo y el desperdicio de un proceso.

Matemáticamente, se expresa de la siguiente manera:

$$D = (\text{DESPERDICIO} / \text{TRABAJO} + \text{DESPERDICIO}) \times 100 \%$$

Igualmente, la Deficiencia de un proceso puede calcularse de la siguiente manera:

$$D = 100 \% - E$$

siendo E la Eficiencia del proceso.

Uno de los objetivos fundamentales de la Reingeniería es hacer que los procesos tengan un 100% de Eficiencia, o bien un 0% de Deficiencia.

Una forma práctica de calcular la Eficiencia y Deficiencia de un proceso, consiste en expresar el Trabajo y el Desperdicio en una unidad de Tiempo, tal como: minutos, horas, días, semanas, meses, etc.

EJERCICIO:

Consideremos el proceso: **REQUISITAR**

A continuación se describe este proceso con un total de 10 actividades, especificándose al lado la duración promedio de cada una de ellas en minutos.

Se pide indicar con una letra entre paréntesis (O, D, T, I, R, A) el tipo de actividad de que se trata, y luego en un segundo paréntesis indicar si se trata de Trabajo (T) o Desperdicio (D). Por último, se pide calcular la Eficiencia (E) y Deficiencia (D) del proceso.

PROCESO: REQUISITAR

Nº	ACTIVIDAD
1.	Inicio de llenado de la forma de requisición (10 min.) () ()
2.	Envío de la forma a abastecimientos (720 min.) () ()
3.	La forma permanece en una charola (75 min.) () ()
4.	Terminación de la forma de requisición (18 min.) () ()
5.	La forma permanece en una charola (75 min.) () ()
6.	Envío de la forma a autorización (720 min.) () ()
7.	La forma permanece en una charola (45 min.) () ()
8.	Revisión y autorización de la forma (12 min.) () ()
9.	La forma permanece en una charola (90 min.) () ()
10.	Envío de la forma a pedidos (720 min.) () ()

TRABAJO =

DESPERDICIO =

E =

D =

IDENTIFICACION DE DEFICIENCIAS DEL PROCESO: REQUISITAR

ACTIVID/ PASO #	DESCRIPCION	TIPO	VALOR AGREGADO (-3 A 3)	TIEMPO (MIN.)	DEFICIENCIAS
1	INICIO DE FORMA	O	2	10	NO
2	ENVIO DE FORMA A ABASTECIM.	T	- 3	720	SI
3	LA FORMA EN UNA CHAROLA	A	- 1	75	SI
4	TERMINACION DE LA FORMA	O	1	18	NO
5	LA FORMA EN UNA CHAROLA	A	- 1	75	SI
6	ENVIO A AUTORIZACION	T	- 3	720	SI
7	LA FORMA EN UNA CHAROLA	A	- 1	45	SI
8	SE REVISA Y AUTORIZA	I	- 1	12	SI
9	LA FORMA EN UNA CHAROLA	A	- 1	90	SI
10	ENVIO A COMPRAS	T	- 3	720	SI
TOTAL			- 1.10	2485	98.8 %

TEMA 4 : DISEÑO DE OPCIONES DE INNOVACION APLICANDO LOS PRINCIPIOS BASICOS DE LA REINGENIERIA

INTRODUCCION

Una vez definidas las deficiencias del proceso en estudio, así como los objetivos del Plan Estratégico de Reingeniería, el Agente de Cambio procede a generar Opciones de Innovación del Proceso bajo estudio.

Para generar estas opciones, hacemos uso de algunos principios básicos que propone la Reingeniería, como son:

1. Eliminar el desperdicio.
2. Reducir el desperdicio al mínimo.
3. Simplificar: pensar siempre en hacerlo sencillo, no complejo.
4. Cada vez que sea posible, combinar pasos del proceso.
5. Diseñar procesos con rutas alternas.
6. Pensar en paralelo, no en línea.
7. Recabar los datos en su origen.
8. Usar la tecnología para mejorar el proceso.
9. Dejar que los clientes y proveedores ayuden en el proceso.

GUIA PRACTICA PARA EL DISEÑO TECNICO DE INNOVACIONES

PRINCIPIO BASICO 1: ELIMINAR EL DESPERDICIO

1) Identificar rendimientos que espera el cliente/usuario:
rapidez, bajo costo y/o calidad

2) Identificar posibles pasos inhibidores de los rendimientos:
demoras, transportes, inspecciones, retrabajos, almacenajes o incluso operaciones.

3) Preguntar: ¿ se afectan negativamente los rendimientos esperados si se elimina cada uno de los pasos antes identificados ?

- Si la respuesta es "si", pasar al principio 2 (reducir el desperdicio)
- Si la respuesta es "no", pasar a 4).

4) Preguntar:

a) ¿ Por qué se está haciendo el trabajo siguiendo estos pasos ?

- Si la respuesta fuera algo así como: "porque si no, se dejarían pasar muchos errores, o porque así se ha hecho siempre, etc."

Volver a preguntar: ¿ si se pudieran corregir los errores en el momento de producirlos, podríamos evitar pasos subsecuentes ?

O bien: ¿Cuál puede ser una mejor forma de hacer este trabajo con menos pasos ?

b) ¿ Por qué es necesario que una persona inicie la forma y otra la concluya ?

Respuesta probable: "porque la primera persona no cuenta con todos los datos necesarios para llenarla totalmente".

Volver a preguntar: ¿ es posible que esta persona pueda contar con todos estos datos, para que de esa manera pueda llenar la forma ella sola ?.

c) ¿ Por qué es necesario que otra persona revise y autorice la solicitud ? ¿ qué se podría hacer para que ella misma haga la revisión y autorización ?

PRINCIPIO BASICO 2: REDUCIR EL DESPERDICIO AL MINIMO

Cuando resulta difícil eliminar el desperdicio, proceder a reducirlo.

Preguntar: ¿ Cómo se podría obtener el mismo resultado haciéndolo en menos pasos ?

- Si la respuesta es "no se puede", volver a preguntar:

" ¿ Y no se podría hacer consumiendo menos tiempo ? "

Respuesta probable: "Tal vez". Volver a preguntar: "¿ Cómo podría hacerse ?".

Respuesta probable: "A través del fax" (para el caso de transporte) o "A través de la computadora" (para el caso de búsqueda de datos) o "A través de un facsímil" (para el caso de las firmas).

PRINCIPIO BASICO 3: SIMPLIFICAR EL PROCESO

A través de:

a) Preguntarse sobre la posibilidad de reducir la cantidad de insumos o de casos a procesar, con tal de no afectar negativamente los rendimientos esperados del proceso.

identificar insumos o casos no estrictamente necesarios, proceder a eliminarlos, simplificando así el proceso.

b) Preguntarse si todos los requisitos que se exigen actualmente son estrictamente los necesarios para cumplir con los rendimientos esperados del proceso.

Detectar requisitos no necesarios, proceder a eliminarlos, simplificando así el proceso.

c) identificar los pasos básicos del proceso, separando todos aquellos que surgieron para darle "calidad", y preguntarse si estos últimos se pueden eliminar o reducir a otros más rápidos, o más económicos o menos complicados.

PRINCIPIO BASICO 4: COMBINAR PASOS DEL PROCESO

a) Combinar un paso de inspección con otro de operación, para eliminar pasos de transportes, demoras y retrabajos.

Para ello, preguntar:

"¿ Cómo se podrían detectar y corregir errores en el momento en que ocurran en lugar de hacerlo varios pasos después ?"

b) Combinar un paso de demora o uno de transporte con otro de operación.

Para ello, preguntar:

"¿ Qué operación se puede hacer mientras se espera o transporta algo ?"

c) Combinar dos pasos de operación.

Para ello, preguntar:

"¿ Qué operación se puede hacer mientras se hace otra ?"

PRINCIPIO BASICO 5: DISEÑAR PROCESOS CON RUTAS ALTERNAS

Preguntar:

¿ El proceso o algunos de sus pasos está diseñado para la excepción o para la regla ?

En caso de que mayormente sea para la excepción, crear rutas alternas a partir de puntos de decisión, dados por la siguiente declaración:

"Si algo es cierto, entonces se hace algo. Si eso mismo es falso, entonces se hace otra cosa"

PRINCIPIO BASICO 6: PENSAR EN PARALELO, NO EN LINEA

Preguntar:

- 1) ¿ Por qué algunos pasos no pueden realizarse en paralelo ?
- 2) ¿ Por qué no es posible reducir en forma importante los tiempos de ciclo?
- 3) ¿ Cómo podemos colocar los pasos en paralelo sin afectar negativamente el valor agregado?

PRINCIPIO BASICO 7: RECABAR LOS DATOS EN SU ORIGEN

Cada vez que se observe que una información es transcrita o recabada más de una vez, preguntar:

- ¿ De qué manera es posible evitar la transcripción o la recabación de información más de una vez ?

Para la respuesta, pensar en el uso de la computadora.

PRINCIPIO BASICO 8: USAR LA TECNOLOGIA PARA MEJORAR EL PROCESO

Preguntar:

- ¿ En qué pasos del proceso podemos usar computadora, fax, teléfonos celulares, correo electrónico, internet u otros medios que puedan mejorar la eficiencia y eficacia del proceso ?

PRINCIPIO BASICO 9: DEJAR QUE LOS CLIENTES Y PROVEEDORES AYUDEN EL EL PROCESO

Preguntar:

- ¿ De qué manera es posible involucrar al cliente o al proveedor en el mejoramiento del proceso, aumentando beneficios para ambos ?

PRINCIPALES CARACTERISTICAS DE LOS PROCESOS SOMETIDOS A REINGENIERIA

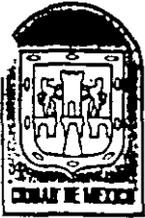
Cuando un proceso es sometido a Reingeniería, presenta las siguientes características:

- a) **En una sola tarea se realizan varias de las anteriores**
Desaparece el trabajo en serie, y varias tareas que antes eran distintas, ahora se integran y comprimen horizontalmente en una sola. Y a las personas que ejecutan esa única "tarea", se les llama "trabajadores o grupos de caso", los cuales actúan como únicos puntos de contacto con el cliente del proceso.
- b) **Los trabajadores toman decisiones**
En aquellos puntos en los que los trabajadores tenían que acudir a sus superiores jerárquicos, hoy pueden tomar sus propias decisiones. De modo que el proceso no sólo es comprimido horizontalmente (en cuanto a tareas), sino también verticalmente porque se reducen los tramos de control.
- c) **Los pasos del proceso se ejecutan en orden natural**
En los procesos rediseñados, el trabajo es secuenciado en función de lo que es necesario hacerse antes o después. A ésto se le llama "deslinearización", y con ello se logra que: a) Muchas tareas se hacen simultáneamente, b) Los tiempos de ciclo disminuyen y c) Hay menos repetición de trabajo.
- d) **Los procesos tienen múltiples versiones**
En virtud de que los procesos son diseñados en forma simple, siguen rutas paralelas de tal manera que cada una de ellas está en sintonía con los requisitos de un determinado tipo de mercado, situación o insumo específico, de modo que el proceso puede atender a múltiples tipos de clientes y situaciones con economías de escalas equivalentes a la producción masiva.
- e) **El trabajo se realiza en el sitio razonable**
El trabajo se desplaza a través de fronteras organizacionales para mejorar el desempeño global del proceso, contándose con la participación de clientes y/o proveedores en la realización de ese trabajo. Se integran partes del trabajo relacionadas entre sí y realizadas por unidades independientes.
- f) **Son mínimas las verificaciones y los controles**
En lugar de verificar estrictamente el trabajo, los procesos rediseñados muchas veces tienen controles globales o diferidos, que toleran "abusos moderados", demorando el punto en que el "abuso" se detecta o examina en patrones colectivos en lugar de casos individuales, logrando fuertes reducciones de costo y de tiempos, que compensan con creces cualquier posible aumento de abusos que se derive de esta "mayor confianza".
- g) **Son mínimas las conciliaciones**
Se disminuye el número de puntos de contacto externo que tiene un proceso, y con ello se reducen las probabilidades de que se reciba información incompatible que requiera conciliación. Por ejemplo, una orden de compra

puede no estar de acuerdo con el documento de recibo o con la factura, y éstos pueden no estar de acuerdo entre sí. Al eliminar la factura, los puntos de contacto externo se reducen de 3 a 2, y la posibilidad de desacuerdo en dos tercios, además que el Departamento de Cuentas por Cobrar se reduce espectacularmente.

- h) Los trabajos se realizan en procesos y no en departamentos funcionales**
Varios departamentos se transforman en uno o varios "Equipos de proceso". Los jefes dejan de actuar como supervisores y se comportan ahora como entrenadores, asesores y líderes. Los trabajadores piensan más en las necesidades de los clientes del proceso y menos en las de sus jefes. Actitudes y valores cambian en respuesta a nuevos incentivos. Los equipos de proceso no incluyen representantes de todos los departamentos funcionales a los que reemplaza, sino que sustituyen a la vieja estructura departamental.
- i) Los oficios de los trabajadores cambian a multidimensionales**
Aunque los trabajadores de equipos de procesos que son responsables colectivamente de los resultados del proceso, más que individualmente responsables de una tarea, no realizan todos el mismo trabajo (al fin y al cabo, todos tienen distintas habilidades y capacidades), la línea divisoria de ellos se desdibuja. Todos los miembros del equipo tienen al menos algún conocimiento básico de todos los pasos del proceso, y probablemente realiza cada uno varios pasos, pero siempre con una apreciación del proceso en forma global. Por ejemplo, un ingeniero puede reparar computadoras, comprar las refacciones, hacer la factura y cobrar, cuando antes lo único que hacía era reparar computadoras. Al ser multidimensional, el trabajo es mejor pagado y permite un desarrollo personal basado en el mayor aprendizaje y no tanto en escalar posiciones de mayor jerarquía. Antes era: "Tareas sencillas para gente sencilla", ahora es: "Oficios complejos para gente capacitada".
- j) El papel del trabajador cambia: de controlado a facultado**
Los trabajadores hacen sus propias reglas y toman sus propias decisiones. Son personas a las que se les permite que piensen, se comuniquen y obren con su propio criterio. Deciden cómo y cuándo se ha de hacer el trabajo, dentro de los límites de sus obligaciones para con la organización, fechas límite convenidas, metas de productividad, normas de calidad, etc. La contratación de trabajadores se hace no únicamente en base a sus estudios y experiencia, sino también por su carácter: si tiene iniciativa, autodisciplina, orientación a los clientes, etc.
- k) La preparación para el oficio no sólo es entrenamiento, sino básicamente educación**
Al trabajador se le educa para discernir qué es lo que debe hacer. El entrenamiento se enfoca únicamente al "cómo" y la educación aumenta la perspicacia y la comprensión del "por qué" y "para qué", que son elementos fundamentales para que pueda estar en capacidad de tomar decisiones acertadas.

**TEMA 5: APLICACION DE LA REINGENIERIA DE
PROCESOS EN DIFERENTES AREAS**



Departamento del
Distrito Federal

División de
Educación
Continua



DIPLOMADO DE REINGENIERIA DE PROCESOS

MODULO I: Reingeniería Básica de Procesos

CASO: Soporte Técnico Operativo a Usuarios de
Sistemas Informáticos en la Oficialía Mayor

INTEGRANTES:

Apodaca Pérez Roberto
Fernández Gutiérrez Armando
Fierro Torres Eduardo
Guzmán Mejía Rafael
Mandujano Soto Francisco
Molina Vergara Jesús

MANUAL DE REINGENIERÍA DE PEQUEÑOS PROCESOS

INSTITUCION O EMPRESA : Departamento del Distrito Federal

PROCESO A INNOVAR : Soporte Técnico Operativo a usuarios de Sistemas informáticos en la Oficialía Mayor.

EQUIPO DEL PROYECTO : Armando Fernández Gutiérrez.
Francisco Mandujano Soto.
Eduardo Fierro Torres.
Rafael Guzmán Mejía.
Jesús Molina Vergara.
Roberto Apodaca Pérez.

RAZONES PARA LA INNOVACIÓN :

- Agilizar tiempos de respuesta.
- Ampliar la atención a los usuarios.
- Mejorar los servicios prestados.
- Ser elementos activos y no reactivos.
- Estandarizar las soluciones a problemáticas afines.

ACTORES INVOLUCRADOS EN LA INNOVACIÓN :

- Usuarios de equipo de cómputo en la O.M.
- Levantadores de órdenes de servicio.
- Analistas de problemática.
- Técnico de primer nivel.
- Técnico de segundo nivel.
- Proveedores externos de servicios.

IDENTIFICACION DE DEFICIENCIAS DEL PROCESO ACTUAL

ACTIVIDAD Nº	DESCRIPCION	TIPO	AGREGA VALOR	TIEMPO (min)	DEFICIENCIAS
1	Solicitud del Soporte	O	3	60	NO(T)
2	Llenado de forma de atención	O	3	10	NO(T)
3	Envío al analista de la problemática	T	-1	15	SI(D)
4	Asignación del trabajo al técnico	I	-2	240	SI(D)
5	Programación de atención al usuario	D	-3	240	SI(D)
6	Traslado a la oficina del usuario	T	-1	10	SI(D)
7	Planteamiento de la problemática	R	2	10	SI(D)
8	Análisis de la posible solución	O	3	15	NO(T)
9	Reasignación del soporte	R	-3	20	SI(D)
10	Aplicación de la solución	O	3	30	NO(T)
11	Comprobación del servicio	I	1	30	SI(D)
12	Firma de aceptación del servicio	O	0	5	NO(T)
13	Cierre de la orden de trabajo	O	-1	10	NO(T)
TOTALES			+4	635	

TRABAJO = 10 + 15 + 30 + 5 + 10 = 70

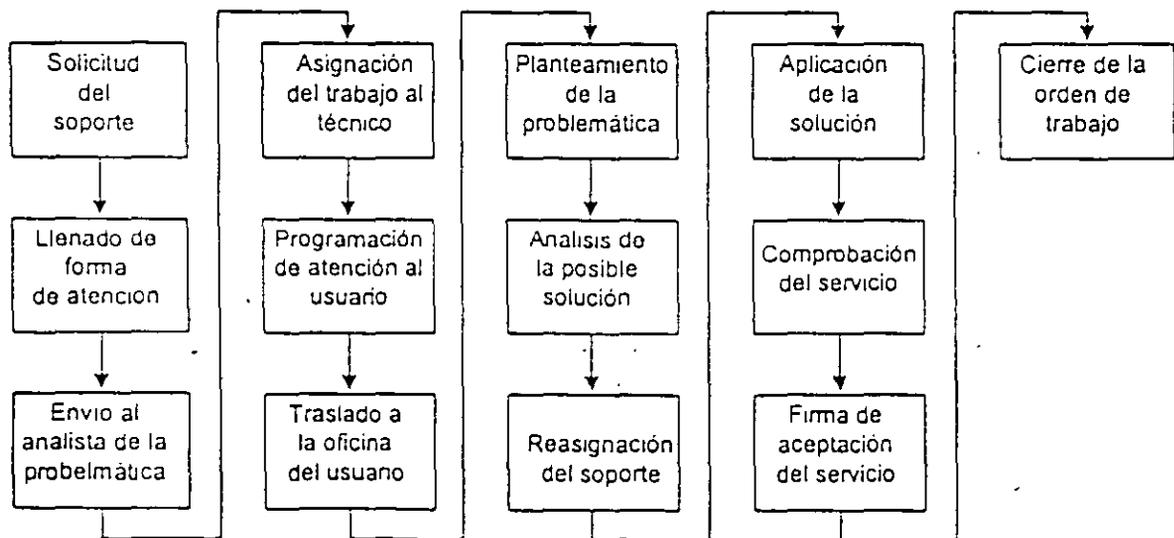
DESPERDICIO = 15 + 240 + 240 + 10 + 10 + 20 + 30 = 565

EFICIENCIA = $(70 / (70 + 565)) * 100 = 11.02$

DEFICIENCIA = $(565 / (565 + 70)) * 100 = 88.98$

Valor Agregado = $+4 / 13 = 0.3076 = 31 \%$

FLUJO DE ACTIVIDADES :



PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

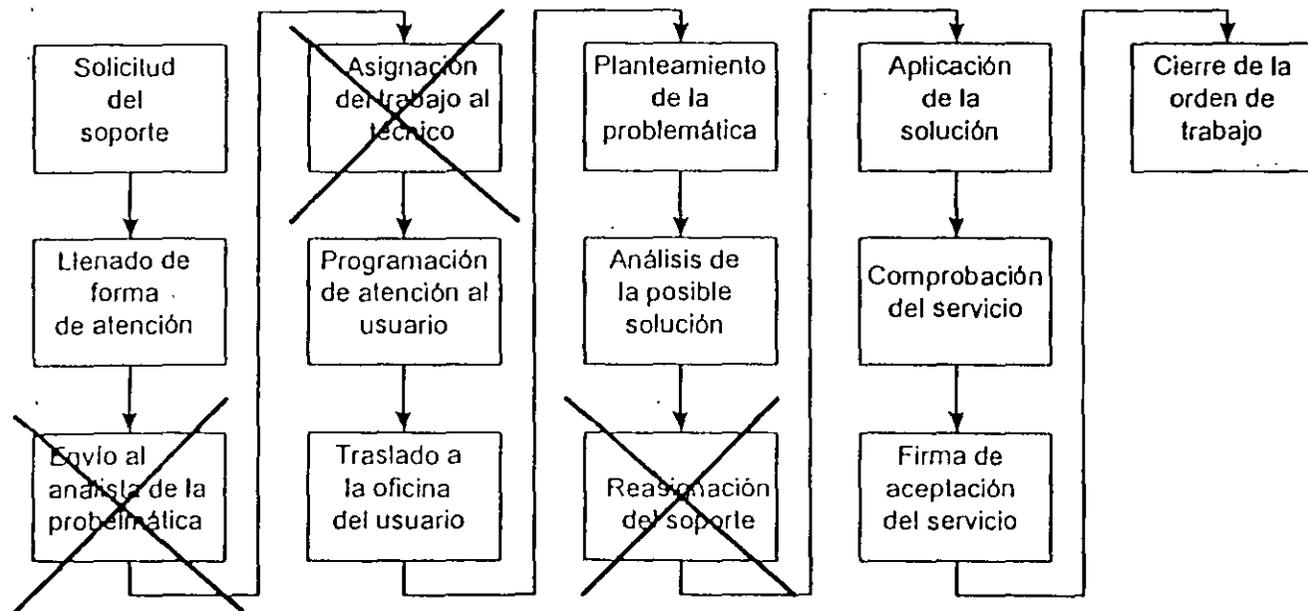
APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 1. ELIMINAR EL DESPERDICIO

Al aplicar el principio se propone eliminar 3 actividades :

3. Envío al analista de la problemática.
4. Asignación del trabajo al técnico.
9. Reasignación del soporte.

Los técnicos revisarán directamente los reportes, identificando aquellos que caigan bajo su ámbito de acción y llevarán los elementos necesarios para resolver la problemática.



PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 2. REDUCIR EL DESPERDICIO -NO ELIMINADO- AL MÍNIMO

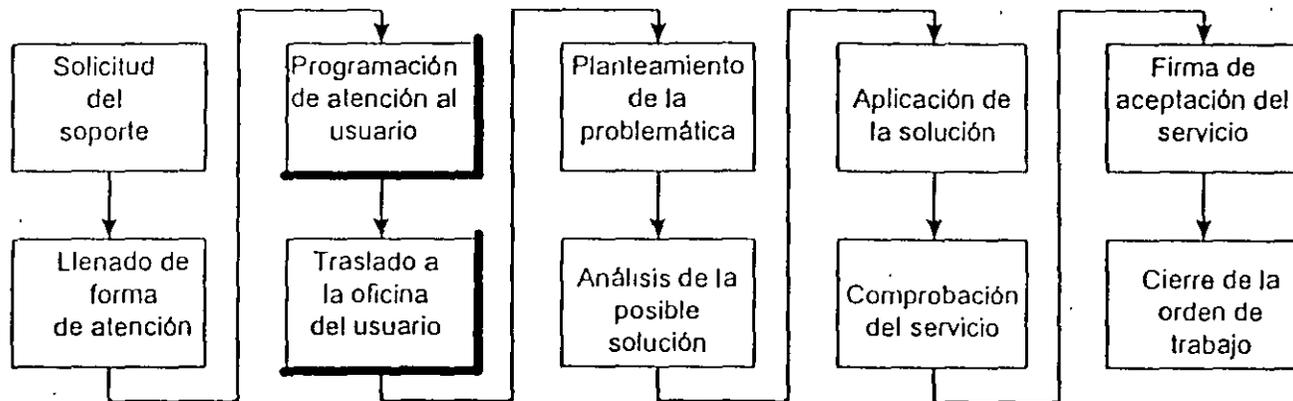
Se analizan las actividades existentes, buscando reducir los tiempos y las tareas que las forman.
Se propone minimizar el desperdicio de las siguientes actividades:

5. Programación de atención al usuario.

- Eliminación de trabajos a asignar, dando solución en línea a los más sencillos.
- Capacitando a los usuarios, con lo que se mejora la operación y se reducen los problemas operativos.
- Se atiende a los usuarios en función de la problemática.

6. Traslado a la oficina del usuario.

- Se avisa a los técnicos que están en un área cercana a la del nuevo servicio para evitar que regrese a la oficina y tenga que trasladarse nuevamente al área.

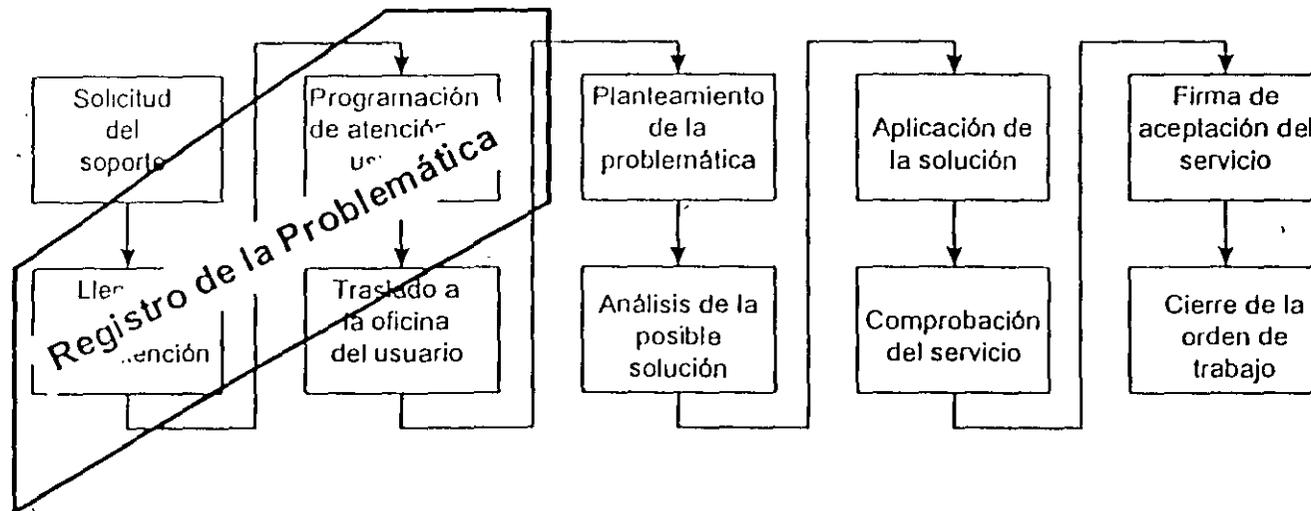


PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 3. SIMPLIFICAR

De las tareas actuales del proceso, se observa que se pueden simplificar la 2. Llenado de la forma de atención y la 5. Programación de atención al usuario en una sola que se denomine "Registro de la Problemática".



82

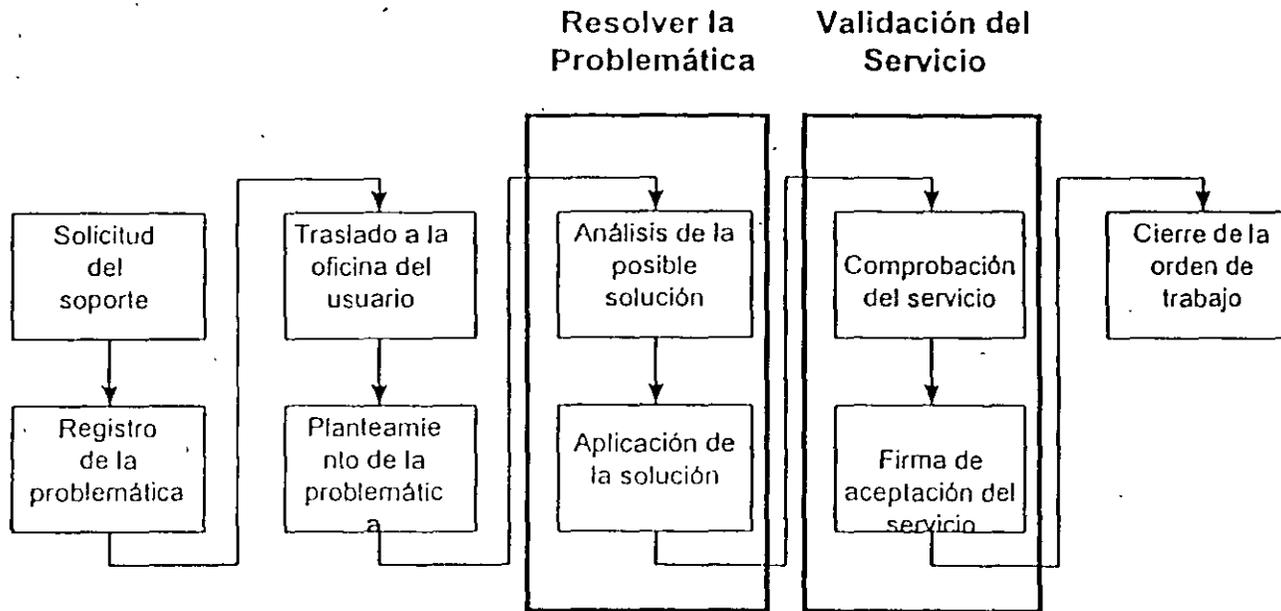
PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 4. COMBINAR PASOS DEL PROCESO.

Existen todavía actividades que pueden combinarse con miras a agilizar el proceso. Las seleccionadas para efectuar esta combinación son:

1. "Resolver la problemática" es la nueva tarea surgida de la combinación de las actividades 8. Análisis de la posible solución y 10. Aplicación de la solución.
2. "Validación del servicio" es el resultado de la combinación de las actividades 11. Comprobación del servicio y la 12. Firma de aceptación del servicio.

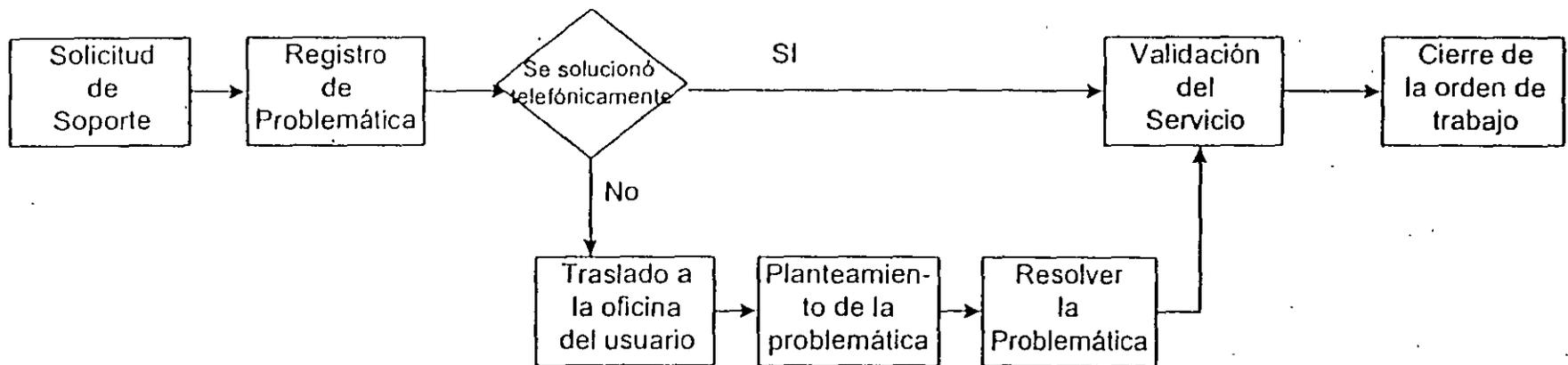


PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 5. DISEÑAR PROCESOS CON RUTAS ALTERNAS.

Con las actividades existentes, se genera este nuevo proceso:



PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

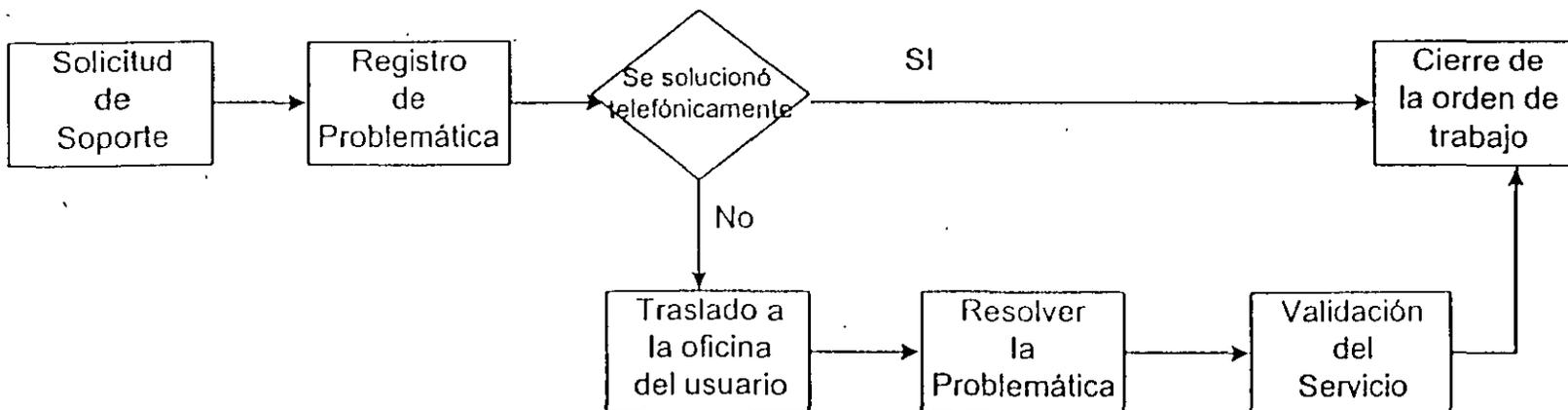
PRINCIPIO BÁSICO 6. PENSAR EN PARALELO, NO EN LÍNEA

Debido a las distintas mejoras generadas con los primeros cinco principios, la opción de actividades en paralelo ya ha sido considerada, por lo que no se plantea una nueva innovación en este punto.

31

PRINCIPIO BÁSICO 7. RECABAR LOS DATOS EN SU ORIGEN.

Al recabar la información de la problemática desde su origen se elimina la necesidad de Plantear la Problemática nuevamente.



PROCESO A INNOVAR: Soporte Técnico operativo a usuarios de sistemas Informáticos en la Oficialía Mayor.

APLICACIÓN DE LOS PRINCIPIOS BÁSICOS DE LA REINGENIERÍA AL PROCESO.

PRINCIPIO BÁSICO 8. USAR LA TECNOLOGIA PARA MEJORAR EL PROCESO

Como parte de las innovaciones consideradas en los puntos anteriores, se mencionó la inclusión en el proceso de una aplicación de cómputo para el levantamiento de las solicitudes de servicio, que al mismo tiempo le sirva al levantador como guía de la entrevista al usuario.

PRINCIPIO BÁSICO 9. DEJAR QUE LOS CLIENTES Y PROVEEDORES PARTICIPEN EN EL PROCESO

En relación a los "clientes", la principal razón del área de servicio es que los usuarios no tienen en su mayoría una cultura informática que les permita ser elementos activos en el proceso. Se puede prever para una segunda etapa, toda vez que reciban una capacitación mayor, el modificar la aplicación del levantamiento para que sean ellos los que la operen y puedan por sí mismos resolver algunas de las problemáticas presentadas, siguiendo las instrucciones que se les den a través de esta aplicación.

Por tener que filtrar todos aquellos servicios que no les competen a los proveedores, su participación se reduce a la necesidad precisa de sus técnicos en problemas específicos.

IDENTIFICACION DE DEFICIENCIAS DEL PROCESO INNOVADO

ACTIVIDAD N°	DESCRIPCION	TIPO	AGREGA VALOR	TIEMPO (min)	DEFICIENCIAS
1	Solicitud del Soporte	O	3	60	NO(T)
2	Registro de la Problemática	O	3	10	NO(T)
3	Traslado a la oficina del usuario	T	-3	6	SI(D)
4	Resolver la Problemática	O	3	30	SI(D)
5	Validación del Servicio	I	0	15	NO(T)
6	Cierre de la orden de trabajo	O	-1	10	SI(D)
	TOTALES		+5	131	

TRABAJO = 60 + 10 + 30 + 10 = 110

DESPERDICIO = 6 + 15 = 21

EFICIENCIA = $(110 / (110 + 21)) * 100 = 83.97$

DEFICIENCIA = $(21 / (21 + 110)) * 100 = 16.03$

Valor Agregado = $+5 / 6 = 0.83333$ = 83 %

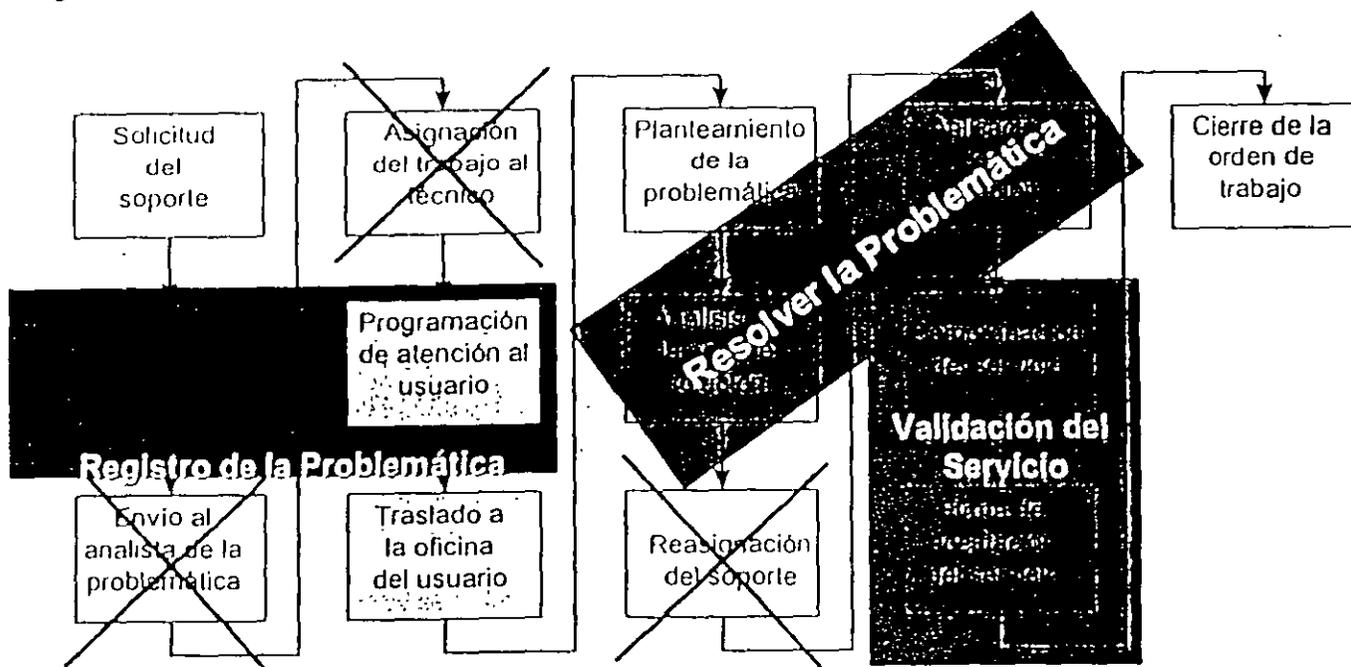
COMPARACION ENTRE AMBAS OPCIONES:

	PROCESO ACTUAL	PROCESO INNOVADO	DIFERENCIAS
TRABAJO	70	110	+ 40
DESPERDICIO	565	21	- 544
EFICIENCIA	11.02%	83.97%	+ 72.95%
DEFICIENCIA	88.98%	16.03%	- 72.95%
VALOR AGREGADO	30.76	83.33%	+ 52.57%

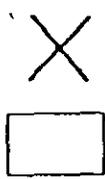
El Proceso Innovado consume únicamente el 20.63% del tiempo original.

Proceso: Soporte Técnico Operativo a usuarios de Sistemas Informáticos en la Oficialía Mayor.

Mejoras al proceso utilizando los principios de la Reingeniería



34



Eliminar el desperdicio

Reducir el desperdicio no eliminado al mínimo

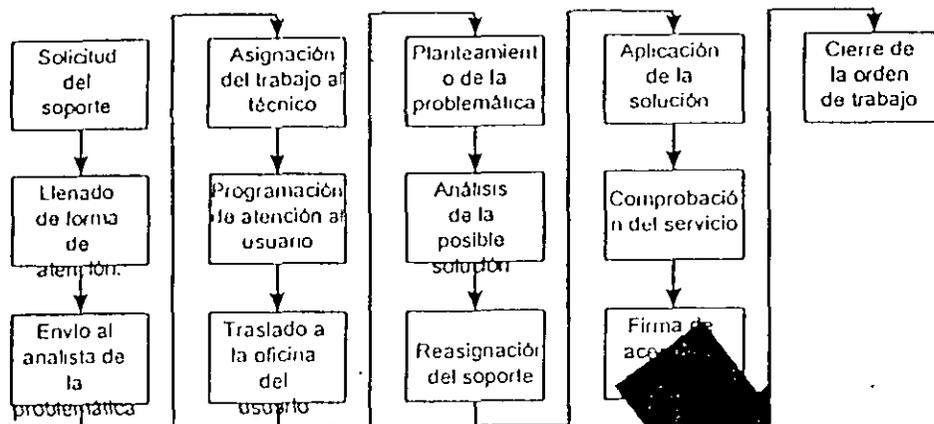


Simplificar

Combinar pasos del Proceso

Proceso: Soporte Técnico Operativo a usuarios de Sistemas Informáticos en la Oficialía Mayor.

Comparación Proceso Original vs Proceso Innovado

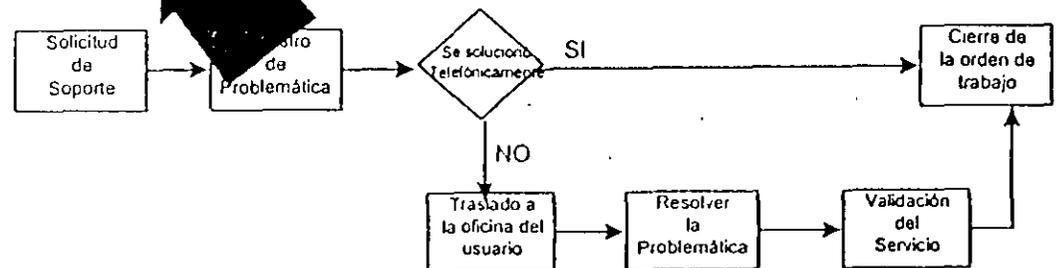


PROCESO ACTUAL

- Consta de 13 pasos
- Se tarda 635 min., promedio
- El 11.02% es Eficiencia
- El 88.98% es Deficiencia

PROCESO INNOVADO

- Consta de 6 pasos
- Se tarda 131 min. promedio
- El 83.97% es Eficiencia
- El 16.03% es Deficiencia



Proceso: Soporte Técnico Operativo a usuarios de Sistemas Informáticos en la Oficina Mayor,

Información Numérica

IDENTIFICACION DE DEFICIENCIAS DEL PROCESO ACTUAL HOJA N°. 1

IDENTIFICACION DE DEFICIENCIAS DEL PROCESO INNOVADO HOJA N°. 1

ACTIVIDAD N°.	DESCRIPCION	TIPO	AGREGA VALOR	TIEMPO (min)	DEFICIENCIAS
1	Solicitud del Soporte	O	3	60	NO(T)
2	Llenado de forma de atención	O	3	10	NO(T)
3	Envío al analista de la problemática	T	-1	15	SI(D)
4	Asignación del trabajo al técnico	I	-2	240	SI(D)
5	Programación de atención al usuario	D	-3	240	SI(D)
6	Traslado a la oficina del usuario	T	-1	10	SI(D)
7	Planteamiento de la problemática	R	2	10	SI(D)
8	Análisis de la posible solución	O	3	15	NO(T)
9	Reasignación del soporte	R	-3	20	SI(D)
10	Aplicación de la solución	O	3	30	NO(T)
11	Comprobación del servicio	I	1	30	SI(D)
12	Firma de aceptación del servicio	O	0	5	NO(T)
13	Cierre de la orden de trabajo	O	-1	10	NO(T)
TOTALES			+4	635	

ACTIVIDAD N°.	DESCRIPCION	TIPO	AGREGA VALOR	TIEMPO (min)	DEFICIENCIAS
1	Solicitud del Soporte	O	3	60	NO(T)
2	Registro de la Problemática	O	3	10	NO(T)
3	Traslado a la oficina del usuario	T	-3	6	SI(D)
4	Resolver la Problemática	O	3	30	SI(D)
5	Validación del Servicio	I	0	15	NO(T)
6	Cierre de la orden de trabajo	O	-1	10	SI(D)
TOTALES			+5	131	

TRABAJO = 60 + 10 + 30 + 10 = 110
 DESPERDICIO = 6 + 15 = 21

EFICIENCIA = $(110 / (110 + 21)) * 100 = 83.97$
 DEFICIENCIA = $(21 / (21 + 110)) * 100 = 16.03$

TRABAJO = 10 + 15 + 30 + 5 + 10 = 70
 DESPERDICIO = 15 + 240 + 240 + 10 + 10 + 20 + 30 = 565
 EFICIENCIA = $(70 / (70 + 565)) * 100 = 11.02$
 DEFICIENCIA = $(565 / (565 + 70)) * 100 = 88.98$

COMPARACION ENTRE LAS OPCIONES:			
	PROCESO ACTUAL	PROCESO INNOVADO	DIFERENCIAS
TRABAJO	70	110	+40
DESPERDICIO	565	21	-544
EFICIENCIA	11.02%	83.97%	+72.95%
DEFICIENCIA	88.98%	16.03%	-72.95%

El Proceso Innovado es un 72.95% más eficiente que el Proceso Actual.

MANUAL DE REINGENIERIA DE PEQUEÑOS PROCESOS

INSTITUCION O EMPRESA:

PROCESO A INNOVAR:

EQUIPO DEL PROYECTO:
.....
.....
.....
.....
.....
.....

RAZONES PARA LA INNOVACION:
.....
.....
.....
.....
.....

ACTORES INVOLUCRADOS EN LA INNOVACION:
.....
.....
.....
.....
.....
.....
.....

TEMA 6: EVALUACION Y SELECCION DE OPCIONES DE INNOVACION Y ESTRATEGIAS PARA VIABILIZAR EL ÉXITO DE IMPLANTACION DE ESAS INNOVACIONES

INTRODUCCION

Una vez realizado el Rediseño del Proceso, pasamos a la etapa de implantación.

Para ello, se hace necesario considerar diversas opciones del proceso innovado, a efectos de evaluarlas para seleccionar aquella opción de mayor conveniencia financiera y factibilidad humana:

Estas opciones son básicamente las siguientes:

- 1) El proceso innovado totalmente (opción básica I).
- 2) El proceso innovado parcialmente (opciones II, III, ...).

Si implantamos el proceso innovado totalmente, estaremos aplicando la Reingeniería en toda su extensión (opción de cambio radical), y muy probablemente obtendríamos con ello el mayor beneficio financiero, pero esta opción puede no ser humanamente factible en el momento de querer hacer la implantación.

Por su parte, si implantamos el proceso innovado parcialmente, estaremos aplicando la Reingeniería a un nivel moderado o superficial (opción de cambio moderado), lo cual puede no ser la mejor opción desde un punto de vista financiero, pero sí desde un punto de vista humano.

En consecuencia, podemos establecer que cuando en la organización no existe un convencimiento pleno ni una voluntad total a favor del cambio radical por parte de todos los actores involucrados en él, deberán plantearse y evaluarse varias opciones de innovación del proceso, para realizar una selección de la más adecuada.

En el caso contrario, el Agente de Cambio puede enfocarse únicamente a evaluar la conveniencia y viabilidad del proceso innovado totalmente (opción básica). En cualquiera de los dos casos, la evaluación se realiza aplicando 2 criterios básicos:

- 1) Relación Beneficio/Costo (B/C)
- 2) Balance de Apoyos y Rechazos (BAR).

Relación Beneficio/Costo

Para aplicar este criterio, se procede a estimar los ahorros a obtenerse con el nuevo proceso rediseñado, calculados en tiempo y estimados en costos. Estos ahorros constituirán los beneficios de la innovación, Asimismo, se procede a estimar los costos necesarios para implantar el proceso rediseñado, con base en un cálculo aproximado de costos de sus requerimientos.

La división entre los beneficios y costos así estimados nos dará un indicador de la relación beneficio/costo de cada opción.

Cuando esta relación es menor que 1, se considera que la opción no es conveniente. Cuando resulta entre 1 y 2, se considera poco conveniente, entre 2 y 3 conveniente, y mayor de 3 muy conveniente. Se sugiere seleccionar aquellas opciones con valores B/C mayores de 3.

Balance de Apoyos y Rechazos

Para aplicar este segundo criterio, se procede a consultar o simular el punto de vista y posición de cada actor sobre qué tanto apoyaría o rechazaría la implantación del proceso rediseñado en cada una de sus opciones.

Para ello se utiliza una escala convencional de -3 a 3, siendo el "-3" Rechazo Total, el "+3" Apoyo Total, el "0" neutralidad, el +2 Mediano apoyo, el +1 Poco apoyo, el -2 Mediano rechazo y el -1 Poco rechazo.

Colocados estos valores, se divide la suma total de los positivos entre la suma total de los negativos, siendo el resultado de estas divisiones el Balance de Apoyos y Rechazos (BAR). Si este Balance es mayor o igual que 2, la opción es humanamente viable, y será inviable si el Balance resulta menor de 2.

Selección

Una vez hechas las evaluaciones de los indicadores B/C y BAR, se procede a ponderarlos a través de asignarles un peso o una ponderación a cada uno, utilizando valores porcentuales o decimales, de modo que los dos ponderadores o pesos deben sumar 100 ó 1, según el caso. Por ejemplo, si le asignamos un 30% a B/C y un 70% a BAR, tendríamos que multiplicar a B/C

por 0.3 y a BAR por 0.7, sumamos los resultados de estas dos multiplicaciones, y la suma total nos debe dar mayor de "3" para que podamos concluir que "Se acepta" la innovación propuesta, de lo contrario "No se acepta". Cuando son varias opciones de innovación las que se proponen, se seleccionará la que, resultando mayor de "3", sea la de máximo valor en comparación con las demás.

Suele ser conveniente que las propuestas de innovación sean calificadas de "Radicales" o "moderadas, según el grado en que logran eliminar o reducir los desperdicios o deficiencias del proceso. Son radicales aquellas innovaciones que eliminan o reducen el desperdicio en más del 60%, y moderadas las que lo eliminan o reducen entre un 30 y un 60%. Menos de ese porcentaje correspondería a innovaciones "Ligeras" o "Superficiales".

Recomendaciones

Las recomendaciones que se pueden hacer se refieren a acciones a tomar para mejorar los valores de los indicadores de las innovaciones que resultaron mejor calificadas. Estas acciones se refieren, por ejemplo a "reducción de costos de implantación de las innovaciones" (para mejorar la relación B/C) y/o "gestionar apoyos de actores rechazantes o neutrales y/o de bajo apoyo (para aumentar el balance BAR).ç

Estrategias para viabilizar el éxito de las innovaciones

En la(s) opción(es) que resulte(n) seleccionada(s) podemos tener actores que estén rechazando o que no estén dando un apoyo total, y habrá casos en que inclusive alguna(s) de las opciones no sea(n) viable(s).

En cualquiera de los casos anteriores, se hace necesario proceder a gestionar los apoyos de esos actores para llevarlos a un nivel de Apoyo Total, para tener la certeza de que el esfuerzo del cambio no se va a ver impedido o limitado para llegar al éxito de la Reingeniería del proceso en estudio.

Al gestionar la obtención o el incremento de los apoyos, estaremos aumentando la viabilidad del éxito de las innovaciones.

Para ello nos valemos de Negociación, Sensibilización, Persuasión, etc. En el caso de nuestra cultura, se ha observado que lo más efectivo es aplicar Estrategias y Tácticas de Negociación

bajo la Filosofía de Ganar-Ganar, mediante la cual exploramos intereses de los actores que podrían inclinar la balanza en favor del cambio, ayudando a esos actores a que satisfagan sus intereses, los cuales deben ser totalmente compatibles con los Paradigmas de la Reingeniería.

Las estrategias de negociación se formulan a través de intercambios de intereses por apoyos, y a estas Estrategias las llamamos Viabilizantes del Cambio.

Tácticas para viabilizar Estrategias

Una vez formuladas las Estrategias para viabilizar el éxito de las innovaciones, se procede a ponerlas en práctica, y seguidamente se lleva a cabo la implantación del Proyecto de Reingeniería.

Para poner en práctica las Estrategias de Viabilización, se requiere formular Tácticas que a su vez actúen como Viabilizantes de esas Estrategias.

Las Tácticas son las acciones con las que se "aterrizan" las Estrategias.

Estas acciones se realizan en los siguientes pasos:

1. Preparación de las condiciones necesarias para iniciar el proceso de negociación.
2. Creación de interés en los actores por las propuestas de negociación.
3. Manejo de objeciones
4. Cierre de acuerdos.

PASO 1: Preparación de las condiciones necesarias para iniciar el proceso de negociación.

a) Reunir información demostrativa y/o testimonial de los beneficios que obtendrá el Actor Objetivo (AO) con la aplicación de la propuesta de negociación.

b) Aumentar poder de negociación del Agente de Cambio (AC), a través de:
. Acumular apoyos de otros actores y personas más accesibles.

- . Minimizar debilidades y mejorar fortalezas del AC, valoradas por el AO.
- . Preparar presentación del Programa/Proyecto en un lenguaje claro, según la preferencia del AO.
- . Preparar opciones de Estrategias Viabilizantes.
- . Darse holgura de tiempo para el proceso de negociación.

c) Investigar intereses del AO, bien sea con gente allegada y/o con el propio AO informalmente.

PASO 2: Creación de interés del AO.

a) Actuar en el sentido de los intereses del AO.

b) Aprovechar encuentros informales para preguntar al AO su opinión sobre el contenido de las EVs. Por ejemplo: "¿ Qué le parecería si lográramos tales cosas (que a él le interesan), aunque haya que sacrificar algunas otras, tales como.... (las que a nosotros nos interesan pero a él no) ?"

c) Desarrollar una Visión atractiva y estimulante en el sentido de "sus" intereses y de la Institución, incluyendo como necesario el apoyo del AO en lo que deseamos.

Paso 3: Manejar objeciones.

Cuando alguna de las respuestas del AO es "No", proceder a manejar esta objeción para convertirla en un "Sí", a través de:

a) Concederle la razón temporalmente (esto es ético, profesional, riguroso y conveniente).

b) Averiguar sutilmente la razón de su objeción, diciendo por ejemplo: "Sí, efectivamente tiene usted razón porque muchas veces sucede que cuando a la gente le damos tal cosa (la que el AO no quiere ceder), ella tiende a hacer esta otra cosa (algo indeseable para el AO), ¿ es por eso que usted lo dice o por alguna otra razón ?". Con ello lo que buscamos es que el AO nos dé "su" razón de fondo (aquella que lo lleva a presentar su objeción).

c) Proponerle una solución en el sentido de la EV o de alguna de sus opciones,

diciéndole por ejemplo:

"¿ Si fuera posible evitar que la gente hiciera tal cosa (su razón de fondo) al darle esta otra cosa (la que se está pidiendo a AO), usted estaría de acuerdo en dársela, o pediría usted alguna otra cosa ?"

Ante ello, el AO podría responder: "Sí estaría de acuerdo, pero es algo muy difícil lograr que la gente no haga tal cosa".

Responderle: "No se preocupe, yo me ocupo de que eso no suceda. Lo que me interesa saber es si usted apoyaría esto": "Sí".

Paso 4: Cerrar acuerdos

Esto se puede hacer con varias opciones:

- a) Pedir a AO su opinión sobre cómo instrumentar los acuerdos alcanzados.
- b) Proponerle alguna forma de instrumentarlos (la que sabemos que él preferiría)
- c) Proponerle escribir los acuerdos.
- d) Preguntarle cómo le parecería si se logra la Visión, enfatizando tanto en sus intereses como en los de los demás actores y de la Institución.
- e) Si aún no está muy convencido, invitarlo a probar (no se pierde nada).

Los formatos que se llenan en este tema se denominan:

"Evaluación y selección de opciones de innovación"

"Intereses y necesidades de los actores involucrados en la implantación de la opción seleccionada"

"Estrategias para viabilizar el éxito de las innovaciones"

"Tácticas para viabilizar estrategias"

EVALUACION Y SELECCION DE OPCIONES DE INNOVACION

PROCESO:

CRITERIOS DE EVALUACION	O P C I O N E S			
	I:	II:	III:	IV:
BENEFICIOS (B)				
COSTOS (C)				
RELACION B/C				
ACTORES	APOYOS Y RECHAZOS A LAS OPCIONES (-3 A +3)			
A1				
A2				
A3				
A4				
A5				
A6				
SUBTOTALES				
BALANCE BAR (+/-)				
PONDERACIONES				
TOTAL				
SELECCION				
RECOMENDACION: _____				

ESTRATEGIAS PARA VIABILIZAR EL EXITO DE LAS INNOVACIONES

OPCION SELECCIONADA:

INNOVACIONES PARCIALES E INTERESES NEGOCIABLES	APOYOS Y RECHAZOS						ESTRATEGIAS VIABILIZANTES
	A1	A2	A3	A4	A5	A6	
1.							EV1:
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							
11.							
12.							
13.							
14.							
15.							

TACTICAS PARA VIABILIZAR ESTRATEGIAS:

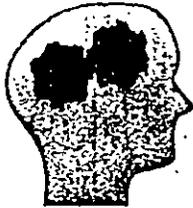
- 1) PREPARAR LAS CONDICIONES INICIALES PARA LA NEGOCIACION
- 2) DESPERTAR INTERES POR LAS PROPUESTAS
- 3) MANEJAR OBJECIONES
- 4) CERRAR ACUERDOS

ESTRATEGIAS A SER VIABILIZADAS	TACTICAS VIABILIZANTES
EV1	1) 2) 3) 4)
EV2	1) 2) 3) 4)
EV3	1) 2) 3) 4)
EV4	1) 2) 3) 4)

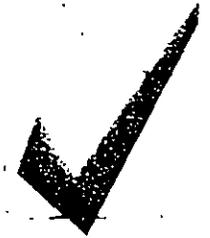
PROCESO A INNOVAR : Otorgamiento de Préstamos Escolares

EVALUACION Y SELECCIÓN DE OPCIONES DE INNOVACION

Criterios De Evaluación	OPCIONES		
	I	II	III
BENEFICIOS	314.100		
COSTOS	150.000		
RELACION B/C	2.094		
ACTORES			
POBLACION DE DECHO HABIENTE	3		
PERSONAL ADMINISTRATIVO	3		
SINDICATO	-3		
DIR. DE PRESTACIONES	2		
SERVS. DE COMPUTO	1		
TECNICOS ANALISTAS	1		
SUB. DE ADMINSTRACION	3		
SUBTOTALES	+13 -3		
BALANCEAR(+ -)	43		
PONDERACIONES	.5(1.04) .5(215)		
TOTAL	3.57		
	>>3 Se acepta		



49



EVALUACIÓN Y SELECCIÓN DE OPCIONES DE INNOVACION.

Proceso: Soporte Técnico Operativo a usuarios de Sistemas informáticos en la
Oficialía Mayor.

Criterios de Evaluación	Opciones			
	I: Unica Radical	II	III	IV
Beneficios (B)	\$310,464.00			
Costos ©	\$27,520.00			
Relación B/C	11.28			
Actores:				
Usuarios	3			
Levantadores	-2			
Analistas	-2			
Técnico 1er N	2			
Técnico 2° N	2			
Proveedor Ext	1			
Subtotales	+8	-4		
Balance E-R (+/-)	+2			
Ponderaciones	.3 (3.38)	0.7 (1.4)		
TOTAL	4.78			
Selección	>>3 Se acepta			

RECOMENDACIÓN:

Motivar a los involucrados cuyas actividades fueron absorbidas por otras, para lograr su cooperación.

Principalmente con capacitación se les podría apoyar para que realicen las nuevas actividades



CIUDAD DE MÉXICO

DIPLOMADO EN REINGENIERIA PUBLICA

EVALUACION Y SELECCION DE OPCIONES DE INNOVACION

PROCESO: Soporte Técnico Operativo de usuarios en el otorgamiento de licencias de uso de suelo

CRITERIOS DE EVALUACION	OPCIONES			
	I:UNICA RADICAL	II	III	IV
Beneficios (B)	\$1,705.600.00			
Costos ©	\$98,880.00			
Relación B/C	17.07			

ACTORES

Ciudadano	3			
Receptor Analista	3			
Subdirector	-3			
Mensajero	-1			
Receptor Seduvi	-3			
Supervisor	-3			
Subdirector	3			
Subtetales	+9	-10		
Balance BAR (+/-)	-0.9			

Ponderaciones	3 (17.07)=5.12			
	0.7(-0.9)=0.63			
TOTAL	5.75			
Selección	>>3 Se acepta			

RECOMENDACION:

Gestionar apoyos de los actores que quedaron relegados del proceso original a fin de aumentar la viabilidad de la innovación

ESTRATEGIAS PARA LOGRAR O AUMENTAR LA VIABILIDAD DEL CAMBIO

ACCIONES DE CAMBIO E INTERESES DE LOS ACTORES	ACTORES CLAVE, SUS APOYOS Y RECHAZOS										ESTRATEGIAS VIABILIZANTES (EVn)
	A	B	C	D	E	F	G	H	I	J	
1. FACULTAMIENTO Y HORIZONTALIZACION	SI	SI	NO	SI	NO	SI	SI	SI	NO	SI	EV1: A y B ofrecen apoyar a C en 2,6,7,8,9 por apoyos en 1,3,4,5
2. MULTIFUNCIONALIDAD Y EVALUACION POR RESULTADOS	SI	SI	SI	SI	NO	SI	SI	NO	NO	NO	EV2: A,B,C ofrecen apoyar a D y G en 1,2,3,4,5,6 por apoyo en 8
3. RESPONSABILIDAD POR PROCESOS	SI	SI	NO	SI	NO	SI	SI	SI	NO	NO	EV3: A,B,C,D,G ofrecen apoyar a F en 1,2,3,5,6,7,8,9 por apoyo en 4
4. IMPLANTACION DE SISTEMA DE MEJORAMIENTO CONTINUO	SI	SI	NO	SI	NO	NO	SI	SI	NO	SI	EV4: A,B,C,D,G,F ofrecen apoyar a H en 1,3,4,5,6 por apoyo en 2
5. EQUIPO, COOPERACION Y ALIANZAS	SI	SI	NO	SI	NO	SI	SI	SI	NO	SI	EV5: A,B,C,D,G,F,H ofrecen apoyar a J en 1,4,5,6,9 por apoyo en 2,3,7,8
6. CONTAR CON HERRAMIENTAS Y EQUIPOS ADECUADOS	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	EV6: A,B,C,D,G,F,H,J ofrecen apoyar a E,I en 6,7,8,9 por apoyo en 1,2,3,4,5
7. CONSOLIDAR INFORMACION	SI	SI	SI	SI	SI	SI	SI	SI	SI	NO	
8. MENORES COSTOS DE OPERACION Y MANTENIMIENTO	SI	SI	SI	NO	SI	SI	NO	SI	SI	NO	
9. MEJOR SERVICIO A USUARIOS	SI	SI	SI	SI	SI	SI	SI	SI	SI	SI	

CURSO: REINGENIERIA DE PROCESOS

BIBLIOGRAFIA RECOMENDADA

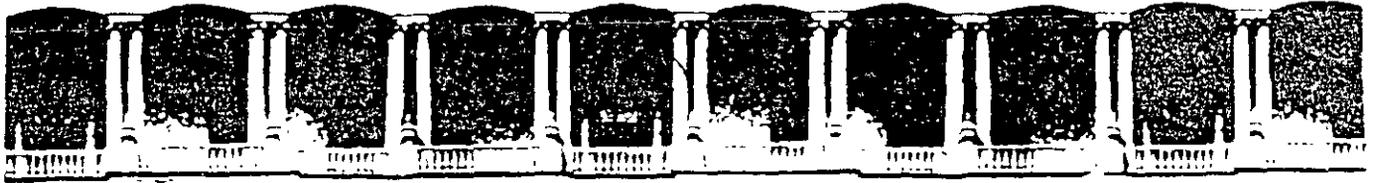
1. "COMO HACER REINGENIERIA"
AUTORES: RAYMOND MANGANELLI Y MARK KLEIN
EDITORIAL: NORMA

2. "REINGENIERIA"
AUTORES: MICHAEL HAMMER Y JAMES CHAMPY
EDITORIAL: NORMA

3. "MAS ALLA DE LA REINGENIERIA"
AUTOR: I.I.E. (INSTITUTO DE INGENIEROS INDUSTRIALES)
EDITORIAL: C.E.C.S.A.

4. "REINGENIERIA DE PROCESOS DE NEGOCIOS"
AUTORES: JOHANSON, McHUGH, PENDLEBURY Y WHEELER
EDITORIAL: LIMUSA

5. "REINGENIERIA"
AUTORES: DANIEL MORRIS Y JOEL BRANDON
EDITORIAL: MCGRAW HILL



**FACULTAD DE INGENIERIA U.N.A.M.
DIVISION DE EDUCACION CONTINUA**

CURSOS INSTITUCIONALES

**DIRECCION GENERAL DE SERVICIOS URBANOS DEL
GOBIERNO DEL DISTRITO FEDERAL**

REINGENIERÍA DE PROCESOS

Del 25 de Septiembre al 6 de Octubre del 2000

NOTAS COMPLEMENTARIAS

Ing. Alberto González Guisar
Delegación Iztacalco
Septiembre-Octubre/2000

Objetivos: Homogeneización

- Usted podrá.
 - Discutir el por que de la homogeneización
 - Determinar cuando deben combinarse dos clases, cuando debe dividirse una clase, cuando debe eliminarse una clase
 - Evaluar diagramas de clases y diagramas de interaccion para mantener la consistencia del modelo
 - Discutir las necesidades de reestructuración de paquete

1

¿Qué es Homogenización?

- Homogeneizar
"mezclar en un conjunto de elementos de manera uniforme, para homogeneizar"
Webster's New Collegiate Dictionary
- Entre más casos de uso y escenarios se desarrollen es necesario homogeneizar el modelo
 - Esto es especialmente cierto si existen multiples equipos que están trabajando en diferentes casos de uso

2

¿Qué buscar?

- Las clases se examinan para determinar si
 - Se pueden combinar dos o mas clases
 - Se debe dividir una clase
 - Se debe eliminar una clase completamente
- Se reestructuran los paquetes para resolver problemas de
 - Acoplamiento
 - Reutilizacion
 - Visibilidad

3

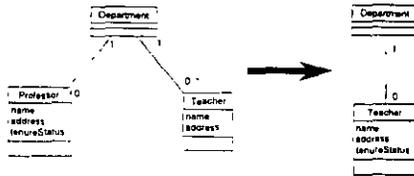
Combinación de Clases

- Cuando se realiza el análisis de casos de uso con varios equipos, cada uno puede asignar un nombre diferente a una misma clase
 - Esto es especialmente cierto ya que el lenguaje natural se usa para el análisis de casos de uso
- Debe conducirse el desarrollo del modelo
 - Evaluar definiciones de clase
 - Evaluar la estructura y comportamiento de las clases
 - Buscar sinónimos
 - Tomar el nombre mas cercano al dominio

4

Ejemplo: Combinación de Clases

- Se eligió a Teacher, ya que no todos los instructores de la Universidad han alcanzado el estatus de Professor



5

Ejemplo: Combinación de Clases

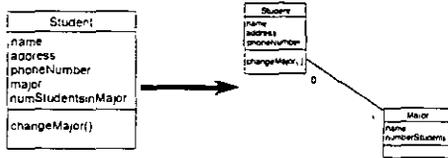
- Al colocar una clase de control por cada caso de uso, es necesario re-evaluar dichas clases
 - Las clases de control con comportamiento similar pueden combinarse
- Ejemplo: El Administrator interactúa con los casos de uso Maintain Student Information y Maintain Professor Information
 - Se crearon dos clases de control
 - La secuencia de acciones en cada una de estas dos clases de control es muy similar (revisar, crear, borrar información del actor)
 - Las clases de control pueden combinarse en una clase de control que se llame RegistrationUserManager

6

División de Clases

- Las clases con estructura y/o comportamiento que no sean cohesivos deben dividirse en clases diferentes

- Ejemplo:



7

Eliminación de Clases

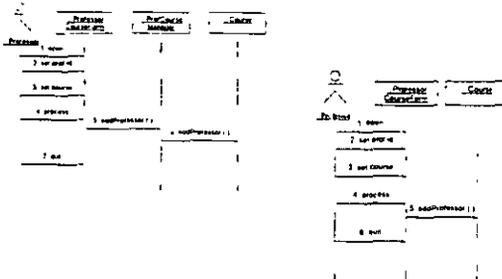
- Debe eliminarse por completo una clase del modelo
 - Que no tenga ninguna estructura o comportamiento
 - Que no participe en ningún caso de uso
- En particular, examine las clases de control
 - La falta de responsabilidad en la secuencia puede conducir a la eliminación de la clase de control

- Ejemplo:

- Un caso de uso para el actor Professor es "Seleccionar cursos para impartir"
 - Esto quiere decir que Professor introduce el nombre y número del curso y se crea una liga a la clase ProfessorInformation
 - Ya que no hay mucho comportamiento relacionado a la secuencia, se elimina esta clase de control

8

Diagrama de Secuencias Actualizado



9

Revisión de Consistencia

- Se debe revisar la consistencia de los Diagramas de Clases y los Diagramas de Interacción
 - ¿Se necesita al menos una operación para cada clase?
 - ¿Existe una clase para cada objeto en el Diagrama de Interacción?
 - Si un diagrama de objeto o diagrama de interacción muestra un mensaje que va de un objeto de la clase A a un objeto de la clase B, verifique:
 - Existe una operación en la clase B espera un evento y lo manipula
 - Existe una asociación correspondiente entre la clase A y la clase B en el diagrama de clases
 - El diagrama de transición de estado para la clase B (si se ha desarrollado uno) incluye ese evento

10

Reestructuración de Paquetes

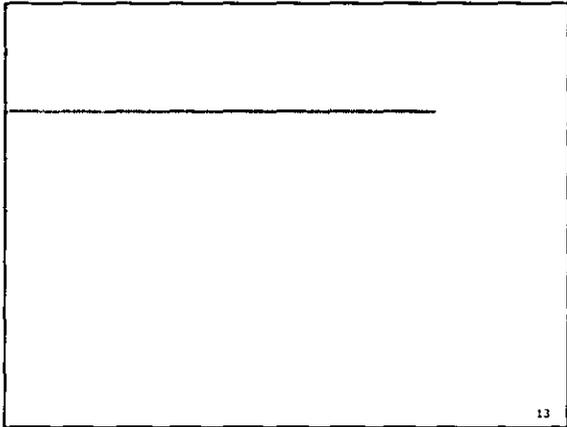
- Al desarrollar más casos de uso puede ser necesario reestructurar los paquetes del modelo
 - El fuerte acoplamiento entre paquetes puede significar que los paquetes deben combinarse
 - La dependencia recíproca entre paquetes puede significar que el paquete debe dividirse
 - Evalúe consideraciones de reuso
 - Un paquete reusable debe tener algunas dependencias
 - Evalúe las partes públicas y privadas de un paquete
 - No todas las clases en un paquete deben ser parte de la interfaz pública del paquete
 - Agregar clases boundary si es necesario encapsular la interfaz del paquete

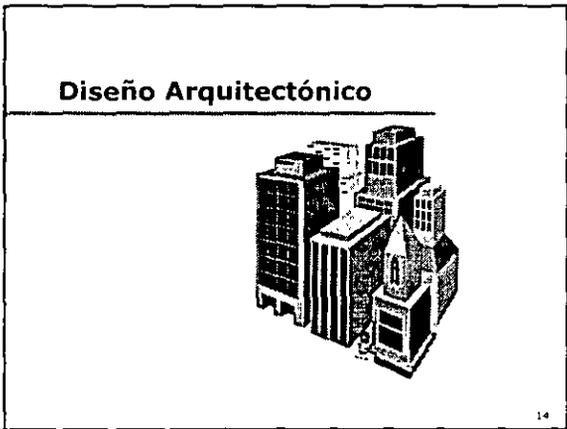
11

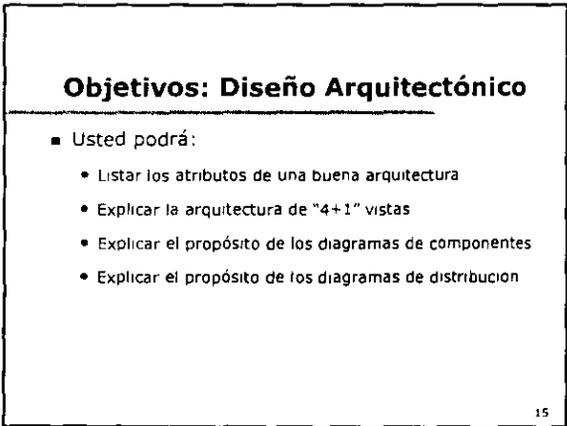
Ejercicio: Homogeneización

- Discutir las decisiones de homogeneización que se necesitan en este punto del análisis

12







Visión Arquitectónica

- Dos cualidades comunes para virtualmente todos los proyectos OO son.
 - La existencia de una visión arquitectónica
 - La aplicación de un ciclo de vida incremental bien-manejado e interactivo
- La arquitectura debe ser simple
 - El logro del comportamiento común a través de abstracciones y mecanismos comunes

16

Una Definición de Arquitectura del Software

"La arquitectura del software tiene que ver con la organización de sistemas de software, la selección de sus componentes, las interacciones entre estos componentes, la composición de componentes interactuando en subsistemas más grandes progresivamente y el total de los patrones que guían estas composiciones. Tiene que ver no sólo con la estructura de sistemas, sino también con su funcionalidad, desempeño, diseño, selección entre alternativas y comprensión"

Mary Shaw

17

Atributos de las Arquitecturas Buenas

- Las buenas arquitecturas se construyen en capas de abstracción bien definidas:
 - Cada capa representa una abstracción coherente
 - Cada capa tiene una interfaz bien definida y controlada
 - Cada capa se construye sobre facilidades bien definidas y controladas en niveles de abstracción bajos
- Hay una clara separación entre la interfaz y la implementación de cada capa
 - Los cambios en la implementación de una capa no viola la hipótesis hecha por sus clientes

18

Desarrollo de la Arquitectura del Sistema

- El diseño de la arquitectura tiene que ver con el manejo de riesgo
- Las buenas arquitecturas se determinan mejor a través de desarrollo iterativo e incremental
- A un experimentado equipo pequeño, guiado por un Arquitecto en Jefe, se le debe asignar la responsabilidad para:
 - Definir y mantener la integridad arquitectónica del sistema
 - Aprobar todos los cambios a la interfaz de paquetes
 - Valorar riesgos del proyecto
 - Proponer el orden y contenido de cada iteración

19

Dimensiones de la Arquitectura del Software

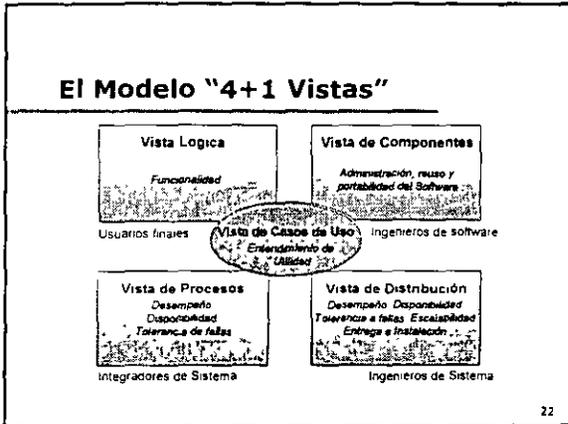
- Perspectivas diferentes para inversionistas diferentes
 - Usuario final, cliente, administrador de proyecto
 - Ingeniero de sistema, desarrollador, arquitecto, evaluador
- Las perspectivas múltiples requieren múltiples vistas
 - Los diagramas de clases no muestran el mapeo del sistema al hardware
 - Los diagramas de bloques de hardware no describen que partes del sistema son obtenidas de software comercial

20

Una Arquitectura requiere Múltiples Vistas

- Para describir completamente una arquitectura, se necesitan cuatro vistas:
 - Una **Vista Lógica** que proporciona una imagen estática de las principales clases y sus relaciones
 - Una **Vista de Componentes** que muestra como está el código organizado en paquetes y librerías, así como el software comercial (COTS, commercial off-the-shelf)
 - Una **Vista de Procesos** que muestra procesos y tareas
 - Una **Vista de Distribución** que muestra los procesadores, dispositivos y ligas en el ambiente operacional
- Finalmente, una **Vista de Casos de Uso** que explica como trabajan juntas las otras cuatro vistas

21



Vista Lógica

- La vista lógica de una arquitectura controla los requerimientos funcionales del sistema
 - Lo que el sistema proporciona en términos de servicios a sus usuarios
- Proporciona una imagen estática de las principales clases y sus relaciones
- La vista lógica se encuentra reflejada en diagramas de clases; paquetes que contienen clases y relaciones, lo cual representa la abstracción del sistema en desarrollo

23

Paquetes Lógicos Globales

- Ciertos paquetes son usados por todos los demás paquetes
 - Foundation Classes
 - Conjuntos, listas, colas, etc
 - Clases de manejo de errores
- Estos paquetes se marcan como globales

Foundation
Classes

global

24

Implicaciones de Dependencia

- Algunas implicaciones de la dependencia entre paquetes son
 - Cada vez que se hace un cambio al paquete Proveedor, el paquete Cliente debe ser re-compilado y reevaluado
 - El paquete Cliente no puede ser reusado independientemente, ya que depende del paquete Proveedor



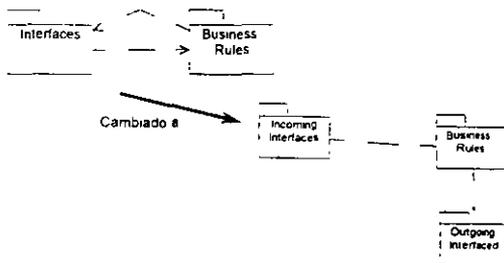
25

Evitar Dependencias Circulares

- Es deseable que la jerarquía del paquete sea a-cíclica
- Esto quiere decir que la situación siguiente debe evitarse (de ser posible)
 - El paquete A usa el paquete B el cual usa el paquete A
- Una dependencia circular como esta, significa que los paquetes A y B tendrán que ser tratados como un solo paquete
- También deben evitarse los círculos con más de dos paquetes
 - El paquete A usa el paquete B que usa al paquete C que usa al paquete A
- Las dependencias circulares pueden romperse, dividiendo uno de los paquetes en dos paquetes más pequeños

26

Dependencia Circular en el Sistema de Inscripción



27

Interfaz Pública de un Paquete

- Una interfaz de paquete debe encapsular su implementación detrás de una interfaz pública justo como lo hace una clase
- Cada clase en un paquete tiene un parámetro de control de exportación que puede establecerse a público o implementación (privado)
 - Solo las clases públicas pueden ser usadas por clases en otros paquetes
 - Las clases de implementación (privadas) sólo pueden ser usadas por su paquete

28

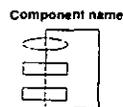
Vista de Componentes

- La vista de componentes tiene que ver con la organización del software en módulos para su desarrollo
- Los diagramas de componentes se crean para mostrar los paquetes y los componentes que conforman el sistema en desarrollo
 - Muestra la distribución de clases a componentes
 - Muestra la distribución de componentes en paquetes
- Los paquetes se organizan en capas, donde cada capa tiene una interfaz bien definida

29

¿Qué es un componente?

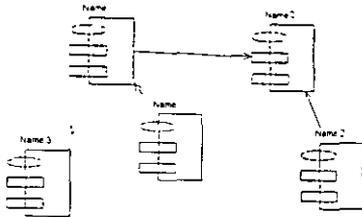
- Un componente es una unidad de código fuente que sirve como bloque constructor para la estructura física de un sistema
- Los estereotipos (con iconos alternos) pueden usarse para definir tipos de componentes específicos
 - Ejemplos: exe, dll, main programs, headers, modules, forms
- Agrupar clases en un componente, ya sea que tenga una función cooperativa o que necesite estar en proximidad para la eficiencia en la implementación
- Notación de componente:



30

Diagramas de Componentes

- Un diagrama de componentes muestra la distribución de clases y objetos en componentes durante la implementación, así como sus dependencias de compilación



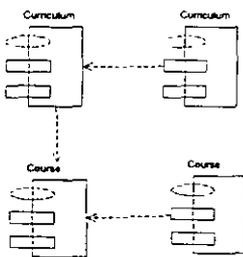
31

Diagramas de Componentes (cont.)

- Se requiere un nombre para cada componente, este nombre denota típicamente el nombre simple del archivo físico correspondiente en el espacio de trabajo actual
- La única relación es la dependencia de compilación, representada por una línea punteada dirigida que apunta hacia donde existe la dependencia
- En C++, la dependencia de compilación se indica por las directivas #include
- No debe haber ciclos en un conjunto de dependencias de compilación

32

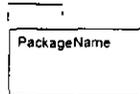
Ejemplo: Diagrama de Componentes



33

Paquetes en la Vista de Componentes

- Un paquete en la vista de componentes es una colección de componentes, algunos de los cuales son visibles a otros paquetes y otros están ocultos
- Un paquete agrupa componentes que están lógicamente relacionados
- Un paquete en la vista de componentes agrupa componentes de manera similar a la que un paquete en la vista lógica agrupa clases
- Cada componente en el sistema debe existir en un solo paquete en el nivel más alto del sistema
- Notación

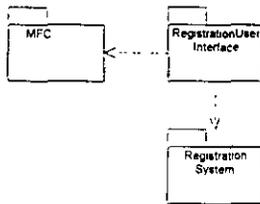


34

Diagrama de Componentes Principal

- Un diagrama de componentes principales es una familia de paquetes conectados por ligas dirigidas que representan dependencias

■ Ejemplo

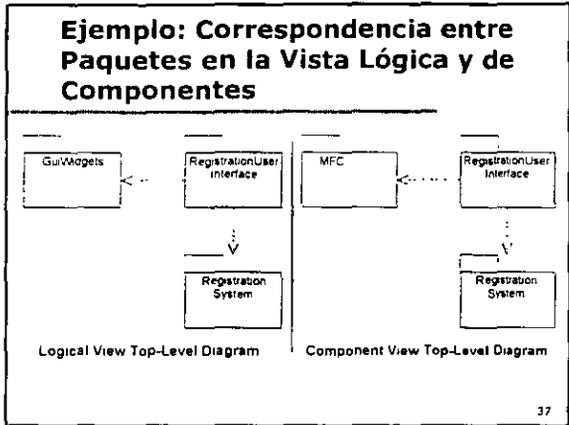


35

Relación entre Paquetes de la Vista Lógica y de Componentes

- En general, un paquete en la vista lógica puede corresponder directamente a un paquete de componentes en la vista de componentes
- La estructura lógica y física puede variar por las siguientes razones
 - Se fusionan paquetes en la vista lógica, para mantener juntos a objetos que se comunican frecuentemente
 - Los paquetes en la vista de componentes se agregan para implementar funcionalidad de bajo nivel

36



Vista de Procesos

- La vista de procesos de la arquitectura se enfoca en la descomposición de procesos
 - La vista muestra la distribución de componentes a procesos
- El diagrama de componentes se actualiza para mostrar la distribución de componentes a procesos
- La vista de procesos está dirigida a: la disponibilidad, la confiabilidad, el desempeño, la administración y la sincronización del sistema

38

Componentes de Proceso

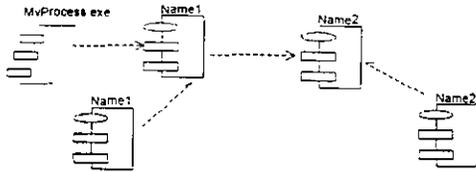
- Las librerías ejecutables y ligadas se representan como componentes UML
 - Especificación de Paquete (DLL)
 - Especificación de Tarea (EXE)

Especificación del paquete (DLL) Especificación de tarea (EXE)

39

Diagrama de Componentes para un Proceso

- Cada componente puede depender de otros componentes



40

Procesos para el Sistema de Inscripción a Cursos

Curriculum.exe



Proceso para la creación y mantenimiento de curriculum

Registration.exe



Proceso para que los alumnos y profesores elijan curso

41

Vista de Distribución

- La vista de distribución de la arquitectura mapea componentes a nodos de procesamiento
- Los requerimientos de desempeño y tolerancia a fallas se toman en cuenta
- Los diagramas de distribución se crean para mostrar los diferentes nodos (procesos y dispositivos) en el sistema

42

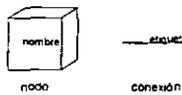
Diagrama de Distribución

- Un diagrama de distribución muestra la ubicación de los componentes en nodos, de tal forma que se obtenga una vista de distribución del sistema
 - Los procesadores y dispositivos son estereotipos comunes de Nodo
- Los nodos se conectan en el diagrama a través de una línea, que refleje la ruta de comunicación entre ellos
- Los elementos esenciales de un diagrama de distribución son los nodos y las conexiones

43

Notación para Diagramas de Distribución

- Un nodo es un objeto físico en tiempo de ejecución que representa recursos computacionales
- Una conexión indica comunicación, generalmente a través de medios de acoplamiento directo al hardware



44

Ejemplo: Diagrama de Distribución para el Sistema de Inscripción

- Este diagrama muestra dos nodos y los dispositivos con los que se comunica el Sistema de Inscripción



45

Procesos

- Un proceso es un hilo de control de la ejecución de un programa o sistema (OO)
 - Un sistema grande puede dividirse en procesos multiples o hilos de control
- Los objetos se asignan a procesos (sus asignaciones pueden ser dinamicas)
- Los procesos se asignan a procesadores (un conjunto de procesos puede ser dinamico)
- Notacion.



46

Mapeo de Paquetes de Desarrollo a Procesos Ejecutables

- El mapeo de paquetes de desarrollo a procesos ejecutables envuelve el entendimiento de la topología del sistema y las prioridades del sistema, que incluyen:
 - Arquitectura de procesador, velocidad y capacidad
 - Mantener las asociaciones de clases juntas para minimizar la comunicacion de interprocesos (IPC interprocess communication)
 - Estrategia IPC -- cliente/proveedor u otro?
 - Técnicas de proceso distribuido
- Debe resolver elementos que envuelvan múltiples procesadores de hardware o sistemas distribuidos durante el diseño del sistema

47

Mapeo de Procesos Ejecutables a Hardware

- Los procesos deben asignarse a un dispositivo de hardware para su ejecucion
- Entre las consideraciones estan
 - Tiempo de respuesta y resultados del sistema
 - Comunicacion ancho de banda/capacidad
 - Localizacion fisica del hardware requerido
 - Necesidades de procesamiento distribuido
 - Balanace de carga de procesos en sistemas distribuidos
 - Tolerancia de fallas
 -

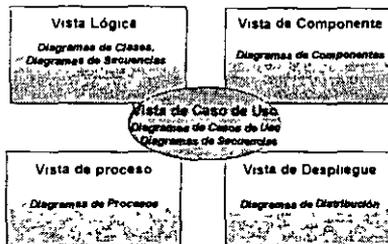
48

Vista de Casos de Usos

- Los casos de usos son los conductores del diseño de una arquitectura
 - Abstracciones de requerimientos largos y complejos
 - Identificación de interfaz crítica
 - Forzar a los diseñadores a concretar elementos
- Demuestran y validan las vistas, lógica, de componentes, de procesos y de distribución de la arquitectura

49

Las "4+1 Vistas" del Modelo UML



50

¿Cómo se documenta una Arquitectura?

- La arquitectura se documenta a través de un texto escrito
 - Aproximadamente 100 páginas para un sistema grande
- El documento incluye
 - Una descripción de la filosofía de la arquitectura (las vistas) y la visión que dirige los requerimientos
 - Ver las ventajas y desventajas para considerar alternativas
 - Una vista de alto nivel de la vista lógica (paquetes y clases principales)
 - Escenarios específicos de la arquitectura
 - Vistas de alto nivel de las vistas de procesos y de distribución
 - Los mecanismos clave

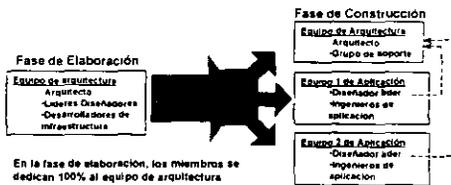
51

¿Quién desarrolla la Arquitectura del Software?

- El equipo de arquitectura esta compuesto por los mejores y mas experimentados desarrolladores
- Establecido tempranamente en el proyecto (no despues de la fase de elaboracion)
- La mayoría de los proyectos de complejidad razonable requieren de un equipo de arquitectura, NO un solo arquitecto
 - Encabezado por el arquitecto en jefe, quien dedica 100% de su tiempo
 - Incluye los líderes diseñadores para funciones mas importantes o criticas del sistema

52

Evolución del Equipo de Arquitectura



En la fase de elaboracion, los miembros se dedican 100% al equipo de arquitectura

Durante la fase de construccion, los miembros se convierten en diseñadores líderes para equipos de aplicacion y soporte de medio tiempo al equipo de arquitectura

53

Beneficios de un Equipo de Arquitectura

- Documentos a entregar
 - Documento de arquitectura
 - Partes del documento de diseño de bajo nivel
 - Guías de diseño y programacion
 - Elementos de los planes de liberación
 - Auditorias de diseño al sistema a liberar
- La habilidad y efectividad del equipo de arquitectura es crítico para el éxito de un proyecto

Con una buena arquitectura un equipo de desarrollo normal puede triunfar. Con una arquitectura debil, hasta los desarrolladores mas expertos no tendran éxito

54

Ejercicio: Diseño de Arquitectura

- Discutir consideraciones de arquitectura para el problema
- Agregar paquetes al modelo como sea necesario
 - Reubicar clases en diferentes paquetes como sea necesario

55

56

Mecanismos Clave



57

Objetivos: Mecanismos Clave

- Usted podrá:
 - Describir algunos mecanismos claves específicos a OO
 - Explicar los elementos asociados con la interfaz a bases de datos
 - Listar algunas consideraciones para evaluar sistemas de administración de bases de datos
 - Describir el manejo de excepciones y sus elementos asociados
 - Explicar los elementos asociados con comunicación inter-proceso

58

¿Qué son los Mecanismos Clave?

- Un mecanismo clave es una decisión estratégica de acuerdo a estándares, políticas y prácticas comunes, por ejemplo
 - Un acercamiento común a un manejo de error, o
 - Un modo común de comunicación entre procesos
- La mayoría del software tradicional se diseña con principios que aun se aplican en el diseño OO
 - Los problemas para resolver son similares, e.g., manejo de recursos, control de riesgos, etc
- Algunas diferencias se deben a
 - Soluciones estructuradas usando metodos OO
 - Los elementos de lenguajes de programación OO

59

Mecanismos Claves Comunes

- Administración de recursos
- Manejo especial requiendo para el inicio y salida del sistema
- Integración con sistemas de almacenamiento de datos persistentes
- Detección/manejo/reporte de errores
- Comunicación interproceso
- Envío de mensajes
- Apariencia y vista (look & feel) de la interfaz de usuario
- Reutilización de software

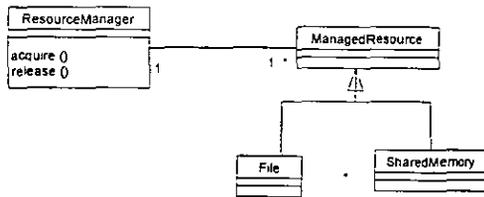
60

Administración de Recursos

- Una clase administradora de recursos puede emplearse para controlar el acceso a los recursos
- La clase administradora de recursos utiliza métodos de software tradicional, tales como semáforos para controlar el acceso a dichos recursos
- Esta clase proporciona operaciones que permiten a los clientes del recurso, obtenerlo, descargarlo, obtener su estado, etc
- Una superclase que contenga la interfaz a los recursos administrados puede ser provista con los recursos del administrador para posibilitar su reuso

61

Administración de Recursos (cont.)



62

Inicio y Salida del Sistema

- Si aun no se ha cubierto durante el analisis, deben definirse los casos de uso para el inicio y salida del sistema
- Los escenarios deben desarrollarse para cada caso de uso -- tantos como sean necesarios para controlar a la mayoría de las situaciones normales y anormales
- Durante este proceso deben descubrirse nuevos estados y comportamientos para clases existentes y la necesidad debe surgir para las clases enteramente nuevas y así controlar el inicio y/o salida del sistema

63

Objetos Persistentes

- Un objeto persistente es aquel que lógicamente existe bajo el ámbito del programa que lo creó
- Los lenguajes de programación OO manejan objetos residentes en memoria, los cuales son esencialmente transitorios
- Un objeto persistente tiene la habilidad de guardar el valor de sus atributos en algún tipo de almacenamiento persistente
- Un objeto persistente puede también crearse en memoria e iniciar con sus valores de atributo desde el almacenamiento persistente
- La estrategia total para proveer persistencia a los objetos en el sistema, es un mecanismo de control

64

Almacenamiento Persistente

- El almacenamiento persistente puede hacerse empleando un sistema de archivos simple o algún tipo de sistema de administración de base de datos
- Hay varios modelos para DBMSs:
 - Jerárquicos
 - Red
 - Relacional (RDBMS)
 - Orientado a Objetos (ODBMS)
- El Relacional y el ODBMS son los más comunes

65

Selección de una Aproximación a Persistencia

- La estrategia de diseño para retener objetos persistentes deberá considerar
 - Tiempos de acceso
 - Capacidad de almacenamiento
 - Confiabilidad del sistema de almacenamiento
 - Acceso a datos existentes
- El modelo elegido influirá al sistema y al diseño de objetos, de tal forma que los diseñadores deberán considerar:
 - Operaciones adicionales para almacenar y recuperar objetos persistentes
 - Adición de atributos para manejar detalles de almacenamiento del sistema
 - Clases adicionales para hacer interfaz con el DBMS

66

Criterios de Evaluación de un DBMS

- Se debe decidir el criterio de evaluación para elegir un DBMS
- Las siguientes laminas contienen cierto criterio encontrado en "Considerations For Evaluating Object Database Management Systems" escrito por Robert Gancarz y Grant Colley, Object Magazine, Marzo/Abril 1992
 - Este criterio tambien se aplica para elegir el sistema de administracion de la base de datos

67

Criterios de Evaluación de un DBMS (cont.)

- Presencia del Vendedor
 - Mirar el poder financiero, estructura de la organización, procedimientos de soporte al cliente, soporte de entrenamiento y consultoria, alianzas con otras compañías
- Desempeño de la Base de Datos
 - Ninguna marca puede probar que un DBMS es más rapido para todas las aplicaciones
 - El desempeño depende de la aplicacion
 - Los prototipos especificos de aplicacion son muy importantes
- Capacidades de la Base de Datos
 - Se debe evaluar administracion de transacciones, control de concurrencia, respaldo y recuperacion, seguridad y soporte de un lenguaje de consulta (SQL)

68

Criterios de Evaluación de un DBMS (cont.)

- Arquitectura de la Base de Datos
 - Evaluar esquemas de control de concurrencia, mecanismos de seguridad y administradores de almacenamiento
- Herramientas de Desarrollo
 - Ver las herramientas para el diseño de la base de datos, modificacion del esquema de las base de datos y navegacion, asi como tambien las herramientas de depuracion y afinacion de la base de datos
- Soporte al lenguaje de eleccion
 - Asegurarse de que hay soporte para el lenguaje de su eleccion para el desarrollo del sistema
- Facilidad de Migracion
 - ¿Que tan facil/dificil es migrar al sistema de la base de datos?

69

Criterios de Evaluación de un DBMS (cont.)

- Integración con Sistemas Legados
 - ¿Qué tan fácil/difícil es integrarse al sistema de administración de base de datos existente?
- Soporte Multi-usuario
 - Evaluación del soporte para desarrollo multiusuario, administración de configuración, control de versiones y estrategias de seguridad

Nota: Invierta tiempo y esfuerzo para seleccionar el sistema de administración de base de datos apropiado para el proyecto

SIEMPRE es más caro corregir que hacerlo bien desde el principio!!

70

Productos de la Base de Datos Relacional

- Hay dos factores principales que deben tomarse en cuenta cuando se diseña un sistema OO usando bases de datos relacionales
- Primero, existe una diferencia semántica natural entre el modelo basado en clases un diseño orientado a objetos y el modelo basado en tablas de una base de datos relacional
 - Se debe definir un mapeo o traducción entre los dos
- Segundo, se debe definir el comportamiento que hace interfaz con el RDBMS y que implementa esta traducción
 - ¿Debe insertarse este comportamiento en objetos persistentes o de algún modo debe mantenerse separado?

71

Mapeo a Bases de Datos Relacionales

- Típicamente, cada clase mapea a una tabla y cada instancia mapea a un renglón

Customer
name
address
discount

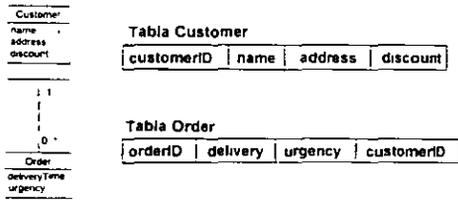
Tabla Customer

customerID	name	address	discount
------------	------	---------	----------

72

Mapeo a Bases de Datos Relacionales (cont.)

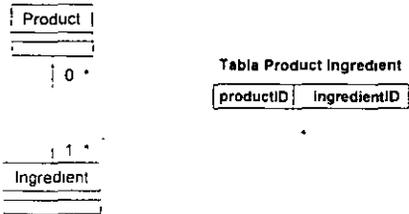
- Las relaciones de uno-a-muchos se implementan usando una llave foránea en la tabla que representa a la clase que tiene la multiplicidad mayor a uno en la relación



73

Mapeo a Bases de Datos Relacionales (cont.)

- Se crean tablas para resolver las relaciones de muchos-a-muchos

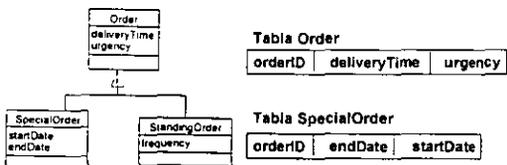


74

Mapeo a Bases de Datos Relacionales (cont.)

- Las superclases / subclases también pueden mapearse a tablas

- Cada clase y subclase es una tabla
- Se proporcionan vistas para la jerarquía



75

Mapeo a Bases de Datos Relacionales (cont.)

- Existen estrategias alternas para superclases y subclases
 - Repetir todos los atributos en la tabla de superclase
 - Problema: espacio desperdiciado
 - Repetir todos los atributos en la tabla de subclase
 - Problema: redundancia
- Se buscan la mejor relación con respecto al desempeño y espacio de almacenamiento para decidir qué mapeo usar en cada situación de herencia

75

Interfaz con el RDBMS

- El elemento principal asociado con la creación de una interfaz con un RDBMS, es si se separa o no el comportamiento específico de la aplicación del comportamiento específico de la base de datos
- Suponga que nuestro sistema tiene una clase Cliente que se ha decidido que será persistente
 - ¿Debera la clase Cliente contener los detalles del mapeo OO-a-RDBMS?
 - ¿Debera la clase Cliente contener el comportamiento para hacer interfaz con el agente RDBMS (es decir, código que genera SQL para leer/escribir de/a la base de datos)?
 - ¿Debera la clase Cliente incluso saber que es persistente?
- De cualquier forma se puede trabajar - cada uno tiene sus ventajas y desventajas

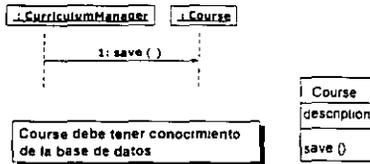
77

Interfaz con RDBMS (cont.)

- El comportamiento específico de la base de datos no está separado del comportamiento específico de la aplicación:
 - Cada clase persistente puede tener las funcionalidades de crear, leer, actualizar y borrar (CRUD) construidas en (operaciones que desempeñan mapeo OO-a-RDBMS y generan SQL para implementarlo)
 - Ventajas
 - No es técnicamente retador para implementar
 - Desventajas
 - Los modelos OO y RDBMS no son separables
 - La funcionalidad CRUD no siempre es limpiamente heredable

78

Comportamiento de la Base de Datos dentro de la Clase



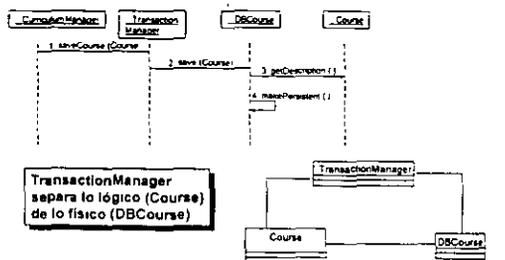
79

Interfaz con RDBMS

- El comportamiento específico de la base de datos se separa del comportamiento específico de la aplicación
 - Se puede modelar al sistema en dos particiones: Aplicación e Interfaz de la Base de Datos
 - Para cada clase persistente, una interfaz de base de datos asociada se define, la cual entiende el mapeo OO-a-RDBMS y tiene el comportamiento para hacer interfaz con el RDBMS
- Ventajas
 - El modelo OO es separable del modelo RDBMS
 - Existen herramientas disponibles para generar las clases de interfaz básicas a la base de datos
- Desventajas
 - Mas técnicamente retardador de implementar

80

Comportamiento de Base de Datos Separado



81

DBMS Orientado a Objetos

- Los ODBMS permiten el almacenamiento y recuperación de los objetos (con datos complejos encapsulados en cada objeto)
- Un ODBMS retiene típicamente
 - Objetos (valores de atributos)
 - Información de la clase sobre cada objeto
- No hay diferencia semántica entre el modelo OO y el modelo ODBMS - son idénticos
- No debe diseñarse comportamiento especial para hacer interfaz con el ODBMS

82

DBMS Orientado a Objetos (cont.)

- Ventajas.
 - Interfaz sin parches entre la aplicación y la base de datos
 - Se necesita relativamente poco código para hacer a los objetos persistentes
 - Muy efectivo con sistemas que deben atravesar estructuras de datos complicadas
- Desventajas
 - Riesgo más alto de desarrollo ya que la tecnología y producto de ODBMS no son tan maduros como sus contrapartes de RDBMS
 - El desempeño con estructuras simples de datos no proporcionan ventajas sobre RDBMS
- Se debe evaluar la inversión de tecnología relacional existente cuando se evalúe la tecnología de bases de datos OO

83

Detección de Errores

- Debe establecerse un mecanismo consistente para la detección de errores
- Los objetos deben detectar errores que podrían violar su integridad - esto incluye errores
 - Que surgen con la operación
 - Que resultan de parámetros recibidos de objetos clientes
 - Que resulten de valores de retorno proporcionados por objetos proveedores
- Se puede establecer un plan para monitorear la salud del sistema
 - Se define operación de prueba para cada clase que verifique la integridad o estructura interna, y
 - Monitoreo de objetos definidos para revisar periódicamente cada función de prueba de objeto

84

Manejo de Errores

- Aun en los sistemas OO, el manejo de errores debe diseñarse cuidadosamente -- en más del 30% del código final existen condiciones de manejo de error
- Los lenguajes OO (como 3.0 C++) proporcionan elementos de manejo de excepción para auxiliar en este diseño
- El manejo de excepciones permite que un error se maneje por un objeto distinto que el objeto que detectó el error
- Esto es con frecuencia apropiado, ya que el objeto detector no siempre conoce el impacto a un sistema por un error

85

Ejemplo de Manejo de Excepción

```
class String {
public:
    class RangeError { int badIndex; }; // error type
    char getChar(int index) const
    //
};
char String::getChar(int index) const {
    if (index < 0 || index > lastValid)
        throw RangeError(index); // throw point
    return contents[index];
};
void foo() {
    try {
        String s = "hello";
        char c = s.getChar(0);
    }
    catch (const String::RangeError& why) { // catch point
        cout << "subscript out of range" << why.badIndex << endl;
    }
};
```

86

Throwing y Catching Excepciones

- Cuando hay un problema que no puede ser manejado en el contexto actual, se puede crear y "lanzar" una excepción
 - throw Problem("things are bad");
- ¿Qué hace throw?
 - Se crea y "regresa" al objeto excepción
- ¿A dónde va el objeto excepción?
 - Cuando se lanza una excepción se busca la llamada a la pila para el primer manejador (handler)
 - Habrá un manejador de excepción para cada tipo de excepción que pueda cargarse
 - catch (Problem&) { // handle exceptions of type Problem }

87

Elementos de Manejo de Excepción

- En el proceso de búsqueda de llamada a la pila, se llaman a los destructores de objetos locales
 - Variables automáticas
 - Parámetros de valor
 - Temporales
- No se llama a los destructores para
 - Variables dinámicas
 - Variables estáticas
- Nunca se ejecuta el código después del punto de lanzamiento
- Las excepciones se apoderan de la administración de recursos (ej. Cerrar archivos abiertos)

88

¿Deben Usarse Siempre las Excepciones?

- No deben usarse las excepciones en las siguientes situaciones.
 - Condiciones ordinarias de error
 - Si hay suficiente información disponible para manejar el error entonces NO hay una excepción
 - Para controlar el flujo del problema
 - Una excepción NO es un regreso alterno

89

¿Cuándo Deben Usarse las Excepciones?

- Las Excepciones se lanzan como resultado de un error serio
 - No hay regreso al punto donde se lanzó la excepción
- Las Excepciones no deben usarse si el error puede manejarse (arreglarse) y el proceso continúa
 - Puede llamarse una operación para "arreglar" el problema y el proceso puede continuar

90

Uso Típico de Excepciones

- Arreglar el problema y continuar el proceso sin procesar de nuevo la función que lanzó la excepción
- Calcular un resultado alternativo
- Lanzar el error a un contexto mas alto
- Terminar el programa
- Ocultar funciones que usan esquemas de error ordinario
- Simplificar el código
- Hacer al código mas fácil de mantener

91

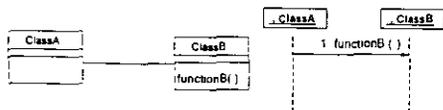
Reporte de Error

- El almacenamiento del registro de errores de forma correcta y el reporte en línea son las claves de la mayoría de los sistemas
- El comportamiento del control de errores consistente puede implementarse en base a la clase Error usada en el manejo de excepción
- Este comportamiento puede incluir
 - Agregar el error a un registro de errores que abarque al sistema (system-wide error log)
 - Distribuir los errores del proceso los cuales facilitan el monitoreo en línea de errores
- Este tipo de acercamiento asegura la consistencia al separar la responsabilidad detallada de las clases de aplicación

92

Comunicación Interprocesos

- ClassA llama a la operación functionB() de la ClassB
- ¿Que sucede cuando las clases ClassA y ClassB están en proceso diferentes?
- Esto se convierte en un resultado crítico en sistemas distribuidos
- Se necesita un mecanismo estándar para la comunicación interprocesos



93

Comunicación Interprocesos (cont.)

- Una solución común que soporta el paradigma OO se ha desarrollado
 - Clases Proxy se insertan en cada proceso el cual proporciona la interfaz de las clases originales y encapsula la comunicación de nivel menor
- La distribución es transparente a clases de aplicación

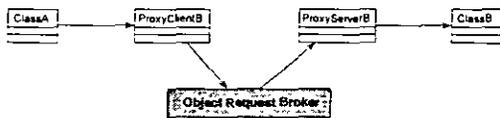
94

Estándares Distribuidos de OO

- La elección de un estándar de distribución es un elemento de diseño (si su sistema usa objetos distribuidos)
- Hay dos estándares para la distribución OO
 - Common Object Request Broker Architecture (CORBA)
 - Component Object Model (COM+/DCOM/COM/OLE)
- Object Request Brokers (ORB) proporciona acceso transparente a objetos en un ambiente distribuido
 - ORB permite la conectividad de ubicación independiente cliente/servidor
 - Las decisiones de distribución pueden tomarse al tiempo de corrida

95

Clases Proxy



96

Planeación para Reuso

- Los componentes reusables deben considerarse previamente en el proceso de diseño para incorporarlo al sistema
- La evaluación de librerías de software comerciales e internas para aplicarlos al sistema por módulos
- Las librerías de clases son grupos de clases que colaboran para proporcionar algún servicio, interfaz o función
- Librerías de clases están comúnmente disponibles como
 - Container objects
 - Interfaces to databases
 - User interface widgets

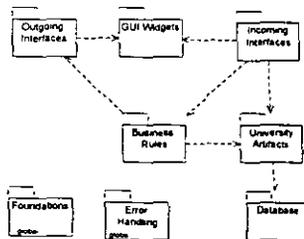
97

Actualización de Diagramas

- Los diagramas de clases se actualizan para mostrar los mecanismos claves elegidos
- Los diagramas de secuencias se actualizan para mostrar las interacciones entre clases descubiertas y clases que representan estrategias de mecanismos clave

98

Diagrama de Clases Actualizado



99

Problema de Nómina



Este ejemplo está basado en la aplicación de nómina encontrada en el libro "Designing Object Oriented C++ Applications Using the Booch Method" Robert C. Martin, Prentice-Hall, 1995

103

Declaración del Problema de Nómina

- El sistema consiste de una base de datos de empleados de una compañía y sus datos asociados, así como tarjetas de chequeo. Todos los empleados se identifican por un número ID único de empleado. El sistema debe pagar a cada empleado la cantidad correcta, a tiempo, por el método que ellos especifican.
- Algunos empleados trabajan pagándoseles una cuota por hora. Entregan tarjetas de chequeo diariamente que registran la fecha y número de horas trabajadas. Si alguno trabaja más de 8 horas, se les paga el 50% adicional de su cuota correspondientes por cada hora extra. Los empleados por hora reciben su pago cada semana.
- Otros empleados reciben un salario y aun así entregan sus tarjetas de chequeo diariamente, ya que contienen las horas trabajadas. De esta manera, el sistema pueda guardar un registro de las horas trabajadas. Estos empleados reciben su pago el último día laborable del mes.

104

Declaración del Problema de Nómina

- Algunos de los empleados asalariados reciben también una comisión basada en sus ventas. De este modo, entregan sus ordenes de compra, en donde se reflejan la fecha y monto de la venta. La cuota de comisión se determina para cada empleado, siendo esta del 10%, 15%, 25% o 35%. Los vendedores reciben su pago cada quince días.
- Inicialmente, el cheque de pago del empleado debe recogerse con el pagador. Los empleados pueden cambiar su método de pago. Pueden obtener sus cheques de pago por correo a la dirección postal que deseen o pueden solicitar depósito directo en una cuenta de banco de su elección.

105

Declaración del Problema de Nómina

- El Administrador de Nómina mantiene actualizada la información del empleado. El Administrador de Nómina es responsable de agregar, borrar y cambiar la información de los empleados, tal como su nombre, dirección, método de pago y clasificación de pago (por hora, asalariado, comisionado)
- La aplicación de nómina se ejecutará cada viernes y el último día laborable del mes. Le pagará a los empleados correspondientes en ese día. El sistema sabrá en qué fecha se les debe pagar a los empleados, entonces generará registros de pagos de la última vez que el empleado recibió su pago en la fecha especificada

106

Solución del Problema Nómina



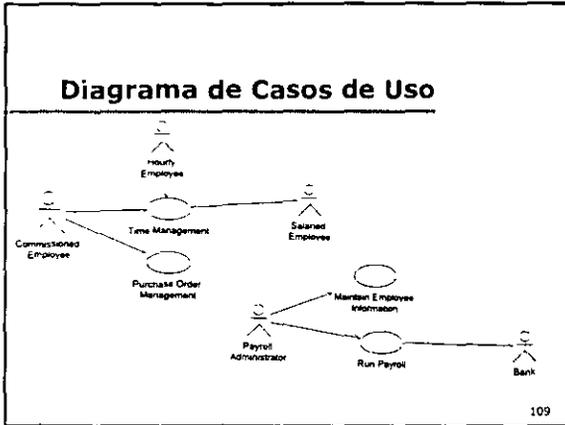
Este ejemplo está basado en la aplicación de nómina encontrado en "Designing Object Oriented C++ Applications Using the Booch Method" Robert C Martin, Prentice-Hall, 1995

107

Ejercicio: Comportamiento del Sistema

- Usando el problema provisto por el instructor
 - Dibuje un diagrama de casos de uso
 - Escriba una definición para cada actor
- Para un casos de uso
 - Escriba una breve descripción (dos sentencias máximo)
 - Escriba el flujo de eventos
 - Liste algunos escenarios posibles

108



- ### Actores para el Sistema de Nómina
- **Empleado por hora**
 - Persona que recibe un pago por hora. Pago de tiempo extra (1 1/2 más de la cuota por hora) se recibe por todas las horas trabajadas en exceso de 40 horas por semana.
 - **Empleado asalariado**
 - Persona que se le paga un salario fijo.
 - **Empleado comisionado**
 - Empleado asalariado que también recibe una comisión por sus ventas.
 - **Administrador de nómina**
 - Persona responsable del mantenimiento de la información de los empleados. El administrador genera la nómina.
 - **Banco**
 - Entidad receptora de los pagos de nómina depositados en cuentas definidas por los empleados.
- 110

- ### Breves Descripciones
- **Mantenimiento de Información de Empleado**
 - Este caso de uso es iniciado por el Administrador de Nómina. Proporciona la capacidad de agregar, modificar, borrar y/o revisar la información del empleado necesaria para el sistema de nómina.
 - **Ejecución de Nómina**
 - Este caso de uso es iniciado por el Administrador de Nómina. Proporciona un mecanismo para generar los pagos a los empleados.
- 111

Breves Descripciones

- **Administración de Tiempo**
 - Este caso de uso es iniciado por cualquier tipo de Empleado
Proporciona la capacidad de registrar, modificar, borrar y/o revisar horas trabajadas para una semana específica

- **Administración de Ordenes de Compra**
 - Este caso de uso es iniciado por un Empleado Comisionado
Proporciona la capacidad de registrar, modificar, borrar y/o revisar ordenes de compra

112

Flujo de Eventos para el Caso de Uso de Mantenimiento de Información de Empleado

- **2.1 Pre-condiciones**

Ninguna

- **2.2 Flujo Principal**

Este caso de uso inicia cuando el Administrador de Nómina introduce su número id de empleado. El sistema verifica que el número id de empleado introducido sea válido. El sistema despliega las actividades disponibles y le pide al usuario que seleccione una: ADD, DELETE, MODIFY, REVIEW, o QUIT.

Si la actividad seleccionada es ADD, el A-1: Se desempeña un subflujo de New Employee

Si la actividad seleccionada es DELETE, el A-2 Se desempeña un subflujo de Delete an Employee.

Si la actividad seleccionada es MODIFY, el A-3 Se desempeña un subflujo de Modify an Employee

113

Flujo de Eventos para el Caso de Uso de Mantenimiento de Información de Empleado

Si la actividad seleccionada es REVIEW, el A-4 se desempeña un subflujo de Review an Employee

Si la actividad seleccionada es QUIT, el caso de uso termina

El usuario puede cancelar el caso de uso en cualquier punto en el flujo de eventos, en el que los cambios no toman lugar para el empleado que se está procesando y el caso de uso inicia de nuevo

- **2.3 Flujos alternos**
 - A-1 Agregar un Empleado Nuevo
El sistema generara un número id de empleado y desplegará la pantalla del empleado. El sistema llenara el número id del empleado. El usuario debe llenar la siguiente información, nombre y dirección del empleado. El usuario deberá escribir el tipo de empleado por hora, asalariado o comisionado

114

Flujo de Eventos para el Caso de Uso de Mantenimiento de Información de Empleado

Si el tipo de empleado es por hora, el usuario deberá proporcionar el rango de horas. Si el tipo de empleado es asalariado, el usuario deberá proporcionar el salario. Si el tipo de empleado es comisionado el usuario deberá proporcionar el salario y el porcentaje de comision. El usuario deberá proporcionar el metodo de pago: recogerlo con el Pagador, deposito directo o correo. Si el metodo de pago es deposito directo, el usuario deberá proporcionar el numero de cuenta del banco y su direccion. Si el metodo de pago es correo, el usuario deberá proporcionar la direccion postal. Toda la informacion es validada por el sistema (E-1) despues de haberla introducido.

Cuando ya se introdujo toda la informacion, el usuario le pide al sistema que procese al empleado. El sistema guarda la informacion para su uso futuro (E-2) y el caso de uso inicia de nuevo.

115

Flujo de Eventos para el Caso de Uso de Mantenimiento de Información de Empleado

A-2 Borrar un empleado

El usuario introduce un numero de id. El sistema lo valida (E-3) y despliega la informacion del empleado. El sistema pide al usuario que confirme el borrado. Si se confirma, el empleado es borrado del sistema. Si el borrado no es confirmado, la peticion se cancela. El caso de uso inicia de nuevo.

A-3 Modificar un Empleado

El usuario introduce un numero de id. El sistema lo valida (E-3) y despliega la informacion del empleado. El usuario actualiza los campos requeridos. El sistema verificara la informacion despues de que se introdujo (E-1). Despues de que se han hecho todos los cambios, el usuario le dice al sistema que procese al empleado. El sistema guarda la informacion para uso futuro (E-2) y el caso de uso inicia de nuevo.

116

Flujo de Eventos para el Caso de Uso de Mantenimiento de Información de Empleado

A-4 Revisar un Empleado

El usuario introduce un numero de id. El sistema lo valida (E-2) y despliega la informacion del empleado. Cuando el usuario indica que ya termino de revisar, el caso de uso inicia de nuevo.

2.4 Flujos de excepcion

E-1 **Informacion de empleado invalida.** El usuario sabe que se ha introducido informacion invalida. El usuario puede re-introducir la informacion para terminar el caso de uso.

E-2 **El sistema no puede guardar la informacion del empleado.** El usuario sabe porque no se puede guardar la informacion. El caso de uso inicia de nuevo.

E-3: **La informacion del empleado no puede ser desplegada.** El usuario sabe porque la informacion no puede ser desplegada. El caso de uso inicia de nuevo.

117

Flujo de eventos para el Caso de "Uso de Ejecución de Nómina"

- 2.1 Pre-condiciones
Ninguna

- 2.2 Flujo Principal

El caso de uso empieza cuando el Administrador de Nómina introduce su número id de empleado. El sistema verifica que el número id de empleado introducido sea válido y capaz de generar la nómina (E-1). El usuario introduce la fecha de nómina y le pide al sistema que genere la nómina. El sistema obtiene los datos de todos los empleados que deben recibir pago la fecha deseada (E-2). El sistema calcula el pago y las deducciones legales. Si el método de pago es recogerlo con el Pagador o Correo, el sistema imprime un cheque de pago. Si el método de pago es depósito bancario, el sistema creará una transacción bancaria. El caso de uso termina cuando todos los empleados que deban recibir pago en la fecha deseada hayan sido procesados.

118

Flujo de eventos para el Caso de "Uso de Ejecución de Nómina"

- 2.3 Flujos Alternos
Ninguno

- 2.4 Flujos de Excepción

E-1 Id. inválido introducido. El usuario puede re-introducir un número id o terminar el caso de uso.

E-2 La información del empleado no puede ser obtenida. El usuario sabe porque no se puede obtener la información. El caso de uso inicia de nuevo desde el principio.

119

Flujo de Eventos para el Caso de "Uso de Administración de Tiempo"

- 2.1 Pre-condiciones
Ninguna

- 2.2 Flujo Principal

El caso de uso inicia cuando el Empleado introduce su número id de empleado. El sistema verifica que el número id de empleado introducido sea válido (E-1). El sistema despliega las actividades disponibles y le pide al usuario que seleccione una. ADD, DELETE, MODIFY, REVIEW, o QUIT.

Si la actividad seleccionada es ADD, el A-1. Se desempeña un subflujo de Add a New Timecard.

Si la actividad seleccionada es DELETE, el A-2. Se desempeña un subflujo Delete a Timecard.

120

Flujo de Eventos para el Caso de Uso de Administración de Tiempo

Si la actividad seleccionada es MODIFY, the A-3 Se desempeña un subflujo de Modify a Timecard

Si la actividad seleccionada es REVIEW, el A-4 Se ejecuta un subflujo de Review a Timecard

Si la actividad seleccionada es QUIT, el caso de uso termina

El usuario puede cancelar el caso de uso en cualquier punto del flujo de eventos y el caso de uso inicia de nuevo

121

Flujo de Eventos para el Caso de Uso de Administración de Tiempo

- 2.3 Flujos alternos

A-1 Agregar una Tarjeta de tiempo nueva
El sistema desplegará la pantalla de la tarjeta de tiempo. El usuario deberá llenar la siguiente información: número id de empleado, fecha y horas trabajadas
Cuando ya se introdujo toda la información, el usuario le pide al sistema que procese la tarjeta de tiempo. El sistema asocia la tarjeta de tiempo con el empleado correcto, guarda la información para uso futuro (E-2) y el caso de uso inicia de nuevo

A-2 Borrar una Tarjeta de tiempo
El usuario introduce el número id de empleado y la fecha. El sistema retrieves (E-3) y despliega la información de la tarjeta de tiempo. Se le pregunta al usuario si confirma el borrado. Si lo confirma, se borra la tarjeta de tiempo del sistema. Si no se confirma el borrado, se cancela la petición. El caso de uso inicia de nuevo

122

Flujo de Eventos para el Caso de Uso de Administración de Tiempo

A-3 Modificar Timecard
El usuario introduce el número de id del empleado y la fecha. El sistema obtiene (E-3) y despliega la información de timecard. El usuario actualiza el número de horas trabajadas y le pide al sistema que procese el timecard. El sistema guarda la información para uso futuro (E-2) y el caso de uso inicia de nuevo

A-4 Revisar Timecard
El usuario introduce un número de id del empleado y una fecha. El sistema obtiene (E-3) y despliega la información de timecard. El usuario indica que ya finalizó la revisión y el caso de uso inicia de nuevo

123

Flujo de Eventos para el Caso de Uso de Administración de la Orden de Compra

■ 2.3 Flujos alternos

A-1 Agregar New Purchase Order

El sistema despliega la pantalla de orden de compra. El usuario debe llenar la siguiente información: número de orden de compra, fecha, producto comprado, cantidad de la venta, nombre del cliente y dirección del cobro del cliente

Cuando se introdujo toda la información, el usuario le pide al sistema que procese la orde. El sistema guarda la información para uso futuro (E-2) y el caso de uso inicia de nuevo

127

Flujo de Eventos para el Caso de Uso de Administración de la Orden de Compra

A-2 Borrar una orden de compra

El usuario introduce un número de id de empleado y un número de orden de compra. El sistema obtiene (E-3) y despliega la información de la orden de compra. Se le pide al usuario que confirme el borrado. Si lo confirma, se borra la orden de compra del sistema. Si no se confirma, se cancela la petición. El caso de uso inicia de nuevo

A-3 Modificar orden de compra

El usuario introduce un número de id de empleado y un número de orden de compra. El sistema obtiene (E-3) y despliega información de la orden de compra. El usuario actualiza los campos necesarios y le pide al sistema que procese la orden de compra. El sistema guarda la información para uso futuro (E-2) y el caso de uso inicia de nuevo

128

Flujo de Eventos para el Caso de Uso de Administración de la Orden de Compra

A-4. Revisión de la orden de compra

El usuario introduce el número de id de empleado y el número de la orden de compra. El sistema obtiene (E-3) y despliega la información de la orden de compra. Cuando el usuario indica que ya termino de revisar, el caso de uso inicia de nuevo.

■ Flujos de Excepcion

E-1 Id introducido no valido El usuario puede re-introducir el número de id o finaliza el caso de uso

E-2 El sistema es incapaz de guardar la información de la orden de compra El usuario sabe porque no se pudo guardar la información. El caso de uso inicia de nuevo desde el principio.

E-3: La información de la orden de compra no puede ser obtenida El usuario sabe porque no se pudo obtener la información. El caso de uso inicia desde el principio.

129

Algunos Escenarios para el Caso de Uso de Mantenimiento de la Información de Empleado

- Crear un empleado por hora
- Crear un empleado comisionado
- Crear un empleado asalariado
- Cambiar una categoría de pago de empleado
- Cambiar un método de entrega de pago de empleado
- Cambiar otra información de empleado
- Revisar información de empleado
- Borrar un empleado

130

Ejercicio: Identificación de Clases

- Tome un caso de uso desarrollado en la lección previa
 - Diagrame al menos un escenario en el diagrama de interacción
 - Encuentre las clases entity, boundary y control que sean necesarias
 - Defina cada clase
- Cree paquetes para el modelo
- Ponga las clases descubiertas en paquetes
- Cree un diagrama de clases iniciales

131

Blank box for student work.

132
