

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Facultad de Ingeniería



**Control de Asistencias del Personal Auxiliar Externo
(CAPAE)**

T E S I S

Que para obtener el título de:

INGENIERO EN COMPUTACIÓN

PRESENTA

HILARIO GARCÍA ANAYA

DIRECTOR DE TESIS: JUAN JOSE CARREÓN GRANADOS.

México D.F. Junio del 2008

AGRADECIMIENTOS

A mis padres†, por su sacrificio y todo lo que nunca les pude dar.

A mi esposa Irma, por su cariño y paciencia.

A mis hijos Luis y Sergio, por lo que significan.

Al Instituto Mexicano del Petróleo y compañeros de trabajo por lo que de ellos he recibido.

CONTENIDO

AGRADECIMIENTOS	1
INTRODUCCIÓN	3
RESUMEN.....	4
Capítulo 1. Antecedentes.....	5
1.1 Acerca del Instituto Mexicano del Petróleo (IMP).....	5
1.2 Problemática	6
1.3 Solución	8
Capítulo 2. Selección de la metodología.....	11
2.1 Consideraciones preliminares.....	11
2.2 Proceso unificado de Rational (RUP) y Lenguaje unificado de modelado (UML)....	12
2.3 Breve descripción del desarrollo	14
2.4 Consideraciones particulares	15
Capítulo 3. Del modelado al diseño	17
3.1 Enfoque del desarrollo.....	17
3.2 Modelado del negocio	18
3.3 Requerimientos.....	20
3.4 Análisis y diseño.....	28
Capítulo 4. De la implementación a la distribución	38
4.1 Implementación	38
4.2 Pruebas	53
4.3 Distribución	56
Capítulo 5. Resultados y discusión	62
Conclusiones	64
Anexo I: Definición de los casos de uso	65
Anexo II: Prototipos y diagramas de secuencia de los “casos de uso”	72
Anexo III: Diagramas de clases	79
Anexo IV: Código relevante	89
Anexo V: Pantallas finales	94
REFERENCIAS.....	103

INTRODUCCIÓN

Este trabajo aunque se ha estructurado siguiendo las reglas de una tesis, más que nada **describe un trabajo realizado en el Instituto Mexicano del Petróleo (IMP)**.

El aspecto más relevante de nuestro trabajo, es el hecho de mostrar como es posible aplicar en la práctica, elementos de diseño de sistemas que normalmente se usan en sistemas “grandes¹ y complejos”, y también a sistemas de dimensiones modestas.

En este documento además de describir de una manera concisa y práctica los pasos que se efectuaron para llegar al sistema que hemos nombrado “Control de Asistencia del Personal Auxiliar Externo” (CAPAE), también se han incluido algunos de los aspectos técnicos mas relevantes.

El CAPAE como surgió de una necesidad real en uno de los proyectos del IMP, pero sus características lo hacen susceptible de aplicarse a cualquier proyecto que contrate personal externo a la institución.

Para una mejor ilustración del trabajo, los diagramas se han trazado de acuerdo al “Lenguaje Unificado de Modelado” [KIMMEL1]

¹ Que hacen uso de una gran cantidad de recursos.

RESUMEN

En el capítulo uno, se explica en forma muy general que es y que hace el IMP, la forma en que trabaja y del porque la necesidad de crear un sistema para facilitar el control de la asistencia del personal, que es contratado a través de compañías externas al IMP (esquema conocido como “outsourcing”).

En el capítulo dos hacemos alusión de algunos de los métodos, que podrían aplicarse en la creación del sistema mencionado, también se indican los criterios que nos llevaron a decidimos por las metodologías conocidas como “proceso unificado” (UP por sus siglas en Ingles) y el “lenguaje unificado de modelado” (UML por sus siglas en inglés), para realizar los diagramas. Finalmente en este capítulo también se plantean cuales fueron las consideraciones iniciales así como las particulares al aplicar las metodologías seleccionadas.

Considerando el UP, en los siguientes dos capítulos nos abocamos a describir, con mayor detalle, cada uno de los pasos que se llevaron a cabo, para la creación del sistema, así como los resultados concretos de haber aplicado la metodología seleccionada, por el tipo de contenido estos son los capítulos mas extensos e importantes de nuestro trabajo.

Finalmente en los dos últimos capítulos, aunque muy breves y no por ello menos importantes, se hace una descripción de los resultados, y las conclusiones de este trabajo.

Para complementar, y aclarar conceptos se ha incluido en los anexos una parte de la información técnica que normalmente este tipo de proyectos de software conlleva.

Capítulo 1. Antecedentes

1.1 Acerca del Instituto Mexicano del Petróleo (IMP)

El Instituto Mexicano del Petróleo (IMP), creado el 23 de agosto de 1965, es el centro de investigación de México dedicado al área petrolera, cuyos objetivos principales son realizar investigación y desarrollo tecnológico, así como servicios especializados orientados a las necesidades estratégicas y operativas de Petróleos Mexicanos (PEMEX); el IMP entrega soluciones integrales innovadoras y desarrolla recursos humanos especializados con un enfoque de calidad, oportunidad y precios competitivos[WEBIMP 1]

La estructura organizacional y de trabajo del IMP, es algo compleja no solo se puede ver como una institución que investiga y hace desarrollo, también es una institución de enseñanza ya que imparte un posgrado con enfoque orientado a la industria petrolera, aunado a esto se debe considerar que PEMEX le encomienda al IMP una gran cantidad de trabajos de muy diversa índole.

Debido a que a PEMEX es su cliente principal, el IMP además de contratar personal de diversas especialidades ha establecido varios centros de trabajo, en las diferentes regiones donde labora el personal de PEMEX.

Para poder administrar al personal que labora normalmente, y los trabajos que estos realizan, se han creado diversos y complejos sistemas informáticos.

El personal tradicional que tiene el IMP registrado como empleados, en sus sistemas de administración lo podemos subdividir en:

- De base
- Investigadores
- Directivos
- Honorarios

Pero el IMP frecuentemente requiere la asistencia de "Personal Auxiliar Externo" (**PAE**), el cual se contrata a través de **compañías externas al IMP** ("outsourcing"), y además por su naturaleza de centro de investigación, ocasionalmente tiene becarios trabajando en algunos de sus proyectos.

1.2 Problemática

En los últimos años el IMP ha estructurado sus actividades usando el modelo de trabajos por proyecto.

Estos proyectos pueden ser:

- **Acotados:** Tienen duración variable que pueden ser de algunos meses a varios años. Estos proyectos son creados, para atender clientes externos como PEMEX.
- **Institucionales:** Su duración es indefinida, generalmente para atender a clientes internos, ("servicios internos") y eventualmente apoyar a proyectos acotados.

En la práctica se ha observado que, tanto en proyectos institucionales, como en proyectos acotados, los controles para la supervisión del personal auxiliar externo (PAE), son controles que no están establecidos bajo ninguna norma, por lo que cada proyecto lo hace a su manera, existiendo la posibilidad de que en algunos casos no se ejerza la debida supervisión de este personal, con el consiguiente desperdicio(o posible desperdicio) de recursos.

En particular el "laboratorio de software" (proyecto institucional), contrató a dos compañías para dar soporte técnico a los usuarios de los equipos de cómputo, al interactuar con el personal de estas compañías, surgieron algunos problemas con respecto a su asistencia al IMP, en vista de lo cual se decidió crear un sistema para registrar y supervisar la asistencia de este personal externo.

El sistema debería de enfocarse primordialmente a:

- Controlar la asistencia del PAE
- La emisión de reportes

Aunque este trabajo inicialmente nació para las necesidades de un proyecto en particular (como se indicó anteriormente), es evidente observar que su aplicación es justificable a cualquiera de los proyectos de la institución.

Si se consideran uno o dos proyectos, realmente el tiempo dedicado a la supervisión de la asistencia del PAE, no es mucho, pero si consideramos todos los proyectos de la institución que contratan personal de “outsourcing”, nos damos cuenta de que el costo del tiempo dedicado a este esfuerzo puede ser realmente considerable, por lo que el sistema que resuelva la problemática mencionada, al aplicarse a una gran cantidad de proyectos, permitiría la optimización de los recursos existentes.

Aun más, por las tendencias actuales de la globalización, se espera que aumente este tipo de contrataciones, en consecuencia, los sistemas de control de este tipo son imprescindibles en cualquier institución que pretenda competir con otras similares.

Debido a la naturaleza de la relación de la institución, con el PAE, puede ser costoso implementar los mismos mecanismos de control de asistencia que tienen los empleados del IMP, ya que implicaría modificar sistemas institucionales de administración que actualmente solo se enfocan al personal tradicional anteriormente mencionado, por ejemplo el control de asistencia que se lleva a cabo en IMP para el personal normal registrado (ver capítulo anterior), es a través del uso de su “gafete” que contiene además de su foto y datos básicos, un código interno, el cual es leído por un sistema electromagnético acoplado a torniquetes en las entradas del IMP.

También se debe considerar que este PAE tiene un alto grado de movilidad, debido a frecuentes casos de:

- Finalización del proyecto que los contrata.
- Cambio de compañía por parte del proyecto.
- Cambio de personal por parte de la misma compañía.

Como resultado de la problemática expuesta se tiene la necesidad de desarrollar un sistema con el principal objetivo de: **supervisar y controlar la asistencia del personal auxiliar**

externo de un proyecto en particular, pero que pueda ser aplicado a cualquier otro proyecto de la empresa.

1.3 Solución

Después de un análisis inicial obtenemos las **características generales del sistema y la funcionalidad deseada.**

1.3.1 Características generales

- El sistema debe ser práctico:
 - Manejar la información a nivel de Proyecto.
 - Con el menor uso de recursos posible, para evitar largos tiempos y altos costos de desarrollo.
 - Manejar información por mes.
 - Generar datos en un formato que pueda ser usado por otros sistemas.
- Se pueda implementar en la red interna del IMP (intranet) para aprovechar la infraestructura y experiencia que se tiene en ese ambiente.
- Permita consultar la información por cualquier persona dentro del IMP. Esto debido a que la información no requiere ocultarse y si ponerla a disposición de las mismas compañías de “outsourcing”, de este modo el mismo personal puede verificar su asistencia y disponibilidad.
- Poder operar (y modificar datos de) el sistema uno o más elementos del proyecto (empleados del IMP).
- Tener un administrador para supervisar y establecer las características iniciales del sistema.
- Tener la posibilidad de supervisar o auditar las acciones del sistema.

1.3.1 Funcionalidad.

Al término del sistema éste deberá tener las siguientes funciones:

CAPAE

- Captura de asistencia del PAE, a través de registros de:
 - Entrada.
 - Salida
 - Entrada/salida simultáneos para casos particulares.
- Obtener reportes de horas trabajadas de cada persona por compañía:
 - A la semana.
 - Al mes.
 - Día por día durante el mes.
- Actualizar datos del PAE:
 - Altas.
 - Bajas.
 - Datos de un elemento.
- De operación y administración para:
 - Identificación (login).
 - Altas y bajas de operadores.
 - Altas y bajas de compañías.
 - Consultar registros de movimientos en bitacora(Log).

Debido a que en la práctica la parte inicial y función más visible es la “Captura de Asistencia”, al sistema a desarrollar se le denominó inicialmente:

“Captura de Asistencia del Personal Auxiliar Externo” (CAPAE)

Pero al final del proceso y debido a que el objetivo es poder controlar las asistencias se decidió renombrarlo como:

“Control de Asistencia del Personal Auxiliar Externo” (CAPAE)

CAPAE

Una vez definidas las características principales del problema, y del sistema que lo resuelve, en el siguiente capítulo estableceremos el método con el cual podremos desarrollar el sistema mencionado.

Capítulo 2. Selección de la metodología

2.1 Consideraciones preliminares

A partir del surgimiento de las computadoras, la programación de los sistemas de información siempre ha sido un gran reto para los profesionales de la informática. De 1980 a 1990, dos elementos dominaron la ingeniería del software, el crecimiento explosivo de aplicaciones, y el florecimiento de nuevas herramientas y paradigmas como lo indica [Braude1]

De tal manera que actualmente en la industria se discute mucho que proceso de desarrollo seguir en la construcción del software, algunos se declaran seguidores de métodos muy elaborados (o pesados) que incluya todas las actividades básicas para la construcción del software:

1. Definición del proceso (de desarrollo de software) que se usara
2. Administración del proyecto de desarrollo
3. Descripción del producto de software que se desea
4. Diseño del producto
5. Implementación del producto
6. Prueba de las partes del producto
7. Integración de las partes del producto y pruebas del producto completo
8. Mantenimiento del producto

Ahora bien en la práctica, es evidente que estas actividades básicas se verán modificadas por factores como la magnitud del problema, recursos (y aun la experiencia misma de quien desarrolla), por lo mismo y en contraposición a los métodos pesados, han surgido otros métodos menos complejos (o ligeros), como por ejemplo la “Programación Extrema”.

Así que cuando nos enfrentamos a desarrollos modestos, donde quien planea, desarrolla, programa, prueba y en ocasiones hasta usa, es una sola persona, nos lleva a considerar el uso de métodos hechos a la medida, que nos permitan aprovechar de una manera efectiva los recursos de que se disponen, y minimizar el tiempo de desarrollo, es debido a lo anterior que en un desarrollo como el que trata este documento, algunas de las actividades básicas son reducidas a su mínima expresión, y aunque inicialmente fueron consideradas no aparecen formalmente en nuestro documento.

Esto no quiere decir que se haga de lado, o se desprece la teoría existente, si no que se debe reconocer que para estas actividades se pueden y deben aplicar uno o más de los paradigmas del desarrollo de software, pero de acuerdo a las condiciones particulares de cada caso.

2.2 Proceso unificado de Rational (RUP) y Lenguaje unificado de modelado (UML)

Después de un breve análisis y considerando lo anteriormente expuesto nos decidimos por un método basado en el “Unified Process” (UP) o Proceso Unificado en español, los criterios que se consideraron principalmente fueron:

- Su adaptabilidad.
- Experiencia en otros proyectos del IMP, donde se ha aplicado
- Amplia aceptación en el medio y extensa bibliografía disponible

El UP es un proceso iterativo para el desarrollo del software creado por Ivar Jacobson y otros autores [ARLOW1], basándose en el UP, la compañía “Rational” creó el “Rational Unified Process”(RUP) en español “Proceso Unificado de Rational”, actualmente “Rational” es una familia de software de IBM, El RUP no es solo un proceso preceptivo concreto, sino un marco de proceso adaptable. Como tal, RUP describe cómo desarrollar el software usando con eficacia técnicas probadas. Mientras que el RUP abarca una gran cantidad de diversas actividades (Ver Fig. 2.1), así como una serie de documentos

auxiliares, que se generan durante todas las etapas del desarrollo del sistema, es importante resaltar que puede ser adaptado, en el sentido de seleccionar los procesos del desarrollo apropiados a una organización o un software en particular.

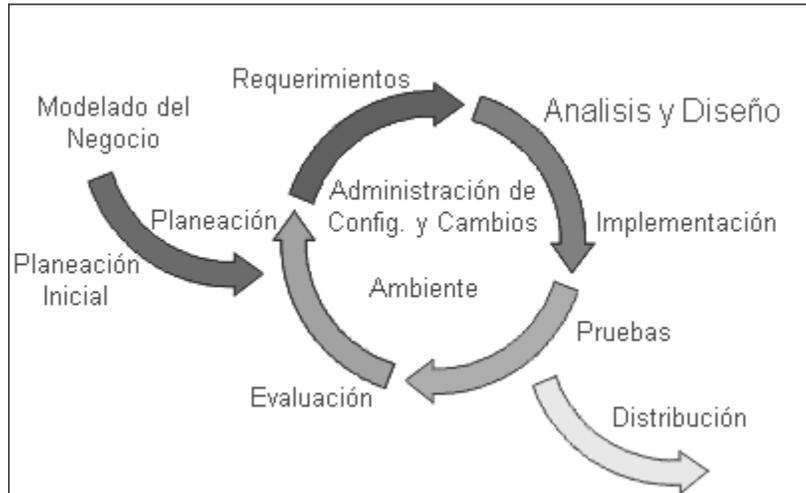


Fig. 2.1 Flujos de trabajo tomado de [JAVAHISPANO1]

Otro de los elementos importantes que nos auxilia para el desarrollo de los sistemas es la manera en que comunicamos y/o documentamos nuestros sistemas, en proyectos grandes donde intervienen varias personas son aun más importantes estos dos elementos (comunicación y documentación).

Como ya se indico anteriormente en RUP, existen una gran cantidad de documentos que es posible generar siguiendo las pautas marcadas por el mismo método, pero en proyectos medianos y pequeños, algunos no son realmente requeridos.

Pero por otro lado dado la importancia que siempre tiene, el poder comunicar nuestro diseño por pequeño que sea, aun para nosotros mismos como referencia o documentación se impone el uso de un lenguaje que haga posible esto, de modo que como lo sugiere el mismo RUP (y otras metodologías) usaremos en nuestros diagramas el “**Unified Modeling language**”(UML) o Lenguaje Unificado de Modelado en español.

“El UML es una definición oficial de un lenguaje pictórico con símbolos y relaciones comunes que tienen un significado común. Si todos los participantes “hablan” UML, entonces las imágenes tienen el mismo significado para todos aquellos que las observen. Por lo tanto, aprender UML es esencial para ser capaz de usar imágenes para experimentar barata, flexible y rápidamente las soluciones.” [KIMMEL1]

2.3 Breve descripción del desarrollo

A continuación una muy breve descripción de las etapas consideradas para el desarrollo del CAPAE.

Modelado del Negocio

En esta etapa se define:

- La arquitectura del negocio.
- Relación y definición de los “actores” del negocio.
- Relación (y definición) de los “actores” con los casos de uso.
- Como interactúan los elementos anteriores.

Requerimientos:

Para establecer y mantener acuerdos con los usuarios y afectados en lo que debe hacer el sistema. Para proveer a los desarrolladores del sistema una mejor comprensión de los requisitos del sistema. Para definir los límites del sistema. Para proporcionar una base para estimar costos y tiempos de desarrollo del sistema. Para definir un usuario-interfaz para el sistema, centrándose en las necesidades y las metas de los usuarios

Se especificaran los detalles de la funcionalidad del sistema, principalmente en lo que se refiere al software y detalles de los casos de uso.

Análisis y Diseño:

A partir de la funcionalidad y los casos de uso se realizará la definición de modelos de análisis de diseño y de implementación del sistema.

Implementación:

Para definir: la organización del código, en términos de subsistemas y/o capas, así como elementos de la implementación (archivos de código, binarios, etc.), pruebas de componentes como unidades e integrar los resultados.

Pruebas:

Actividades que permitirán asegurar se cumple con los requerimientos establecidos.

Distribución:

Actividades asociadas con el aseguramiento de la disponibilidad del producto para los usuarios.

2.4 Consideraciones particulares

Como lo indica la figura 2.1, nuestros procesos están conectados en forma de círculo indicando que para llegar a la implementación, es posible se requieran varias repeticiones de los procesos (iteraciones) de tal modo que sea posible refinar el producto en forma gradual y no de una manera directa al realizar una sola vez estos procesos.

Para complementar es importante establecer que algunos elementos (o procesos) **no fueron considerados** debido a las características particulares de este desarrollo, y los mencionamos a continuación:

- **Planeación inicial y planeación:** Estas actividades son usuales en proyectos de gran tamaño, que por requerir el manejo de una gran cantidad de elementos es muy importante definirlos y acotarlos.
- **Evaluación:** Actividad que se efectúa una vez que el sistema esta en funcionamiento y permite establecer requerimientos para futuros cambios.
- **Administración de configuración y cambios:** Se enfoca en los cambios a productos preexistentes y al manejo de versiones.

- **Ambiente (del sistema):** Para definir y manejar el ambiente en el cual se está desarrollando el sistema. Incluye descripciones de proceso

La descripción que se hace en este documento del método usado (UP y UML) es sólo un breve resumen para enmarcar este trabajo.

Una vez que hemos definido brevemente el método (UP y UML) a usar para resolver nuestro problema (desarrollo del sistema CAPAE), así como las consideraciones, en el siguiente capítulo se hará la descripción de su aplicación.

No está demás recordar y hacer hincapié en el hecho que tanto estas metodologías como los lenguajes relativamente modernos, se apoyan en el uso de *objetos*, por lo que se hace imperativo que la persona o personas que intervengan en este tipo de desarrollos, manejen con cierta facilidad los términos y conceptos asociados, como clases, propiedades instancias, etc.

Por la misma razón algunos de estos términos, se usan en los siguientes capítulos.

Capítulo 3. Del modelado al diseño

3.1 Enfoque del desarrollo.

Se hicieron varias iteraciones, para llegar al producto final, aunque aquí solo se describirá lo realizado durante cada etapa o flujo de trabajo (ver fig. 3.1) sin indicar en que iteración. Lo anterior no solo fue debido a que la teoría formal dicta que las iteraciones se realicen hasta llegar al objetivo trazado, sino también a que en la práctica algunos detalles de la funcionalidad fueron modificados durante el desarrollo, por lo que sobre la marcha se realizaron ajustes e iteraciones adicionales a las esperadas.

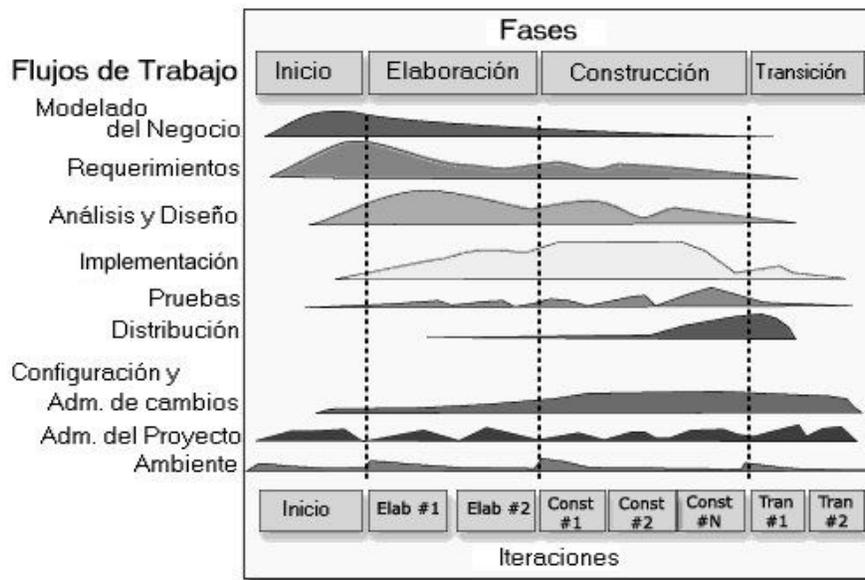


Fig. 3.1 Relación de flujos de trabajo con las Fases en el UP [JAVAHISPANO1].

Recolección de Información: Aunque esta actividad no parece como una actividad formal dentro de nuestros diagramas, es evidente que al principio de cada iteración la recolección de la Información toma un papel preponderante, e invariablemente se realiza a través de entrevistas a todos los involucrados (“stakeholders”), como lo indica la teoría RUP, en nuestro caso particular el resumen de esta información, es la que se especificó en el primer capítulo.

3.2 Modelado del negocio

Un modelo de negocio (también llamado diseño de negocio) es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Es un resumen de cómo una compañía planifica servir a sus clientes. Implica tanto el concepto de estrategia como el de implementación.[WIKIPEDIA1]

Generalmente, los modelos de negocio de las compañías de servicio son más complejos que las de fabricantes y vendedores, pero para el problema en particular podemos concentrarnos solo en aquellos elementos que están involucrados.

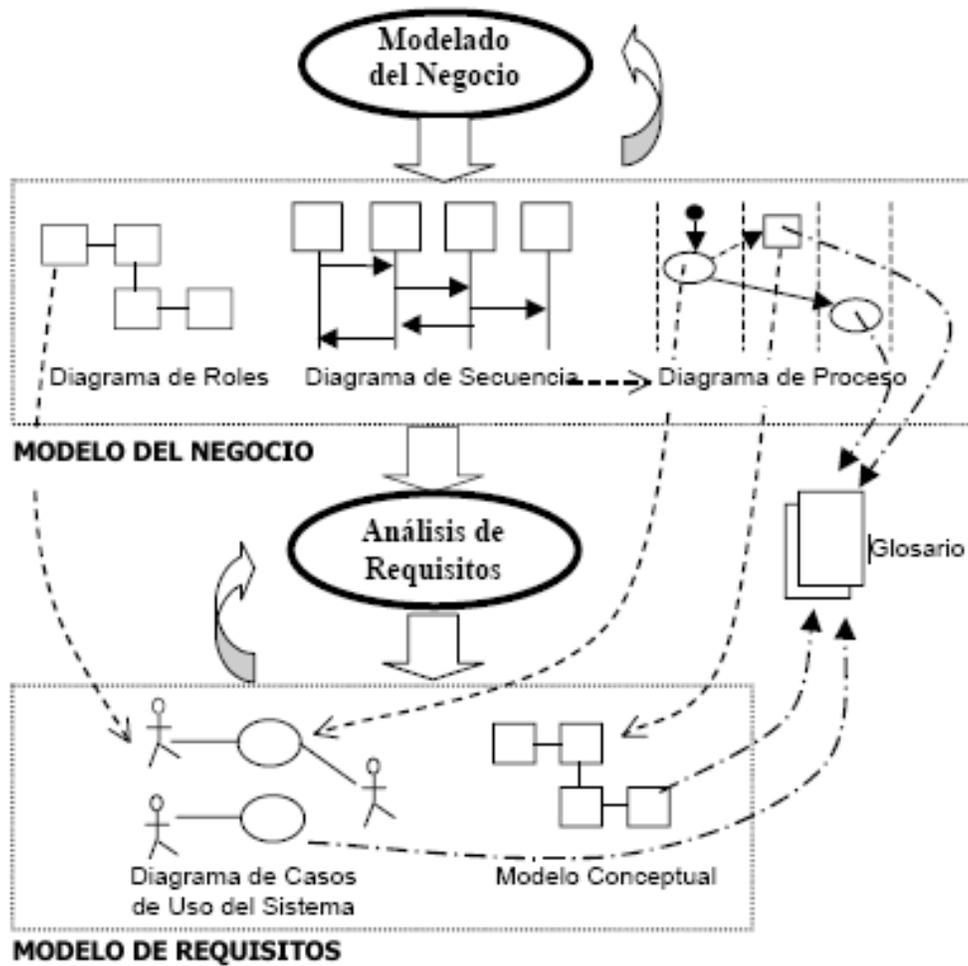


Fig. 3.2 Relación de trazabilidad entre los modelos de negocio y de requisitos [ORTIN1].

3.1.1 Acotaciones al modelo de negocios del IMP

Dadas las especificaciones descritas en la 1ra parte, se establece:

- El IMP trabaja en función de proyectos
- Los proyectos son relativamente independiente unos de otros.

Al resolver el problema para un proyecto se puede aplicar la misma solución a cualquier proyecto del IMP, en la solución a nivel proyecto, se deberá poder obtener información que pueda ser integrada con todos los proyectos del IMP.

Los diagramas de secuencia y de proceso realmente no son necesarios, debido a su simplicidad ya que solo comprenden las siguientes acciones:

- Personal Auxiliar Externo (PAE) llega al IMP.
- PAE reporta entrada con operador que registra entrada
- Al terminar el turno PAE se reporta con operador que registra salida
- En cualquier momento el operador genera reportes de los registros de E/S

También de la definición del problema vista en la primera parte de este documento, se definen los “actores” del negocio:

- Jefe de proyecto
- Operador del sistema CAPAE
- Personal Auxiliar Externo (PAE)
- Visitantes.

Una **instancia de un actor** es alguien o algo externo al sistema que interactúa con el sistema.

Interacción entre actores:

El jefe de proyecto: Determina (tiene el conocimiento exacto de) quienes son los elementos que componen su proyecto y establece el papel que tienen dentro del proyecto.

Operador del sistema: Cualquier empleado dentro del proyecto, que el jefe de proyecto designe como el empleado que realizará la captura manual de los registros de entrada/salida, pueden ser dos o más, esta actividad solo le tomará parte de su tiempo, se espera que el (o ellos) tenga(n) sus propias actividades además de efectuar los registros que hagan en el CAPAE.

Personal Auxiliar Externo(PAE):, aunque son los elementos que proporcionan la información (Entrada/Salida) no tendrán ninguna interacción directa, importante con el CAPAE, debido a que ellos solo se reportaran con los operadores del sistema para que a su vez, sean los que realicen las anotaciones de entrada/salida.

Visitantes: Los visitantes podrán realizar consultas de la información capturada, así como la de los reportes, de hecho el PAE, asumirá este “rol” para así poder consultar y verificar que el sistema realmente refleje su tiempo que permanece dentro de las instalaciones del IMP.

3.3 Requerimientos

- *Para establecer y mantener acuerdos con los usuarios y afectados en lo que debe hacer el sistema.*
- *Para proveer a los desarrolladores del sistema una mejor comprensión de los requisitos del sistema.*
- *Para definir los límites del sistema.*
- *Para proporcionar una base para estimar costos y tiempos de desarrollo del sistema.*
- *Para definir un usuario-interfaz para el sistema, centrándose en las necesidades y las metas de los usuarios*

Se especificarán los detalles de la funcionalidad del sistema, principalmente en lo que se refiere al software y detalles de los casos de uso.

3.3.1 Objetivo y desglose de requerimientos

Objetivo Principal del Sistema:

Registro de asistencia al IMP del Personal Auxiliar Externo (PAE) contratado bajo el régimen de “outsourcing” y obtención de reportes de estos registros.

Los Requerimientos principales son:

- Registrar la hora de entrada.
- Registrar la hora de salida.
- Modificar entrada/salida en excepciones.
- Reportes de horas trabajadas por día y por persona, para cada compañía por mes.
- Reportes de horas trabajadas por día y por persona, para cada compañía por semana.
- Reportes de asistencias del personal, día por día, por compañía por mes.

Requerimientos secundarios o auxiliares:

- Alta de compañía de PAE
- Alta de PAE
- Baja de PAE
- Alta de operador.
- Baja de operador.
- Consulta de bitácora.
- Ayuda (documentación)

3.3.2 Casos de uso.

Pueden ser definidos como:

Una secuencia de acciones, incluyendo variaciones, que el sistema puede ejecutar y que produce un resultado observable, de valor para un actor que interactúa con el sistema [ORTIN1]

Un elemento adicional que interviene en los casos de uso es la extensión (**extend**), la cual permite definir la relación que existe entre algunos casos de uso, de esta manera es posible extender la funcionalidad de los casos de uso, además de hacer el diseño a nivel de código más modular [GUIANDIN1].

Para la definición de los “Casos de uso”, requerimos establecer los actores o roles, y de la sección anterior se desprende que la interacción con el sistema estará limitada a los siguientes “**Actores**”:

1. **Administrador:** Para definir o establecer quienes son los operadores y compañías.
2. **Operador:** Los usuarios que alimentaran de datos al sistema.
3. **Invitado anónimo:** Solo pueden consultar los datos.

Los factores incidieron en el diseño de cada unos de los casos de uso:

- Funcionalidad esperada.
- Entrevistas efectuadas al personal involucrado
- Manejo del sistema a través de páginas web.

De este último factor se desprende que uno de los conceptos importantes a manejar es el de “**sesión**”, (se menciona en la descripción de casos de uso) y es **el tiempo que se define en el sitio web, para mantener validos algunos de los valores, que se establecen al interactuar con el sistema.**

La mecánica de funcionamiento del sistema es muy simple: Al entrar y salir del IMP, los elementos del PAE, se presentaran físicamente con un operador autorizado en el CAPAE para que este ultimo, abra la Página web del CAPAE y haga la anotación correspondiente en el sistema de la entrada o salida.

Para que el sistema identifique los permisos del ‘actor’ que usa el sistema, debe al inicio pedirle un nombre de usuario y contraseña.

El administrador determinará a través de comandos propios, los nombres y contraseñas de los operadores, así como los datos de las compañías del PAE.

El caso de uso: “Iniciar Sistema” (abrir la Página web del CAPAE) se requiere antes que cualquier otro.

En la descripción de los casos de uso (ver anexo) se usa como actor al “usuario”, aunque para poder ejecutar cada caso se requiere que se valide con uno de los roles autorizados.

En las figuras 3.3 a 3.7 se muestran los casos de usos del CAPAE y los actores que los ejecutan.

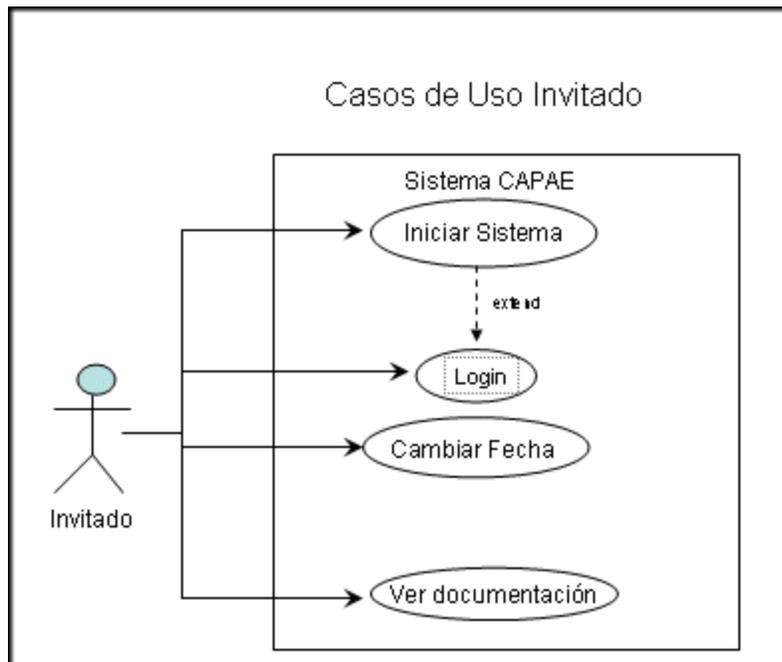


Fig. 3.3 Casos de Uso del Rol Invitado.

Extensión (extend): es una relación que amplía la funcionalidad de un Caso de Uso mediante la extensión de sus secuencias de acciones

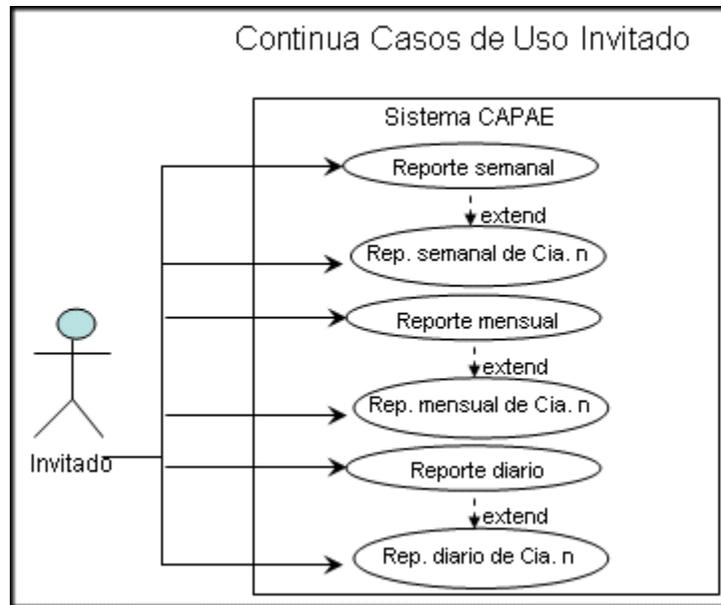


Fig. 3.4 Continuación Casos de Uso del Rol Invitado

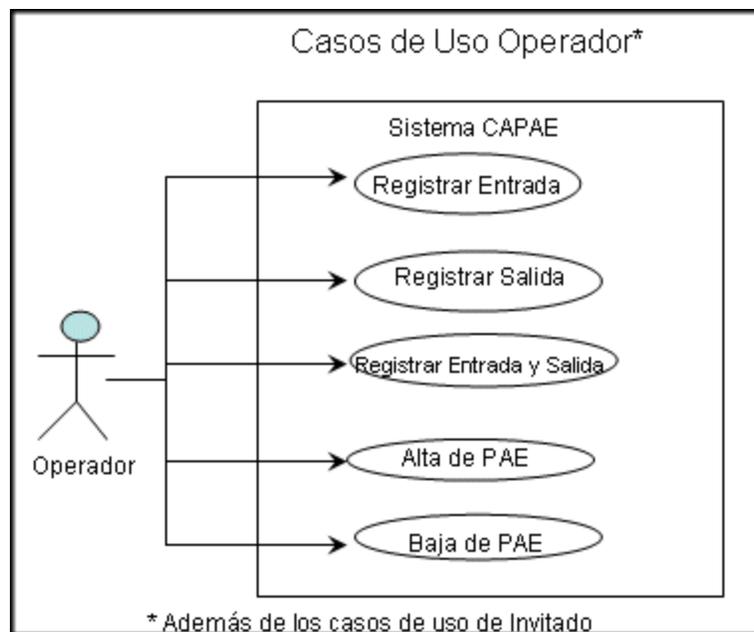


Fig. 3.5 Casos de uso del rol de Operador.

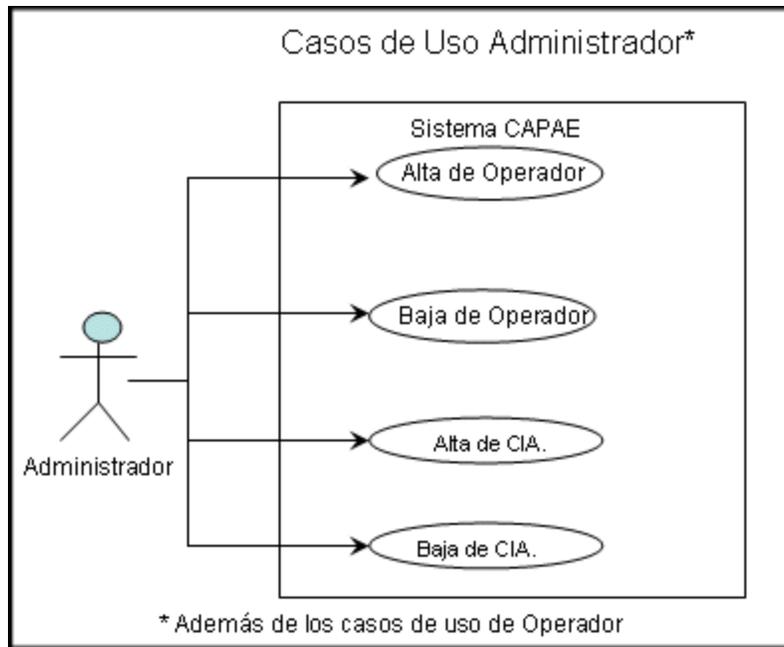


Fig. 3.6 Casos de uso del Rol Administrador.

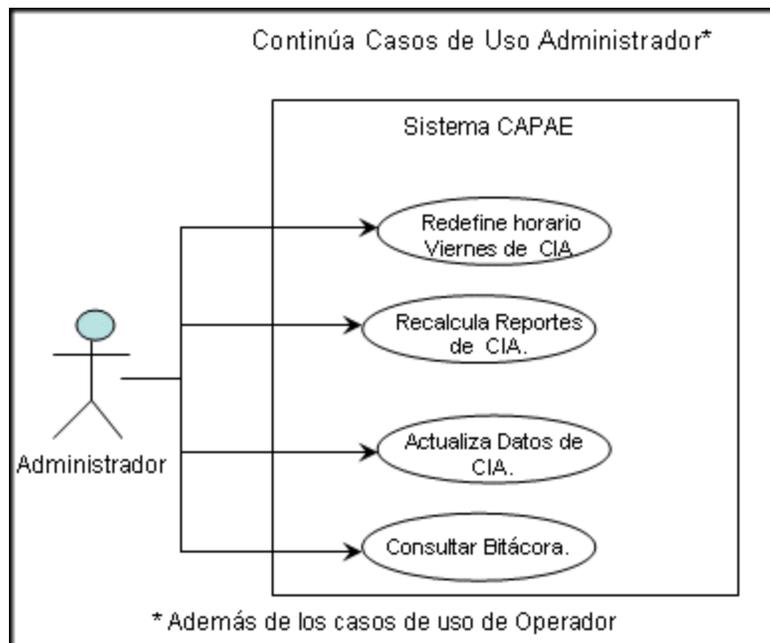


Fig. 3.7 Continuación casos de uso del rol de Administrador.

En la práctica se creó la definición y descripción de cada uno de los 24 “casos de uso”, requeridos para el CAPAE, en el Anexo I “Casos de Uso” mostramos la definición de algunos de ellos y los nombres de los no definidos en este documento.

3.3.3 Comandos a implementar

De los casos de uso, se concluye que se requiere que nuestro sistema maneje 19 comandos, agrupados por tipo quedan de la siguiente manera:

Comandos usuales de operación:

LOGIN	Identificarse como un operador o administrador
ENTRADA	Registrar entrada de un elemento del PAE
SALIDA	Registrar salida de un elemento del PAE
ENTSAL	Registrar simultáneamente entrada y salida

Comandos para reportes por compañía:

REPMES	Reporte de horas trabajadas por día y por persona, mensual
REPSEM	Reporte de horas trabajadas por día y por persona, semanal
REPDIA	Reporte de asistencias del personal, día por día, mensual

Comandos auxiliares de operación:

ALTA	Dar de alta a un elemento del PAE
BAJA	Dar de baja a un elemento del PAE
MODIFICA	Modifica datos de elemento PAE
AYUDA	Consultar documentación del sistema

Comandos de administración:

ELIMINA	Dar de baja del sistema a un operador
NUEVO	Dar de alta en el sistema a un operador
NUEVACIA	Dar de alta en el sistema a una compañía
ELIMINARCIA	Dar de baja en el sistema a una compañía
MODIFICACIA	Modifica una o más características de una compañía previamente dada de alta

CAPAE

E/SVIERNES Redefine horario de trabajo, para los viernes de una compañía previamente dada de alta

RECALCULA Recalcula horas trabajadas debido a cambios hechos en registros para una compañía en particular

BITACORA Mostrar registros de movimientos importantes

3.4 Análisis y diseño

A partir de los requerimientos (funcionalidad, casos de uso, comandos) se realizará la definición de modelos de análisis de diseño y de implementación del sistema.

Dentro de las mejores prácticas y recomendaciones para el desarrollo de sistemas, basados en objetos, está la de usar “Patrones de Diseño”.

Un patrón de diseño (Design Pattern) es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

3.4.1 Patrón de diseño.

Debido a las características de los requerimientos, se decidió modelar el sistema en base al patrón de diseño: “Modelo-Vista-Controlador”.

Éste es el clásico y el más famoso de todos los patrones de diseño, especialmente cuando hablamos de interfaces de usuario. Una interfaz de usuario es básicamente una capa de presentación de una cierta clase de “modelo de datos”. Si los datos cambian, la vista por consiguiente es actualizada, o viceversa. En este panorama particular, existen dos objetos implicados para realizar esta tarea, es decir, la “vista” y un “modelo”. Podría haber más de una vista del modelo de datos, una vista podría ser una tabla de datos, y otro podría ser un diagrama gráfico de algún tipo, como una grafica de barras, o grafica circular, así que para poder seleccionar la “vista” se requiere un “controlador”.

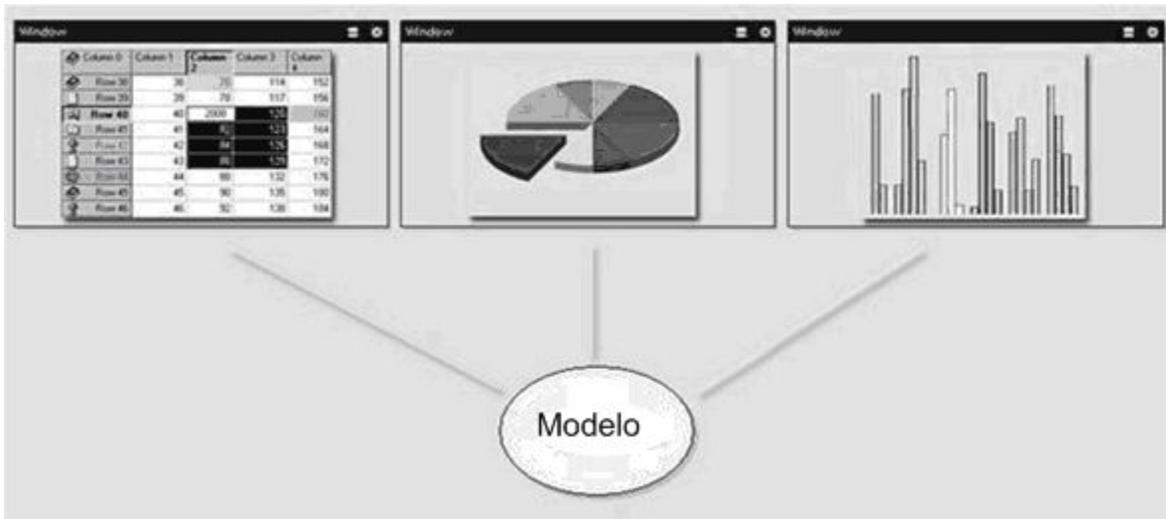


Fig. 3.8 Vistas y modelo [MUKHTAR1].

Análisis de robustez: Este patrón de diseño (Modelo-Vista-Controlador) se puede describir mejor, usando el análisis de robustez, [JACOBSON1], y [ROSENBERG1]). El Análisis de robustez implica analizar el texto narrativo de los casos del uso, identificar inicialmente los objetos que participan en esos casos del uso, y clasificar estos objetos basados en los papeles ellos juegan, esto ayudará a establecer los objetos dentro del paradigma del Modelo-Vista-Controlador. Ivar Jacobson los ha clasificado como:

Objetos “vista” (o interfaz en algunas traducciones) representan enlaces de comunicaciones, entre el sistema y el ambiente. Son los objetos con los cuales los actores (usuarios) se comunicarán al sistema (ventanas, pantallas, cuadros de diálogos, menús, etc.).

Objetos "control" (controladores) representan objetos específicos del ambiente de los casos de uso. Estos son los objetos del negocio, aquí es donde se establecen las reglas del negocio, (Ej: Para filtrar los datos a ser presentados, o validar los que se capturan).

Objetos “entidad” representa los objetos de vida larga, se enfocan principalmente en los “estados” del sistema. No son otra cosa que la información o datos que sus objetos “interfaz” están “buscando” (bases de datos, archivos excel, datos de sesión, etc.).

Reglas del juego: Estas son las cuatro reglas básicas que se deben seguir cuando se definan los objetos en el análisis de los casos de uso:

1. Los actores solo pueden comunicarse con los objetos “vista”.
2. Los objetos “vista” solo pueden comunicarse con objetos control y actores.
3. Los objetos control pueden comunicarse con objetos “vista” y objetos “entidad” y a otros objetos control, pero no a los actores.
4. Los objetos “entidad” solo pueden comunicarse con objetos control.

Dentro del paradigma Modelo-Vista-Controlador (MVC) los objetos mencionados son considerados (o renombrado) de la siguiente manera:

El objeto “Frontera” se considera como el objeto “**vista**”.

El objeto “**Control**” mantiene el nombre en los dos casos.

El objeto “Entidad” se considera como el objeto “**modelo**”.

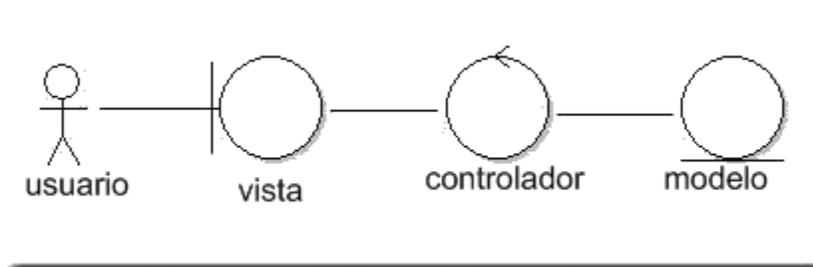


fig. 3.9 Iconos UML par el Modelo Vista Controlador [MUKHTAR1].

A continuación el diagrama de secuencia simplista e hipotético para el MVC. En él un usuario inicia una pregunta, un evento es generado el cual es manejado por el controlador que consigue la información del modelo, valida la información y la trasmite de regreso a la vista.

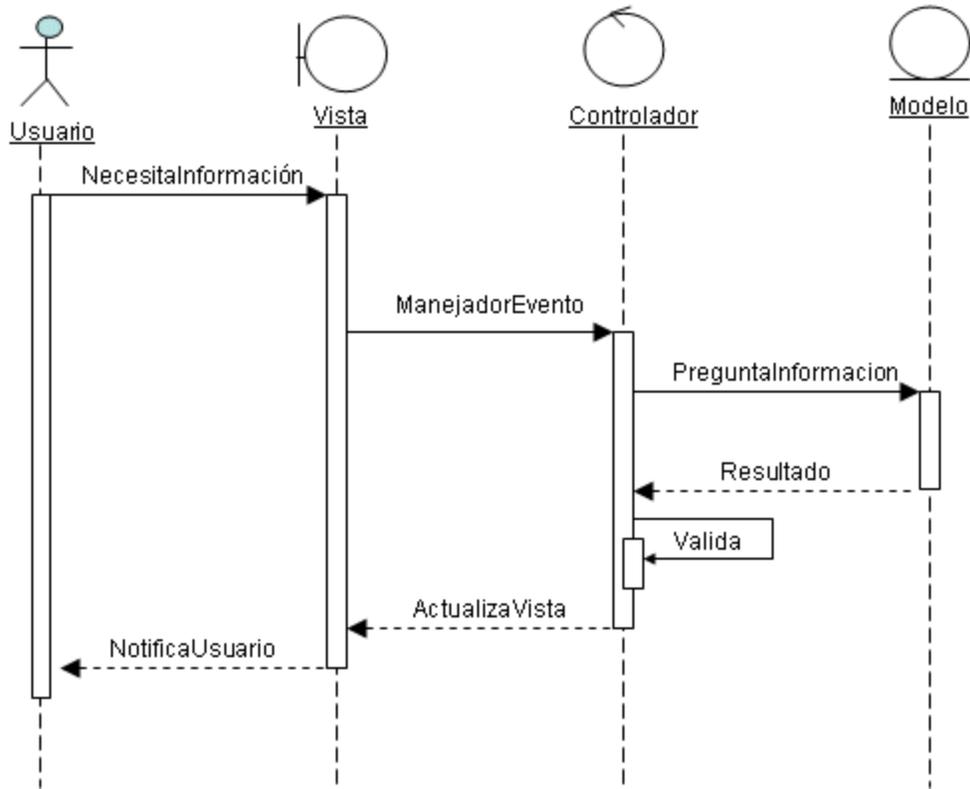


Fig. 3.10 Modelo Vista Controlador (MVC)

En este punto el siguiente paso es definir (y programar) los componentes (u objetos) que satisfagan los requerimientos hasta ahora definidos, pero la experiencia nos indica que podemos hacer uso de ciertos componentes previamente diseñados para otros trabajos similares y de esta manera acelerar nuestro desarrollo.

3.4.2 Páginas maestras

La “reusabilidad” es un concepto que también podemos aplicar al mismo código, por lo que para el desarrollo de las páginas web, nos apoyaremos en el desarrollo de un “marco de trabajo” de una página maestra extensible, para ASP.NET 1.1, de: Shams Mukhtar [MUKHTAR1], el cual aplicó el paradigma mencionado, en el se buscó que las páginas de una aplicación además de mantener, el mismo estilo, puedan ser modificadas de acuerdo a las necesidades particulares.

El trabajo mencionado se puede visualizar en el diagrama mostrado en la Fig. 3.11:

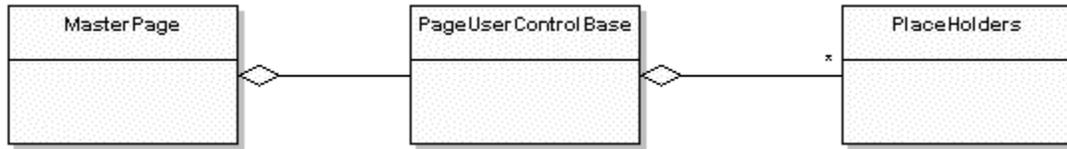


Fig. 3.11 Esquema de las clases necesarias para usar “Páginas Maestras” [MUKHTARI].

Al usar este marco de trabajo, las páginas (.NET) de la aplicación se basan en, la clase “Master Page” (MasterPageBase), la cual usa un “control de usuario” basado en PageUserControlBase, este “control de usuario” contiene una serie de propiedades: “PlaceHolders”, y cada “Place Holder” a su vez se implementa también con “controles de usuario”, los cuales son los que realmente definen el contenido y funcionalidad de las diferentes secciones de la página, un esquema más general se muestra en la Fig. 3.12.

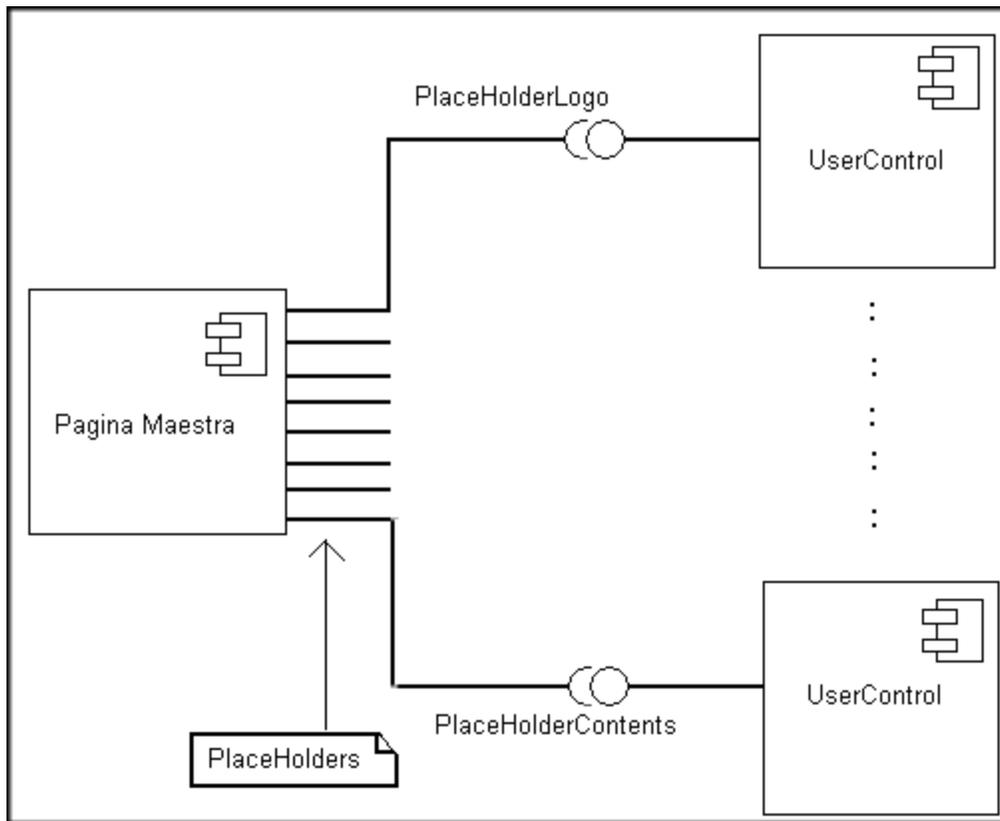


Fig. 3.12 Diagrama de la relación entre las Páginas Maestra y los “Controles de usuario”.

Las secciones originales establecidas en el artículo mencionado son:

- Logotipo (PlaceholderLogo).

- Encabezado (PlaceHolderHeader).
- Menú (PlaceHolderMenu).
- Menú secundario (PlaceHolderSubMenu).
- Navegación (PlaceHolderNavigator).
- Contenido (PlaceHolderContents).
- Contador (PlaceHolderSiteCounter).
- Pie de Página (PlaceHolderFooter)

Con este enfoque cada uno de los “Place Holders” es “instanciado” a tiempo de ejecución por los objetos que son “Controles de usuario” (objetos UserControl de .NET).

3.3.3 Prototipo general

Para nuestro desarrollo particular podemos partir de un prototipo general, en la figura 3.13 se observa el prototipo, y en el se indica la posición que guardarán en la pantalla las diferentes secciones.

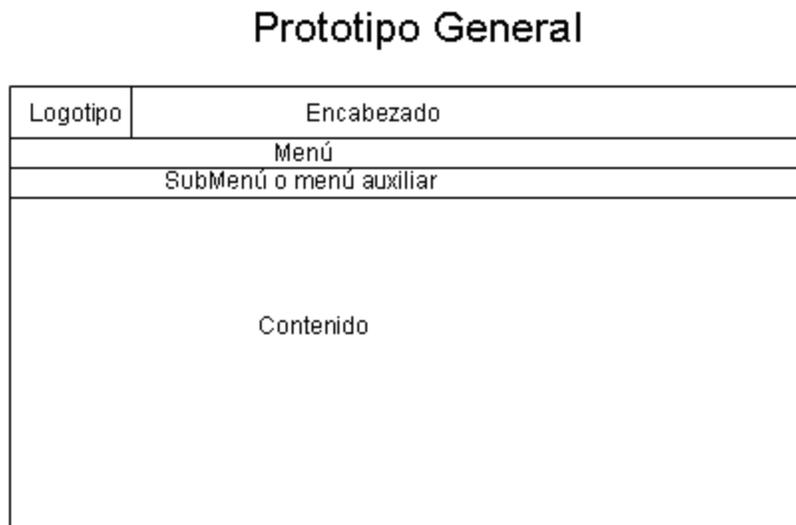


Fig. 3.13 Prototipo General

Como cada sección corresponde a un “control de usuario”, para nuestras páginas web, se diseñaron y crearon los siguientes controles de usuario:

Logotipo: Con la imagen de la empresa.

Encabezado: Texto con datos del proyecto y aplicación.

Menú: Con los comandos (ligas) para que el usuario los use en cualquier momento.

Menú Secundario: Para mostrar usuario activo, la hora del sistema, y permitir la selección de elemento PAE y compañía.

Contenido: Para mostrar los controles con los que interactúa el usuario.

En principio se considero crear una página (maestra) para cada comando (elemento del menú) y los cuatro primeros controles se usarían en todas las páginas, pero tendrían un “Contenido” diferente, que dependería del comando que se ejecute.

3.3.4 Prototipos y diagramas de secuencia

El siguiente paso en nuestro análisis consistió en determinar los prototipos y diagramas de secuencia.

Los **prototipos** nos auxilian a determinar exactamente cuales son los componentes visuales (controles de usuario) que se usaran para cada “contenido” de cada comando, como cada control de usuario se compone de uno o mas controles .NET, aquí es evidente que en esta etapa se decidió cuales son cada uno de esos controles a usar.

Los **diagramas** de secuencia nos permiten establecer cuales son las acciones que se realizaran en el tiempo y relacionando los elementos (u objetos) que intervienen en una determinada situación o comando.

Por ejemplo para la pantalla inicial tenemos el prototipo para el control (de usuario) del contenido en la Fig. 3.14:

≤ Septiembre de 2006 ≥							(17) EN: Viernes, 22 de Septiembre de 2006		
D	L	M	M	J	V	S	Cia-Nombre	Entrada	Salida
27	28	29	30	31	1	2	SERVINEXT-HECTOR MANUEL GARCIA MORALES	08:19 a.m.	
3	4	5	6	7	8	9	SERVINEXT-ALEJANDRO RAMIREZ TAPIA	08:02 a.m.	
10	11	12	13	14	15	16	SERVINEXT-MAURICIO MORENO VILLEGAS	08:13 a.m.	
17	18	19	20	21	22	23	SERVINEXT-ENRIQUE GONZÁLEZ ESTÉBANEZ	08:03 a.m.	
24	25	26	27	28	29	30	CEPRA-WENDY PAULIN MARTINEZ	08:43 a.m.	
1	2	3	4	5	6	7	CEPRA-ISRAEL VILLA MORALES	08:58 a.m.	
							CEPRA-JAVIER CEDILLO ITURIBE	08:18 a.m.	
							CEPRA-GUILLERMO CASTRO BARBEYTO	08:32 a.m.	
							LANWARE-FERNANDO GALLARDO ALONSO		
							LANWARE-ROBERTO ESPINOSA		
							LANWARE-YOLANDA PADILLA ANGELES	08:12 a.m.	05:14 p.m.
							SERVINEXT-ROBERTO HERNÁNDEZ TORRES	08:13 a.m.	
							CEPRA-GUADALUPE IVÁN VELÁSQUEZ SÁMANO	08:28 a.m.	
							UVM-AVRIL PEREZ MUÑOZ	10:02 a.m.	
							SERVINEXT-PEDRO ARTURO GARCÍA CERDA	08:03 a.m.	
							SERVINEXT-LUIS ANTONIO HERNÁNDEZ GARCÍA	08:13 a.m.	
							POLI-OLMO CASTANEIRA RIOS	09:16 a.m.	01:59 p.m.

Fig. 3.14 Prototipo de Contenido en Pantalla Inicial

Los controles .NET, que componen el **control de usuario contenido** para la pantalla inicial::

- **Calendario:** Para que el usuario indique que fecha de trabajo desea usar.
- **Texto con Fecha de Trabajo** (y Cantidad de elementos PAE registrados).
- **Tabla** con Nombres y los registros capturados en la fecha de trabajo

Considerando al control de usuario esbozado anteriormente y usando el patrón de diseño “Modelo-Vista-Controlador”, el diagrama de secuencia asociado que corresponde a la acción apertura de pantalla inicial se muestra en la Fig. 3.15.

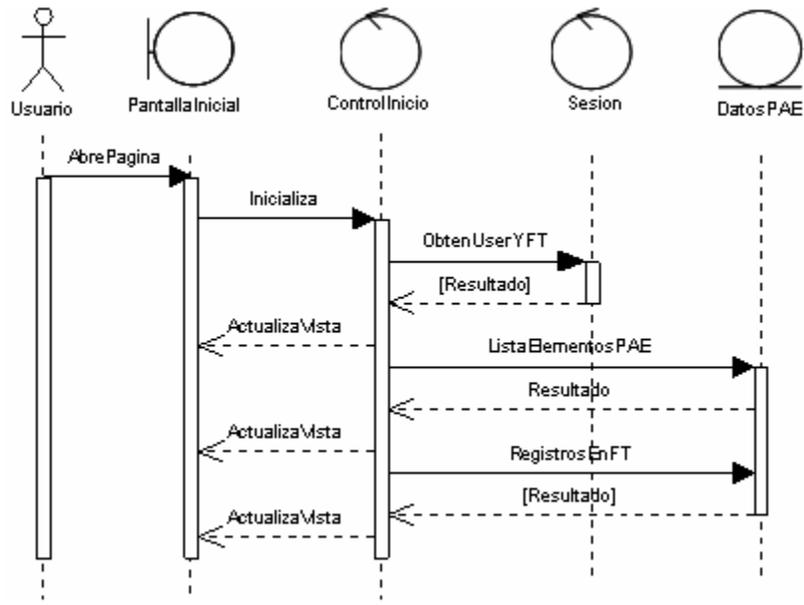


Fig. 3.15 Diagrama de Secuencia de Página inicial.

En el anexo II puede consultar los "Prototipos y diagrama de secuencias", correspondientes a los casos de uso.

3.3.5 Objetos principales

Elementos a desarrollar para nuestra aplicación.

De los diagramas de secuencia anterior análisis se observa que los objetos principales a desarrollar para nuestro sistema son:

Pantallas: Se refiere desde el punto de vista operacional a las formas donde se ubican los “**controles de usuario .NET**” (CUN), estas pantallas son “instancias” de las páginas maestras, las cuales a su vez se derivan de clases nativas de ASP.NET²

Una manera simplista de visualizar cada uno de los elementos que conforman nuestras formas es:

² Herramienta de desarrollo web comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

- Cada **forma** contiene uno o más **CUN**
- Un CUN, puede contener: cero, uno o mas “**controles .NET**”.
- Un **control .NET**, es un elemento simple (etiqueta, caja de texto, etc.) al cual se le puede añadir código.

Control: Objeto que será común a toda la aplicación, se deriva directamente de la arquitectura seleccionada. Maneja toda la lógica de la aplicación siendo manejado principalmente desde el código alojado en los “controles .NET”, para responder a la interacción del usuario (en la implementación son “instancias” de la clase: ClsControl.cs)

Sesión: Este objeto cobra especial importancia su uso es imprescindible, ya que nuestra aplicación será creada para el ambiente de una intranet, y solo a través de este objeto podemos “guardar datos”, durante toda la sesión del usuario (sin ser una base de datos), al igual que los “controles .NET” es un objeto (Session) disponible en forma nativa en ASP.NET

DatosPAE: Para el manejo de todas las operaciones relativas a la consulta y actualización de los datos del PAE (en la implementación son “instancias” de la clase ClsDatosPAE.cs.)

Una vez definidos los elementos más importantes de nuestro diseño y análisis se decidió iniciar implementación como se describe en el siguiente capítulo.

Capítulo 4. De la implementación a la distribución

4.1 Implementación

Uno de los grandes dilemas en el desarrollo de los sistemas es:

¿A que nivel de detalle tengo que llegar en el diseño para empezar a codificar?,

es sabido que generalmente se está ansiosos por iniciar la programación y es importante elegir el punto adecuado, también se debe considerar que es posible dedicar demasiado tiempo a las especificaciones del diseño haciendo diagramas (y documentación) de todos los detalles que conforman nuestra aplicación, cuando en realidad algunos de ellos podrían omitirse.

Aun los especialistas en el tema no se ponen de acuerdo ya que es común encontrar autores que establecen que se debe diseñar todo el sistema y aun el más mínimo detalle y que en un futuro cercano serán posibles los sistemas que alimentándose de estas especificaciones puedan generar automáticamente otros sistemas [WEBOMG1], mientras que otros difieren grandemente indicando que está muy lejano el día que podamos hacer esto. [KIMMEL1]

La regla que seguimos en este trabajo, es establecer nuestra documentación a un nivel bastante general y solo detallar aquellas partes que consideramos más complejas.

Teniendo lo anterior en mente decidimos iniciar la implementación creando un prototipo el cual inicialmente siguió fielmente los diseños indicados en el anterior capítulo, además este prototipo gradualmente se convertiría en el diseño final y permitiría realizar las pruebas pantalla por pantalla.

4.1.1 Refinación del diseño en la implementación

Al crear el prototipo, y realizar el análisis bajo esta nueva perspectiva, surgieron otros elementos que modificaron algunos aspectos del diseño que se tenía hasta el momento, pero siempre respetando las acciones indicadas en el diagrama de secuencia.

Modificaciones a Comandos

- **“recalcula” se elimina del menú:** Esto se detecto ya que esta "acción" solo se requiere efectuar, cuando se modifica el horario de una compañía, por lo que aunque se implementa, se ejecuta internamente en el caso indicado.
- **“inicio” se añade en el menú:** Cuando otro comando se ejecuta es posible que muestre un mensaje y el usuario desea volver a ver la pantalla de inicio, o en aquellos casos en que se arrepienta al iniciar otro comando y este en proceso de confirmación.

Agrupación de “controles .NET” y esquema general.

Como el diseño se baso en las librerías de Shams Mukhtar, inicialmente se debería crear “Páginas Maestras”, con una página por comando y en cada página en la sección de contenido el “Control de Usuario .NET” (CUN), que seria el que manejaría la ejecución del comando.

Pero al observar la similitud de algunos comandos en el uso de “**controles .NET**”, se decidió donde fuera posible agrupar en los CUN, mas “controles .NET”, de los inicialmente contemplados para que una misma página (y un solo CUN) sirviera para dos o más comandos, de esta manera se evitó el manejo de una mayor cantidad de páginas, controles y código.

Evidentemente para poder manejar esta agrupación de comandos, a cada CUN se le añadió el código necesario, para que al pedirse la ejecución de un comando:

- Inicialice algunos valores que, identifiquen el comando, muestre o esconda “controles .NET” de acuerdo al comando.
- En algunos comandos muestra botón para confirmar y disparar ejecución..
- Al ejecutarse: considere el comando seleccionado.

Lo anterior se muestra en el diagrama de secuencias de la Fig. 4.1.

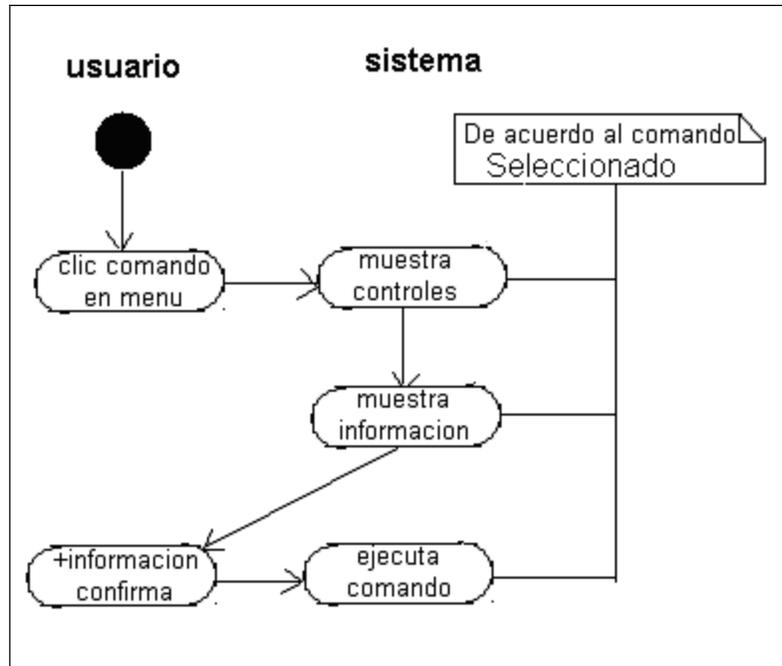


Fig. 4.1 Diagrama de actividades para la “Agrupación de controles”.

A continuación el esquema que siguen la mayoría de las páginas de la aplicación, en el se ejemplifican los comandos: “**entrada**”, “**salida**” y “**entsal**”.

Para los comandos mencionados, se creó una sola página web (**registro.aspx**), con un control de usuario (**registro.ascx**), este control de usuario, (ver abajo Fig. 4.2) contiene varios controles .NET:

- Un Calendario con nombre interno: *Calendario1*
- Una Etiqueta con nombre interno: *lblmensaje*
- Un Panel con nombre interno: *Panel1* conteniendo:
 - Texto: “**Entro**” con tres listas para selección de hora de entrada
 - Texto: “**Salio**” con tres listas para selección de hora de salida
- Botón: con texto “**Aceptar**”

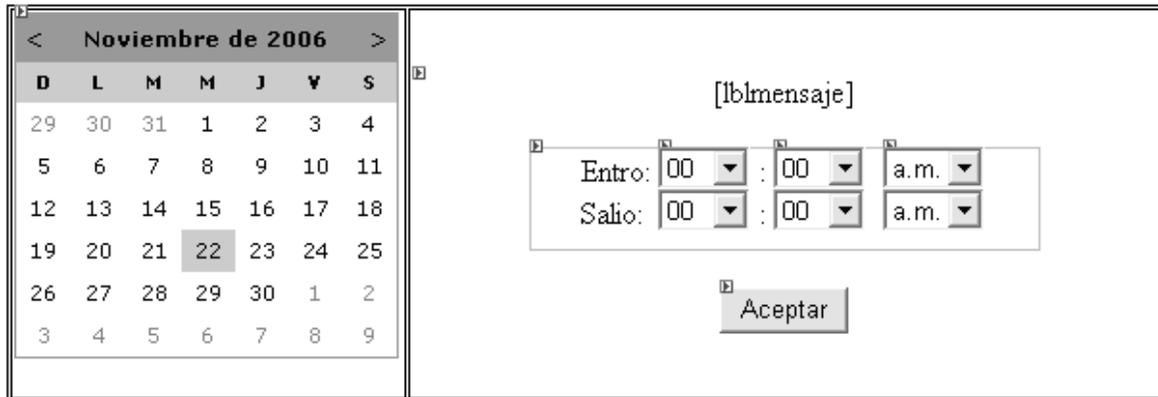


Fig. 4.2 Controles para tres comandos: entrada, salida y entsal.

Al ejecutarse cualquiera de los tres comandos mencionados, en la práctica el menú hace un direccionamiento a la página:

<http://intranetimp/capae/registro.aspx?comando=nombrecomando>

Donde **nombrecomando** es el comando (entrada, salida o entsal) a ejecutar.

En el código del control de usuario (Ver en Anexo IV Código relevante: *Registro.ascx.cs*), al abrirse la página, se guarda el **nombrecomando** en la “sesión” a través de el objeto “control,” y se ejecutan dos procedimientos que ejecutan y que actúan según el comando en ejecución:

- “**actualizaControles**”: Inicializa los controles
- “**muestraControles**”: Llamada desde “actualizaControles”, Muestra/Esconde controles

Por Ejemplo: Dependiendo del comando se cambia el texto de *lblmensaje*, y solo en el caso del comando **entsal** se pone visible el control *panel1*, para que el usuario edite los valores de la hora de entrada y la hora de salida, en cualquier otro caso la hora se toma del sistema, y finalmente el texto del botón se cambia para que refleje el comando en ejecución.

Ya en ejecución el aspecto con el comando **entsal** es el de la Fig. 4.3

hilario **02:50 p.m.** PAE: WENDY PAULIN MARTINEZ

Febrero de 2007						
D	L	M	M	J	V	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	1	2	3
4	5	6	7	8	9	10

Viernes, 23 de Febrero de 2007
WENDY PAULIN MARTINEZ
 CIA: CEPRA
 Indique Hora de Entrada, Hora de Salida.
 Y oprima boton Aceptar.

Entro: 08 : 30 a.m.
 Salio: 02 : 00 p.m.

Fig. 4.3 Secciones MenuAuxiliar y Contenido al Ejecutar comando entSal .

Al oprimir el usuario el botón con el texto “Aceptar Entrada y Salida” el código correspondiente asociado al botón (*btnAceptar_Click*):

- Obtiene el nombre del **comando** del objeto control
- Ejecuta el código de acuerdo al comando

En todo momento se respetan las acciones indicadas en el diagrama de secuencia, pero debido al agrupamiento de controles también se agruparon procedimientos y en consecuencia los nombres de los procedimientos fueron cambiados para reflejar esta agrupación.

Como se indicó anteriormente al crear el prototipo se inicio la fase de programación de cada uno de los elementos previamente definidos, el resultado final del trabajo de programación es el que a continuación se describe.

4.1.2 Relación entre páginas web, controles y comandos

En base a los comandos que se implementaron, y su agrupación en controles de usuarios, se crearon las páginas y controles mostrados en la tabla 4.1, en ella se muestra³ la relación de

³ Se usan nombres de los archivos creados, para el ambiente de desarrollo la extensión del archivo indica el tipo de contenido “.cs”: código de una clase, “.aspx” pagina web, “.acsx”: Control de Usuario .NET

las páginas que su funcionalidad depende de un control de usuario, y el (los) comando(s), que la abren

Página	Control de Usuario .NET	Comandos
Default.aspx	MasterContentsControl.ascx	Inicio
login.aspx	login.ascx	Login
Registro.aspx	registro.ascx	entrada, salida, entSal
reportes.aspx	Reportes.ascx	repSem, repMes, repDia
Actpers.aspx	actpers.ascx	alta, baja, modifica, bajaOp, altaOp
admCia.aspx	admcia.ascx	altaCia, bajaCia, modCia, vieCia, bitacora

Tabla 4.1 Relación de Controles y comandos por página.

4.1.3 Código que asocia las páginas maestras

Un elemento primordial en nuestra implementación fue el control **MasterPageUserControl**, [MUKHTAR1] ya que en el se define el aspecto de la aplicación en la pantalla considerando la relación que existirá a tiempo de ejecución, entre cada uno de los “controles de usuario” que se carguen en las páginas maestras que usemos.

En nuestro desarrollo fue copiado, y se uso tal como fue desarrollado por [MUKHTAR1], su aspecto a tiempo de diseño en el ambiente de desarrollo de “Visual Studio 2003” es el mostrado en la Fig. 4.4.

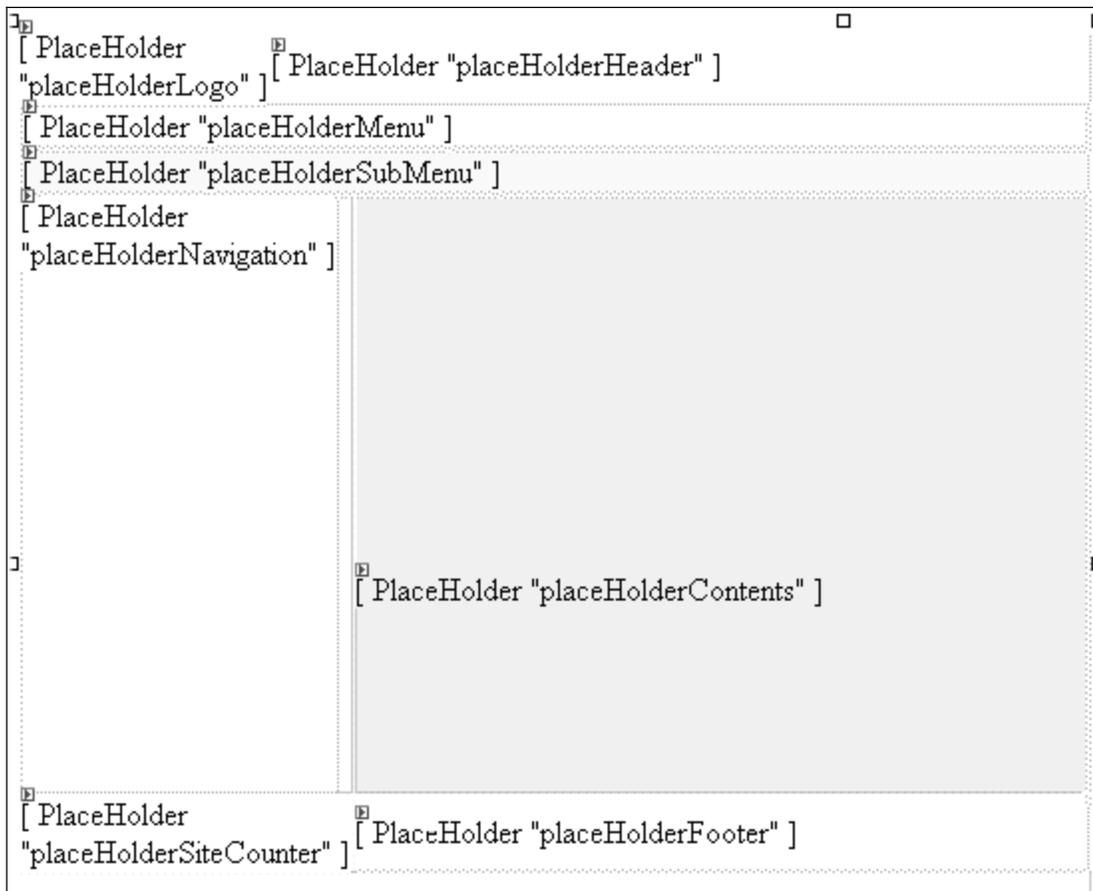


Fig. 4.4 Aspecto en ambiente de desarrollo de MasterPageUserControl.ascx.

Obsérvese que se tienen todas las secciones originalmente definidas por el autor, pero en la pantalla a tiempo de ejecución solo aparecerán las que se carguen en el control, a través de las páginas maestras, su nombre se debe especificar en un archivo de configuración (ver al final de este capítulo).

El código respectivo de “MasterPageUserControl.ascx” se puede consultar en el anexo IV “Código relevante”.

Para que cada página cargara sus controles de usuario, fue necesario crear una clase “Master” (c#) para cada una de las páginas (ver tabla 4.2), estas clases fueron basadas en la clase *Shams.Web.UI.MasterPages.MasterPageBase* de las librerías de [MUKHTAR1], que hereda el control mencionado MasterPageUserControl.ascx (el cual fue personalizado para la aplicación)

Página	Basada en clase
Default.aspx	MasterPage.cs
login.aspx	LoginMaster.cs
registro.aspx	registromaster.cs
reportes.aspx	reportemaster.cs
Actpers.aspx	actpersmaster.cs
admCia.aspx	admCiaMaster.cs

Tabla 4.2 Relación de Páginas con Clases “Master Page”.

Ejemplo de clase “Master Page” creada para la aplicación.

Como lo indica la tabla 4.3, la clase “registromaster” fue creada para la página registro.aspx, las primeras líneas son mostradas en la figura 4.5 (código completo en el anexo “Código relevante”):

```

public class registromaster : Shams.Web.UI.MasterPages.MasterPageBase
{
    public registromaster()
    {
        base.pageController.PageTitle = "CAPAE-Registro de Asistencia";
    }
    override public void AddPlaceholderControls()
    {
        ClearPlaceholderControls();
        base.AddPlaceholderControls();
    }
    override public void AddPlaceholderHeader()
    {
        System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/MasterHeaderControl.ascx");
        base.userPageControl.PlaceHolderHeader.Controls.Add(userControl);
    }
}

```

Fig. 4.5 primeras líneas de registromaster.cs,

Es fácil distinguir en el código, en la primera línea usa de base la clase: “Shams.Web.UI.MasterPages.MasterPageBase”, en línea cuatro pone el título a la página: “CAPAE-Registro de Asistencia”

En el procedimiento “AddPlaceholderHeader”, se carga el CUN contenido en el archivo: *MasterHeaderControl.ascx*, creado para el encabezado de la página, obsérvese que el

control mencionado, pertenece al grupo de controles que fueron creados para que se usen en casi todas las páginas y que se muestran en la tabla 4.3

Controles de usuario .NET	Descripción
menú.ascx	Menú principal con los comandos (enlaces) de la aplicación
menuaux.ascx	Menú auxiliar con controles .NET auxiliares
MasterHeaderControl.ascx	Imagen de encabezado y un texto para individualizar el proyecto.
MasterLogoControl.ascx	Solo contiene la Imagen Logo de la empresa

Tabla 4.3 Controles de usuario .NET (CUN) que se repiten.

De la misma manera en el resto del código (véase Fig. 4.5), lo único que se hace, es cargar en otros “PlaceHolder”, los controles que fueron creados para la aplicación (código completo de esta clase en Anexo “Código relevante” *registromaster.cs*), particularmente importante “AddPlaceHolderContents”, ya que carga el control de usuario *registro.ascx*, creado para el contenido.

4.1.4 Objetos principales y adicionales

En nuestro diseño (ver diagramas de secuencias del capítulo anterior) se indico que se manejan principalmente tres objetos:

- **Control:** Maneja toda la lógica de la aplicación siendo usada principalmente por el código en los controles .NET, para responder a la interacción del usuario (instancia de la clase: **ClsControl**)
- **Datos:** En principio es la clase que realiza todo el manejo de datos (instancia de la clase: **ClsDatosPAE**)
- **Session:** Esta clase tiene como objeto poder mantener datos durante la sesión, este objeto, se genera automáticamente y forma parte de las librerías de .NET

Para hacer mas modular y manejable el código de nuestro desarrollo se implementaron los siguientes objetos adicionales:

- **Bdxml:** Clase bdxml, que al instanciarse se usa para manejar todo lo referente a la creación y actualización de los archivos mensuales con formato xml, de esta manera si posteriormente se desea cambiar el esquema de datos de mensual y/o a una base más robusta solo se tendrá que sustituir esta clase, como es de esperarse, solo es usada por la clase **ClxDatosPAE**
- **ClxRegistro:** Maneja la bitácora, esta clase es parte de las librerías personales que usamos en varias aplicaciones, solo es usada por: **ClxControl**, su código fue alojado en el mismo archivo (**ClxControl.cs**).
- **RepSemMes:** La función de esta clase es auxiliar en la generación de reportes, solo es usada por: **ClxControl**, su código se creo en el mismo archivo (**ClxControl.cs**).
- **barGraph:** Control que permite crear una grafica simple de barras, auxiliar para los reportes, parte de las librerías personales que usamos en varias aplicaciones, en la práctica solo crea una tabla html con renglones de diferentes longitudes y colores.

Los “paquetes” o “Espacios de Nombres” que comprenden nuestra aplicación WEB son los mostrados en la Fig. 4.5.



Fig. 4.6 Paquetes

Shams: Contiene las clases bases usadas como base para las páginas maestras y los controles de usuario [MUKHTAR1].

Capae4: Contiene las clases desarrolladas en este trabajo.



Fig. 4.7 Clases creadas para el sistema.

4.1.5 Clases auxiliares

Debido a su simplicidad (para el programador del “framework.NET”) o por ser nativas del ambiente, explicamos muy brevemente algunas de las clases auxiliares:

- **Global:** Objeto nativo de .NET que permite añadir código para los eventos de inicio de aplicación e inicio de sesión.
- **MasterHeaderControl:** CUN, que muestra el encabezado que aparece en todas las páginas, en realidad es solo una tabla, con una imagen de fondo y dos renglones, en el segundo renglón se creó una etiqueta *lbltitulo*, al momento de cargarse las páginas en el evento *Page_Load*, se copia a la etiqueta mencionada un texto definido previamente en la configuración de la aplicación (`Application["Proyecto"].ToString()`), inicializado en evento de inicio de aplicación de Global.

- **MasterLogoControl:** CUN, muestra el logotipo que aparece en todas las páginas, solo contiene una tabla con una imagen.
- **menu:** CUN, contiene una tabla HTML con los enlaces a los comandos.
- **menuaux:** CUN, contiene los controles .NET para selección de usuario y de compañía, que serán afectados por los comandos que se ejecuten.

4.1.6 Clases derivadas de **Shams.Web.UI.MasterPages.MasterPageBase**.

Las siguientes clases son similares y su único objetivo es cargar los controles de usuario en sus respectivos “placeholders”, sirven de base para las clases que se definieron, para las diferentes páginas de la aplicación, como se indica en la siguiente lista.

- **MasterPage:** Base de *_Default*, clase que se define en la Página Inicial (default.aspx)
- **Actpersmaster:** Base de *actpers*, clase que define en la página de comandos que actualizan datos del personal (actpers.aspx).
- **loginMaster:** Base de *login*, clase que se define en la página del comando login (login.aspx).
- **registroMaster:** Base de *registro*, clase que se define en la página de comandos que manejan la captura de E/S (registro.aspx).
- **admCiaMaster:** Base de *admCia*, clase que se define en la página de comandos que manejan la actualización de datos de las compañías (registro.aspx).
- **MasterPageUserControl.ascx:** basado en Shams.Web.UI.MasterPages.PageUserControl, control de usuario muy importante, se personaliza de acuerdo a la aplicación, y en el se definen los “Placeholders” y su posición, para que las páginas maestras (basadas en Shams.Web.UI.MasterPages.MasterPageBase) puedan cargar los controles de usuario cada vez que se carga una página maestra, en consecuencia es muy

importante que se defina en la aplicación a través la configuración en el archivo **myproy.config**, como se indica en la sección 4.3.

4.1.7 Manejo de los datos

Para el manejo de los datos se implementó el manejo de archivos mensuales, los cuales son creados y almacenados, por la misma aplicación, en un subdirectorío (bajo el directorío de la aplicación web) llamado “datos”. En el nombre del archivo se indica el mes y año de los datos con el formato general para nombre de los archivos: **yyyyymm.xml**.

Donde:

- **yyyy**: es el año de los datos.
- **mm**: numero del mes de los datos.

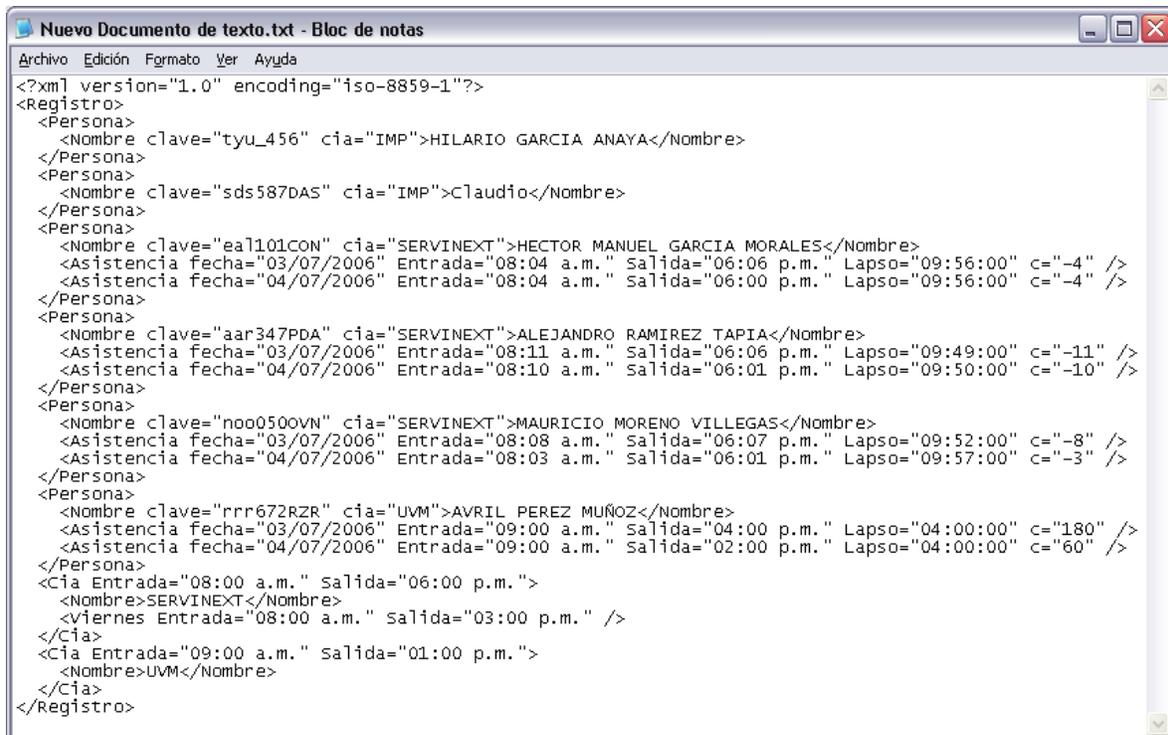
Ejemplo: “**c200607.xml**”, se generó automáticamente para contener los datos del mes de julio del 2006.

Como lo indica la extensión del archivo, la estructura que se usa para el manejo de los datos es el formato “xml”, se usan cuatro tipos de nodos, para el almacenamiento de los registros, como se muestra en la tabla 4.4:

TIPO	Atributos(datos)	Texto contenido(datos)
Persona	Clave, CIA	Nombre de la Persona.
Asistencia	Fecha, Entrada, Salida, Lapso, c	No se usa.
CIA	Entrada, Salida	Nombre de la Compañía.
Viernes	Entrada, Salida	No se usa.

Tabla 4.4 Nodos que se usan para el manejo de los datos.

Ejemplo de un Archivo xml con registros en Fig. 4.8:



```

Nuevo Documento de texto.txt - Bloc de notas
Archivo Edición Formato Ver Ayuda
<?xml version="1.0" encoding="iso-8859-1"?>
<Registro>
  <Persona>
    <Nombre clave="tyu_456" cia="IMP">HILARIO GARCIA ANAYA</Nombre>
  </Persona>
  <Persona>
    <Nombre clave="sds587DAS" cia="IMP">Claudio</Nombre>
  </Persona>
  <Persona>
    <Nombre clave="ea1101CON" cia="SERVINEXT">HECTOR MANUEL GARCIA MORALES</Nombre>
    <Asistencia fecha="03/07/2006" Entrada="08:04 a.m." Salida="06:06 p.m." Lapso="09:56:00" c="-4" />
    <Asistencia fecha="04/07/2006" Entrada="08:04 a.m." Salida="06:00 p.m." Lapso="09:56:00" c="-4" />
  </Persona>
  <Persona>
    <Nombre clave="aar347PDA" cia="SERVINEXT">ALEJANDRO RAMIREZ TAPIA</Nombre>
    <Asistencia fecha="03/07/2006" Entrada="08:11 a.m." Salida="06:06 p.m." Lapso="09:49:00" c="-11" />
    <Asistencia fecha="04/07/2006" Entrada="08:10 a.m." Salida="06:01 p.m." Lapso="09:50:00" c="-10" />
  </Persona>
  <Persona>
    <Nombre clave="noo0500VN" cia="SERVINEXT">MAURICIO MORENO VILLEGAS</Nombre>
    <Asistencia fecha="03/07/2006" Entrada="08:08 a.m." Salida="06:07 p.m." Lapso="09:52:00" c="-8" />
    <Asistencia fecha="04/07/2006" Entrada="08:03 a.m." Salida="06:01 p.m." Lapso="09:57:00" c="-3" />
  </Persona>
  <Persona>
    <Nombre clave="rrr672RZR" cia="UVM">AVRIL PEREZ MUÑOZ</Nombre>
    <Asistencia fecha="03/07/2006" Entrada="09:00 a.m." Salida="04:00 p.m." Lapso="04:00:00" c="180" />
    <Asistencia fecha="04/07/2006" Entrada="09:00 a.m." Salida="02:00 p.m." Lapso="04:00:00" c="60" />
  </Persona>
  <Cia Entrada="08:00 a.m." Salida="06:00 p.m.">
    <Nombre>SERVINEXT</Nombre>
    <viernes Entrada="08:00 a.m." Salida="03:00 p.m." />
  </Cia>
  <Cia Entrada="09:00 a.m." Salida="01:00 p.m.">
    <Nombre>UVM</Nombre>
  </Cia>
</Registro>

```

Fig. 4.8 Ejemplo de Contenido de Archivo xml.

Persona: Contienen la información relativa a una persona. Puede o no contener nodos hijos de tipo asistencia, la clave solo se usa cuando el nodo se refiere una persona con rol de operador, en este ultimo caso al crearse su **atributo CIA**, se inicializa para que contenga el nombre predeterminado por la configuración como *compañía de los operadores* (ver posteriormente en configuración el archivo **myproy.config**).

Asistencia: Se crean al registrar la entrada o si se registra simultáneamente entrada y salida, la existencia de los tres últimos atributos (*salida*, *lapso* y *c*) depende de si se ha registrado o no la salida, lapso es el tiempo transcurrido entre la entrada y salida, “c” fue implementado para el manejo de un indicador que permite comparar la asistencia con respecto al horario de la compañía.

Cia: Sus atributos establecen el horario que deben cumplir el PAE de la compañía, puede o no contener un nodo hijo de tipo viernes lo cual indicaría que el horario del viernes es diferente al resto de la semana.

Viernes: Contiene horario de viernes de la compañía padre.

4.1.8 Otras páginas y archivos auxiliares

Archivo	Descripción
reloj.aspx	Se auto refresca cada minuto, ya que su única función es mostrar la hora, se ubica en un “frame” dentro de menuAux.aspx .
ayuda.htm	Comando ayuda se implemento como una página html (en una ventana nueva), debido a que solo requiere de mostrar documentación se realizo con dos “frames”, en el izquierdo un índice, en el derecho el contenido.
contenido.htm	Documento de ayuda: Índice que se muestra en “frame” izquierdo.
introducción.htm	Documento de ayuda: Breve introducción al sistema.
principal.htm	Documento de ayuda: Explica contenido de pantalla inicial.
inicio.gif	Imagen de ayuda: Imagen que se muestra en principal.htm
comandos.htm	Documento de ayuda: Explica brevemente los comandos del sistema.
acercade.htm	Documento de ayuda: Características generales del sistema.
login.htm	Documento de ayuda: Breve explicación como cambiar de rol.
entsal.htm alta.htm baja.htm	Documentos de ayuda: Explica el uso de comandos
myproy.config web.config	Contienen parámetros de configuración de la aplicación.

Tabla 4.5 Páginas auxiliares de la aplicación.

El objetivo y contenido de los archivos html de la tabla anterior se pueden visualizar en las “pantallas finales” del anexo V.

web.config: es un archivo para la configuración de la aplicación web que forma parte del marco de desarrollo del framework.NET, se crea automáticamente como parte del proyecto en el ambiente de desarrollo de “visual studio”, solo se usó para indicar que los parámetros de configuración se encuentran en el archivo “myproy.config”.

myproy.config: El uso de este archivo permite cumplir, con el requerimiento de que la aplicación pueda ejecutarse para proyectos en forma independiente, esto se hace anotando en el archivo ciertas claves (key), con los datos correspondientes al proyecto para el cual corre la aplicación.

La lectura de estos datos se hace al iniciar la aplicación y al iniciar la sesión gracias a procedimientos existentes en los objetos aplicación y de sesión (código en Archivo global.aspx) del ambiente .NET, que se muestran en la Fig. 4.9.

De esta manera estos datos quedan disponibles en los objetos “**Application**” y “**Session**” para cualquier página.

```
protected void Application_Start(Object sender, EventArgs e)
{
    Application["Proyecto"] = ConfigurationSettings.AppSettings["Proyecto"];
    Application["ciaOp"] = ConfigurationSettings.AppSettings["CiaOp"];
    Application["DirDatos"] = ConfigurationSettings.AppSettings["DirDatos"];
    //
}

protected void Session_Start(Object sender, EventArgs e)
{
    Session["NombreAdm"] = ConfigurationSettings.AppSettings["NombreAdm"];
    Session["PassAdm"] = ConfigurationSettings.AppSettings["PassAdm"];
}
```

Fig. 4.9 Código de inicio de Aplicación e inicio de Sesión.

Los archivos myproy.config y web.config, se volverán a mencionar mas adelante en la sección 4.3.3 configuración, dentro de 4.3 Distribución.

4.2 Pruebas

Aunque en la teoría de RUP la pruebas de “software” se deben realizar en forma independiente para cada ciclo y para cada iteración, en particular consideramos que hay muchos planteamientos a la hora de abordar el ‘probar software’, pero para verificar productos de forma efectiva se requiere de un proceso de investigación más que seguir un procedimiento al pie de la letra.

Una definición de "pruebas de software" es: proceso de evaluación de un producto desde un punto de vista crítico, donde el "probador" somete el producto a una serie de acciones inquisitivas, y el producto responde con su comportamiento como reacción[WIKIPEDIA2].

Durante el desarrollo del sistema se hicieron las pruebas en paralelo, con diversos datos, para cada uno de los comandos en las pantallas correspondientes.

También antes de iniciar la etapa de producción, se pidió a los usuarios hicieran uso de una versión Beta.

- **Login:** Se probó con los campos en blanco, o con nombres inexistentes de operadores campos y permitió la entrada, poniendo el rol de invitado como es correcto y solo mostrando en el menú comandos: *inicio, login, repMes, repSem, repDia y Ayuda*. Se probó con operador registrado y mostró adicionalmente los comandos: *entrada, salida, entSal, alta, baja y modifica*. Se probó con administrador registrado y mostró adicionalmente los comandos: *bajaOs, altaOp, altaCia, bajaCia, modCia, vieCia, y bitacora*.
- **Inicio:** Después de usar otros comandos este nos mostró la pantalla de inicio correctamente, con un operador firmado (y administrador) se comprobó que estando en esta pantalla y cambiar a una fecha diferente a la real, desaparecían los comandos entrada y salida.
- **entrada:** Al usarlo se apareció pantalla para confirmar registro de entrada del elemento mostrado en el menú secundario, y al aceptar quedo registrada la entrada.
- **salida :** Al usarlo se apareció pantalla para confirmar registro de salida del elemento mostrado en el menú secundario, y al aceptar quedo registrada la entrada
- **entSal:** Mostró pantalla para captura de horario de entrada y salida, al aceptar quedaron registrada la captura, si la hora de salida es menor a la de entrada no se acepta.
- **repMes:** Muestra reporte respectivo de la compañía en menú secundario.

- **repSem:** Muestra reporte respectivo de la compañía en menú secundario
- **repDia :** Muestra Reporte respectivo de la compañía en menú secundario
- **alta:** Aparecen controles para dar de alta a un elemento del PAE, al aceptar es anexado del sistema.
- **baja:** Aparece mensaje para confirmar, dar de baja al elemento del PAE que esta en el menú secundario, al aceptar es eliminado del sistema
- **modifica:** Aparecen controles para modificar datos del elemento del PAE que esta en el menú secundario, al aceptar son modificados sus datos.
- **ayuda:** Abre otra ventana con información de uso del sistema en una página html, con un menú en la parte izquierda.
- **bajaop:** apareció pantalla para seleccionar nombre de operador y confirmar eliminación del sistema, al aceptar quedo eliminado del sistema.
- **altaOp:** Mostró pantalla para captura datos de un operador, al aceptar quedaron registrados los datos, se comprueba que si se deja en blanco alguno de los datos muestra mensaje: “Debe indicar nombre y contraseña del operador”.
- **altaCia:** Mostró pantalla para captura datos de un compañía, al aceptar quedaron registrados los datos, se comprueba que si se deja en blanco alguno de los datos muestra mensaje : “Falta nombre de la compañía o alguno de los valores de entrada/salida “
- **bajaCia:** Aparece mensaje para confirmar, dar de baja a compañía que esta en el menú secundario, al aceptar es eliminado el registro respectivo del sistema.
- **modCia:** Aparecen controles para modificar datos de compañía que esta en el menú secundario, al aceptar son modificados sus datos.

- **vieCia:** Aparecen controles para modificar horario de viernes de compañía que esta en el menú secundario, al aceptar son modificados sus datos.
- **bitácora:** Muestra el contenido del archivo de bitácora.

Se pueden observar en el Anexo V “Pantallas del Sistema” las imágenes correspondientes, a la ejecución de algunos de los comandos.

4.3 Distribución

Por las características propias del IMP, la aplicación desde un principio se pensó dirigida al ambiente de la intranet, así que su distribución consistió básicamente en ponerla disponible en la intranet, esto se consiguió con los siguientes tres pasos:

- Establecimiento(o instalación) del servidor web
- Instalación de la aplicación en el servidor web
- Configuración de la aplicación para un proyecto en particular.

4.3.1 Establecimiento del servidor web

Se requirió de un servidor accesible en la red (intranet), con windows 2003, aunque también hubiese sido posible usar un equipo con windows 2000 y aun con windows XP.

La capacidad de almacenamiento del Servidor, para esta aplicación inicialmente no es muy relevante ya que el tamaño final de los archivos que componen la aplicación (sin datos) aproximadamente es de: 500 Kb.

En este servidor el administrador del equipo instaló el servicio del IIS (Internet Information Server), el IIS o servidor web, es una aplicación que casi todos los equipos con windows tienen, aunque por seguridad en las versiones mas recientes este servicio no esta habilitado, pero habilitarlo es una tarea relativamente fácil para el administrador del servidor. A continuación los pasos que se efectúan para habilitar el servicio mencionado en un equipo Windows XP:

1. Clic⁴ en **Inicio**, se selecciona Panel de control y, a continuación, clic en **Agregar o quitar programas**. Se inicia la herramienta Agregar o quitar programas.
2. Clic en Agregar o quitar componentes de Windows. Aparece el **Asistente para componentes de Windows**.
3. En la lista Componentes de Windows, clic en **Servicios de Internet Information Server (IIS)**.
5. Clic en botón '**Detalles**' y aparece la lista de componentes opcionales de IIS.
6. En la lista de componentes opcionales de forma predeterminada, están seleccionados los componentes siguientes:
 - Archivos comunes
 - Extensiones de servidor de FrontPage 2002
 - Complemento de Servicios de Internet Information Server
 - Administrador de Servicios de Internet Information Server
 - Servicio NNTP
 - Servicio SMTP
 - Servicio World Wide Web
7. Clic en **Servicio World Wide Web** y con el botón '**Detalles**' se muestra la lista de subcomponentes opcionales de IIS, debe estar seleccionado **Servicio World Wide Web**.
8. Clic en Aceptar hasta que el foco vuelva al Asistente para componentes de Windows.
9. Clic en Siguiente y, a continuación, finalizar el asistente.

⁴ Se refiere a oprimir el botón izquierdo del 'ratón' (Mouse) sobre un elemento de la pantalla.

4.3.2 Instalación de la aplicación

Aunque existe la opción de crear un paquete de instalación, totalmente automatizado, debido a que es un desarrollo particular para una sola empresa (IMP) fue más práctico que el administrador del servidor efectuara:

1. La creación del directorio: `\web\appcapae` para contener las páginas de CAPAE.
2. La creación del sitio web: `http://intranetimp/capae`, asociado al directorio recién creado (usando “Administrador del IIS”), y estableció los permisos adecuados para que cualquier usuario de la red (intranet), pudiera acceder de manera anónima al sitio, recuérdese que la seguridad la manejaría la aplicación.
3. El establecimiento de los permisos adecuados en el directorio `\web\appcapae` para que el desarrollador de la aplicación, pueda tener control total del directorio, de tal modo que desde su equipo, pueda transferir archivos al directorio mencionado.

Por su parte el desarrollador de la aplicación, para instalar los archivos que componen la aplicación, realizó los siguientes pasos desde su equipo:

- Conectándose a la intranet tomó acceso compartido del directorio del servidor, (creado en el paso uno del administrador), en un directorio virtual(`\\web\appcapae`)
- Abrió la aplicación de Visual Studio (de Microsoft)
- Abrió aplicación web, desarrollada CAPAE.
- Ejecutó el comando: “Copiar proyecto...” del menú Proyecto.
- En la ventana que se muestra (Fig. 4.10) usando la opción de “Recurso compartido en archivo” puso en “carpeta destino” el URL del sitio creado por el administrador en paso dos, y en “ruta de acceso”, el directorio virtual.

De esta manera los archivos necesarios de la aplicación son copiados al Servidor con el IIS.

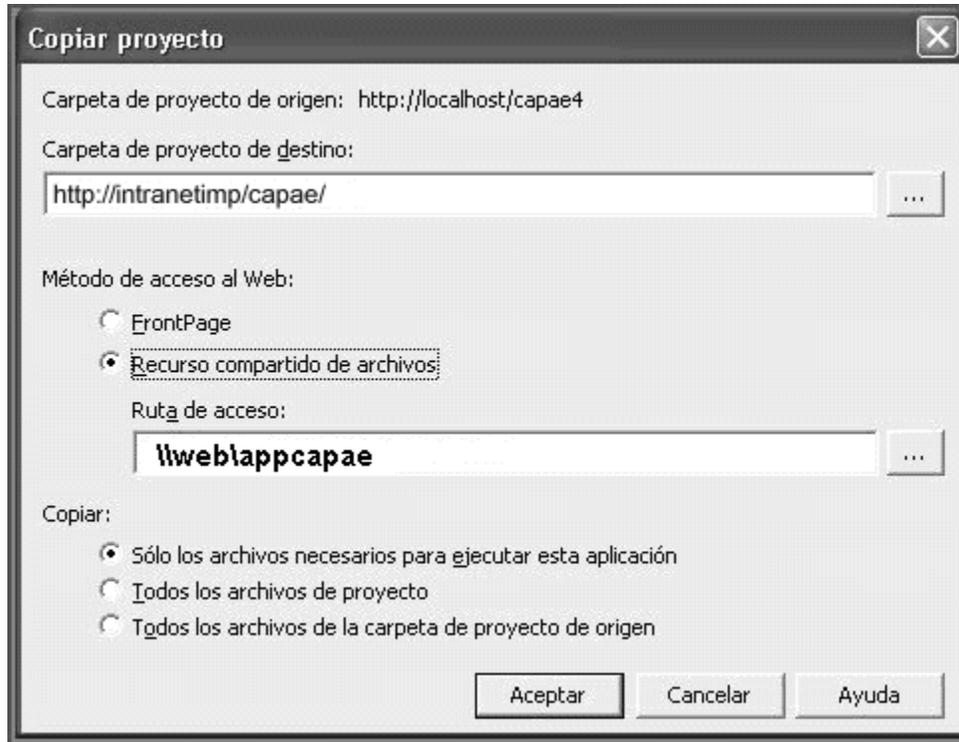


Fig. 4.10 ventana de dialogo al copiar la aplicación al servidor web.

4.3.3 Configuración de la aplicación

A través de la configuración se establecen algunas características asociadas al proyecto para el cual se va a usar, como nombre del proyecto, nombre del administrador, contraseña del administrador, etc.

Recuérdese que el archivo **Web.config** contiene la siguiente línea indicando a nuestra aplicación que use el archivo “myproy.config”.

```
<appSettings file="miproj.config"/>
```

Como ya se mencionó anteriormente para almacenar los parámetros de configuración se debe modificar el archivo **myproy.config**, en él se establecen ciertos campos o llaves (key) para almacenar los datos correspondientes al proyecto como:

- **MasterPageUserControl:** Nombre del control de usuario que se usara para especificar el aspecto (y relación entre las secciones) de la aplicación (se puede

cambiar sin recompilar la aplicación), en nuestro caso, este parámetro es siempre el mismo.

- **NombreAdm:** Un Nombre corto, que servirá para identificar al administrador (no se pone en los archivos xml de datos).
- **PassAdm:** Contraseña del administrador, para validarlo.
- **Proyecto:** Texto que identifica al proyecto.
- **CiaOp:** Nombre de compañía de operadores, para distinguirlos del PAE.
- **DirDatos:** Directorio para datos, bajo el directorio de la aplicación debe existir y es donde se fuardaran los archivos .xml.

En la figura 4.11 se observa el ejemplo de un archivo myproy.config.

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
<add key="MasterPageUserControl" value="MasterPageUserControl.ascx"/>
  <add key="NombreAdm" value="Hilario"/>
  <add key="PassAdm" value="XXXX.:" />
  <add key="Proyecto" value="Centro de Atencion a Usuarios"/>
  <add key="CiaOp" value="IMP"/>
  <add key="DirDatos" value="dts"/>
</appSettings>
```

Figura 4.11 Contenido de myproy.config

Los dos último paso de la configuración antes de iniciar su uso diario son:

- El administrador (usualmente jefe de proyecto) interactuando con el sistema establece las compañías (nombre, horarios) y el (los) operador(es) encargados de realizar las capturas.
- Un operador interactuando con el sistema da de alta al Personal Auxiliar Externo de las compañías establecidas atrás.

Finalmente se deben considerar otros elementos, que aunque están fuera del ámbito de este documento son muy importantes para la operación y mantenimiento del sistema: las políticas y procedimientos que se deben establecer, para:

- **Responsable** de monitoreo de la aplicación.
- **Respaldos periódicos** de los datos (en el directorio: datos) que conforme pase el tiempo incrementarán su tamaño.
- **Mantenimiento del sistema** para mejorar y optimizar su uso.

El desarrollo de cualquier sistema implica una serie de documentos que pueden ser muy extensos (como lo pide la metodología de RUP), o muy breves, nuestra intención en este y el anterior capítulo fue de hacerlos muy breves pero lo bastante claro como para que se pueda comprender el trabajo realizado en la implementación de la solución. Evidentemente en algunos casos pude haber confusión debido al uso(o mal uso) de los diagramas y conceptos de UML y Framework.NET.

Capítulo 5. Resultados y discusión

Al realizar la implementación del sitio Web de acuerdo a la metodología expuesta, el resultado que se obtiene es que todas las pruebas (capítulo 4) se realizaron satisfactoriamente, de acuerdo a la definición de los casos de uso propuestos inicialmente (capítulo 1) y las pantallas que muestran el resultado se pueden consultar en el Anexo V “Pantallas del Sistema”.

Es evidente que en la práctica siempre surgen detalles en las pruebas, que no funcionan como uno espera, pero creo se deben considerar como algo natural, ya que es parte de RUP si consideramos **la fase de transición** (Fig. 3.1)

El foco de la fase de transición es asegurarse de que el software esté disponible para sus usuarios finales. La fase de transición puede atravesar varias iteraciones, e incluye la prueba del producto con objeto de lanzamiento, y la realización de los ajustes de menor importancia basados en la retroalimentación del usuario. En este punto del ciclo de vida, la retroalimentación del usuario, debe principalmente enfocarse al ajuste fino del producto, configuración, características de la instalación y de la usabilidad, todas las principales características estructurales se deben haber resuelto mucho antes en el ciclo vital del proyecto

Los principales detalles que surgieron en la fase de transición fueron:

- Falta de Permisos para escribir en los directorios donde se guardan los datos.
- Petición de los usuarios de modificaciones en los colores usados de fondo.

Estos dos puntos fueron resueltos de inmediato, en la versión final del CAPAE.

También al final como los usuarios hicieron otras sugerencias, que serán consideradas para la siguiente versión:

CAPAE

- Implementar comandos de respaldo de información, ya que procedimiento actual es manual (no documentado).
- Anexar campos extras de comentarios, a cada persona, para realizar seguimiento de movimientos de sus actividades, o anotar motivos de inasistencia.

Este sistema ha estado funcionando durante los dos últimos años, en el IMP para los dos proyectos de la Gerencia de Tecnologías de Información(GTI) que hacen uso del “outsourcing”, con un resultado satisfactorio, el uso en otros proyectos no se ha realizado debido a las políticas del IMP .

En futuras versiones también sería positivo considerar:

- Migrar el sistema a Framework.NET 2.0 para aprovechar el uso de Páginas Maestras que ya vienen integradas como librerías nativas.
- Uso de técnicas habituales en Web 2.0
- Optimizar la validación del personal PAE con el uso de lectores de huellas.
- Mejorar el diseño de las pantallas.
- Crear otros módulos que permitan una mayor explotación de la información existente como la consulta de graficas donde se reflejen las estadísticas de asistencia del personal.

Conclusiones

La experiencia derivada de este sistema y otros en los que hemos trabajado nos permite hacer hincapié en la importancia que tiene para el buen desarrollo de sistemas:

- El estudio y uso de metodologías de diseño probadas, robustas, ya establecidas.
- El uso de tecnologías basadas en *objetos y patrones de diseño*, para acelerar el proceso de creación del sistema.
- El uso de lenguajes y productos ampliamente usados, pero, relativamente modernos como c# o Java.
- La mejora continúa del sistema.

Evidentemente el sistema aquí expuesto es un sistema sencillo(o de pocos recursos), pero por lo mismo, considero que es un buen ejemplo de los que se pueden hacer, al aplicar las recomendaciones esbozadas en el párrafo anterior.

En sistemas grandes toma especial relevancia la coordinación del personal que interviene, así que ya sea para la comunicación entre ellos, como para el posterior mantenimiento y mejora del sistema, es imprescindible la creación de su documentación durante su desarrollo bajo algún estándar ya probado.

También es importante ser flexible y considerar desviaciones de los estándares, siempre y cuando sea en provecho del desarrollo de nuestros sistemas.

Anexo I: Definición de los casos de uso

En este anexo se muestra solo un grupo de “casos de uso”, al final se mencionan los nombres de los faltantes, en la documentación del proyecto se documentaron todos.

Para simplificar esta documentación se inicia con “usuario” las acciones de quien esta interactuando con el sistema (el usuario puede ser cualquiera de los actores definidos con los permisos correspondientes), y evidentemente “sistema” indica las acciones que efectúa el CAPAE.

I- Iniciar sistema

Descripción: Abrir la página inicial del sistema para poder interactuara con el sistema.

Precondición: Ninguna.

Poscondición: Pantalla principal abierta, con comandos activados de acuerdo al rol y condiciones del sistema.

Curso normal de eventos:

1. **Usuario:** usa su navegador para acceder al sitio web donde se encuentra la página “html” inicial (o principal) del CAPAE.
2. **Sistema:** obtiene fecha de trabajo, rol de usuario de la actual sesión.
3. **Sistema:** Muestra pantalla principal con comandos, controles e información actual del sistema compuesta por:
 - *Menú:* repmes, repsem, repdia, entrada, salida, entsal, alta, baja, modifica, ayuda,....
 - *Menú Secundario:* Lista de elementos PAE registrados, lista de compañías.
 - *Controles:* Calendario (con fecha de trabajo).
 - *Información:* Hora actual del sistema, fecha de trabajo, tabla con los registros de entrada/salida capturados de la fecha de trabajo.
- 4- **Sistema:** De acuerdo al rol del usuario deshabilita comandos en sesión.

CAPAE

Curso alterno #1: Si en el paso dos no existe definido el rol en la sesión:

2.1 Ejecuta caso de uso login

El proceso termina.

Curso alterno#2: En el paso tres si no hay datos registrados (compañías y/o elementos PAE):

- 3.1 **Sistema:** Muestra mensaje indicando no hay datos.
- 3.2 **Sistema:** Deshabilita todos los comandos (menos ADMIN)

El proceso continua con el paso cuatro del curso normal.

II C1 Login

Descripción: Registrar al usuario para establecer su rol.

Precondición: Ninguna.

Poscondición: Pantalla principal abierta,

Curso normal de eventos:

1- **Sistema:** Muestra pantalla de login con controles y comandos para su registro:

- Controles para captura de nombre y contraseña.
- Control botón para enviar datos.

2- **Usuario:** Teclea y envía “identificación”(a través de los controles).

3- **Sistema** en base a “identificación” establece el rol del usuario (valido para la sesión)

4- **Sistema:** Anota inicio de sesión para el usuario en bitácora.

III C2 ENTRADA

Descripción: Registrar entrada de un elemento del PAE. El elemento PAE se presenta ante el operador para que este registre su hora de entrada.

Precondición: Usuario con rol de operador o administrador, en pantalla principal.

Poscondición: Datos de entrada registrados o mensaje de por que no se puede hacer el registro.

Curso normal de eventos:

1- **Usuario:** Selecciona nombre de lista de elementos PAE y activa comando ENTRADA.

2- **Sistema:** Muestra pantalla con:

- Calendario con fecha de trabajo
- Datos de elemento seleccionado.
- Hora del sistema.
- Botón “Aceptar”.

3- **Usuario:** Confirma acción con botón “Aceptar”.

4- **Sistema:** Anota hora de entrada, usando los datos mostrados.

5- **Sistema:** Vuelve a mostrar pantalla principal.

Curso alterno#1: En paso dos: Si ya se capturo antes la entrada para esta fecha, y este elemento:

- 2.1 **Sistema** Muestra mensaje “Ya se capturo la hora de entrada”.
- 2.2 **Sistema** Deshabilita botón “Aceptar”, Proceso ya no puede continuar con los pasos tres a cinco

IV C3 ENTRADA

Descripción: Registrar salida de un elemento del PAE. El elemento PAE se presenta ante el operador para que este registre su hora de salida.

Precondición: Usuario con rol de operador o administrador, en pantalla principal.

Poscondición: Datos de entrada registrados o mensaje de por que no se puede hacer el registro.

Curso normal de eventos:

1- **Usuario:** Selecciona nombre de lista de elementos PAE y Activa comando SALIDA.

2- **Sistema:** Muestra pantalla con:

- Calendario con fecha de trabajo
- Datos de elemento seleccionado.
- Hora del sistema.
- Botón “Aceptar”.

3- **Usuario:** Confirma acción con botón “Aceptar”.

4- **Sistema:** Anota hora de salida, usando los datos mostrados.

5- **Sistema:** Vuelve a mostrar pantalla principal.

Curso alterno#1: En paso dos: Si ya se capturo antes la salida para esta fecha, y este elemento.

- 2.1 **Sistema** Muestra mensaje “Ya se capturo la hora de salida”.
- 2.2 **Sistema** Deshabilita botón “Aceptar”.
- Proceso ya no puede continuar con los pasos tres a cinco

V C4 ENTSAL

Descripción: Registrar entrada y salida de un elemento del PAE. El operador desea registrar(o cambiar) entrada y salida simultáneamente por necesidades fuera de lo normal.

Precondición: Usuario con rol de operador o administrador, en pantalla principal.

Poscondición: Datos de entrada registrados o mensaje de por que no se puede hacer el registro.

Curso normal de eventos:

1- **Usuario:** Selecciona nombre de lista de elementos PAE y activa comando ENTSAL.

2- **Sistema:** Muestra pantalla con:

- Calendario con fecha de trabajo
- Datos de elemento seleccionado.
- Controles para captura de hora de entrada
- Controles para captura de Hora de Salida
- Botón “Aceptar”.

3- **Usuario:** llena datos de entrada y salida, Confirma acción con botón “Aceptar”,

4- **Sistema:** Lee datos de entrada y salida

5- **Sistema:** Anota hora de entrada y salida, usando los datos mostrados.

6- **Sistema:** Anota en bitácora este movimiento.

7- **Sistema:** Vuelve a mostrar pantalla principal.

Curso alterno#1: En paso cuatro: Sistema detecta combinación invalida:

4.1 **Sistema** Muestra mensaje con error respectivo.

Proceso No continua con siguientes pasos.

VI C5 REPMES

Descripción: Crea y muestra reporte de horas trabajadas por día y por persona, mensual.

En este Caso se desea consultar el reporte mensual, un vez desplegado se puede imprimir con los comando del navegador para imprimir la página abierta.

Precondición: Usuario con cualquier rol, sistema inicializado.

CAPAE

Poscondición: Reporte mensual en pantalla de una compañía en particular.

Curso normal de eventos:

1- **Usuario:** Usa comando (REPMES) para Reporte mensual.

2- **Sistema:** Muestra en pantalla:

- Lista de compañías registrada en CAPAE
- Reporte de 1ra compañía, indicando horas trabajadas por día y por persona del mes (según fecha de trabajo).

VII C6 REPSEM

Descripción: Reporte de horas trabajadas por día y por persona, semanal.

En este Caso se desea consultar el reporte semanal, de horas trabajadas un vez desplegado se puede imprimir con los comando del navegador para imprimir la página abierta.

Precondición: Usuario con cualquier rol, en pantalla principal.

Poscondición: Reporte Semanal en pantalla de una compañía en particular.

Curso normal de eventos:

1- Usuario: Usa comando (REPSEM) para reporte semanal.

2- Sistema: Muestra en pantalla:

Lista de Compañías registrada en CAPAE

Reporte indicando horas trabajadas por día y por persona, para 1ra Compañía, por semana del actual mes, (mes de la actual fecha de trabajo).

Casos de uso faltantes.

A continuación se muestran solo los nombres y funciones, del resto de los casos de uso

VIII C7 REPDIA Reporte de asistencias del personal, día por día, mensual.

IX C8 ALTA Dar de alta un elemento PAE.

X C9 BAJA Dar de baja un elemento PAE.

XI C10 MODIFICA Modifica datos de un elemento PAE.

XII C11 AYUDA Consultar documentación del sistema.

XIII C12 ELIMINA Dar de baja del sistema a un operador.

XIV C13 NUEVO Dar de alta en el sistema a un operador.

XV C14 NUEVA CIA Dar de alta en el sistema a una compañía.

XVI C15 ELIMINAR CIA: Dar de baja en el sistema a una compañía.

XVII C16 MODIFICA CIA: Modifica una o más características de una compañía previamente dada de alta.

XVIII C17 E/S VIERNES: Redefine horario de trabajo, para los viernes de una compañía previamente dada de alta.

XIX C18 RECALCULA: Recalcula horas trabajadas debido a cambios hechos en registros para una compañía en particular.

XX C19 BITACORA: Consultar bitácora.

XXI- Cambiar fecha.

XXII- Cambiar de compañía en reporte mensual

XXIII- Cambiar de compañía en reporte semanal

XXIV- Cambiar de compañía en reporte diario

Anexo II: Prototipos y diagramas de secuencia de los “casos de uso”

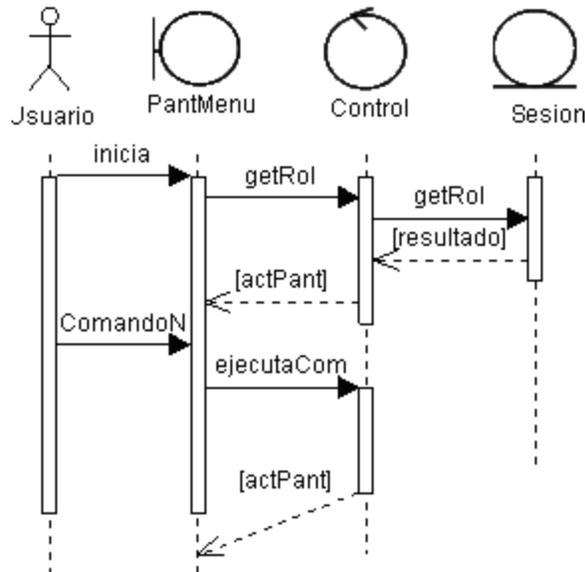
A continuación se muestra un reducido grupo de prototipos y sus diagramas de secuencia, al final se indican los nombres de los faltantes.

I -M1 Menú Principal:

login | entrada | salida | entSal | repMes | repSem | repDia | alta | baja | modifica ayuda | bajaOp | altaOp | altaCia | bajaCia | modCia | vieCia | recalcula | bitacora

Cada liga dará acceso a las páginas que ejecutaran los comandos establecidos.

Controles: Un Enlace por cada comando.



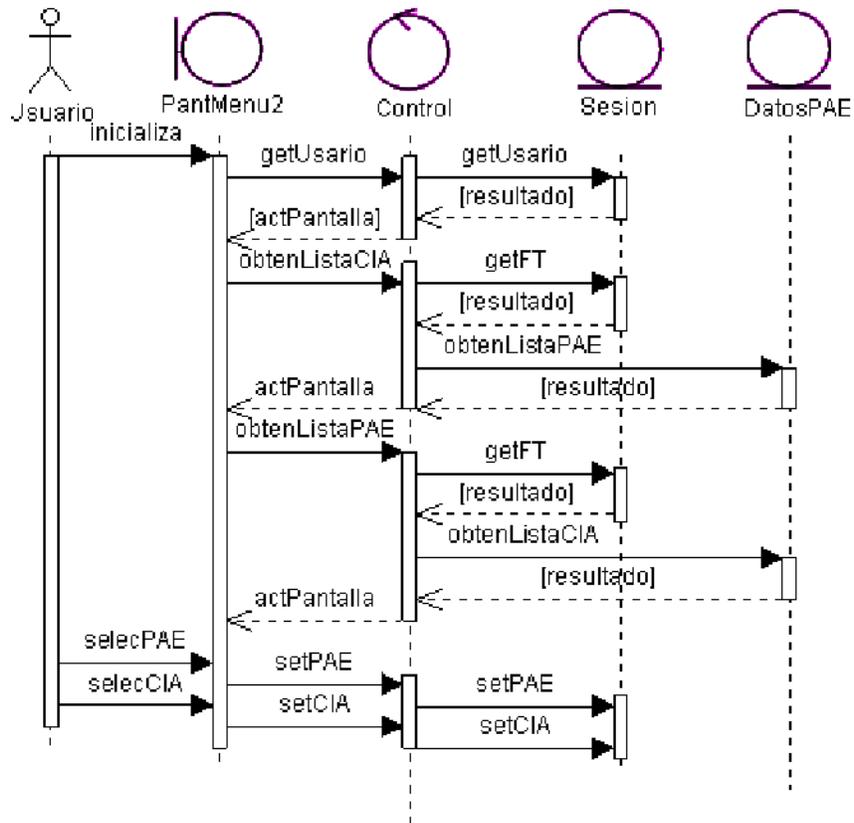
II- M2 Secundario:

usuario	12:07 p.m.	PAE: HECTOR MANUEL GARCIA MORALES	CEPRA
---------	------------	-----------------------------------	-------

Controles:

- Texto usuario
- Texto hora actual*
- Lista de elementos PAE
- Lista de compañías

Diagrama de secuencia:



*Para el texto hora actual se usará un “frameset” [EVANS1], que haga referencia a una página, que se auto refresque cada minuto, su código solo mostrará la hora del servidor.

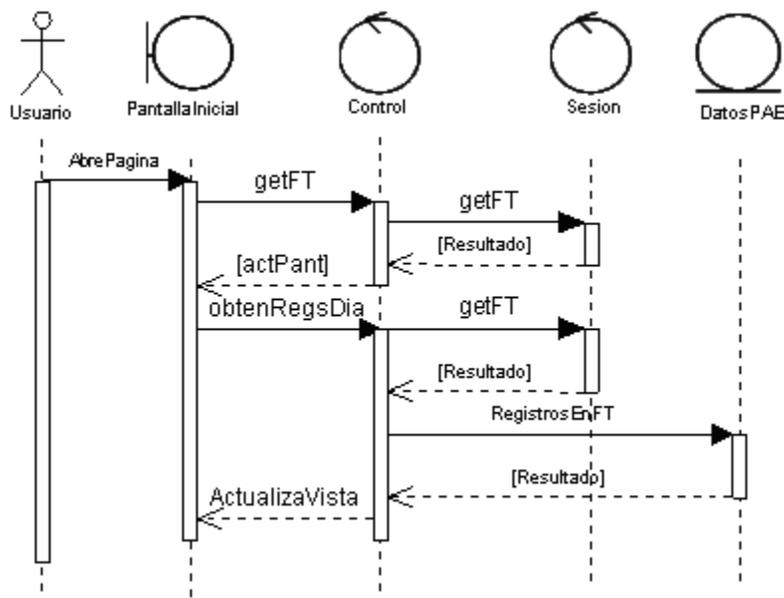
III - Pantalla Inicial

Septiembre de 2006							(17) EN: Viernes, 22 de Septiembre de 2006		
D	L	M	M	J	V	S	Cia-Nombre	Entrada	Salida
27	28	29	30	31	1	2	SERVINEXT-HECTOR MANUEL GARCIA MORALES	08:19 a.m.	
3	4	5	6	7	8	9	SERVINEXT-ALEJANDRO RAMIREZ TAPIA	08:02 a.m.	
10	11	12	13	14	15	16	SERVINEXT-MAURICIO MORENO VILLEGAS	08:13 a.m.	
17	18	19	20	21	22	23	SERVINEXT-ENRIQUE GONZÁLEZ ESTÉBANEZ	08:03 a.m.	
24	25	26	27	28	29	30	CEPRA-WENDY PAULIN MARTINEZ	08:43 a.m.	
1	2	3	4	5	6	7	CEPRA-ISRAEL VILLA MORALES	08:58 a.m.	
							CEPRA-JAVIER CEDILLO ITURIBE	08:18 a.m.	
							CEPRA-GUILLERMO CASTRO BARBEYTO	08:32 a.m.	
							LANWARE-FERNANDO GALLARDO ALONSO		
							LANWARE-ROBERTO ESPINOSA		
							LANWARE-YOLANDA PADILLA ANGELES	08:12 a.m.	05:14 p.m.
							SERVINEXT-ROBERTO HERNÁNDEZ TORRES	08:13 a.m.	
							CEPRA-GUADALUPE IVÁN VELÁSQUEZ SÁMANO	08:28 a.m.	
							UVM-AVRIL PEREZ MUÑOZ	10:02 a.m.	
							SERVINEXT-PEDRO ARTURO GARCÍA CERDA	08:03 a.m.	
							SERVINEXT-LUIS ANTONIO HERNÁNDEZ GARCÍA	08:13 a.m.	
							POLI-OLMO CASTANEIRA RIOS	09:16 a.m.	01:59 p.m.

Controles:

- Calendario: Para que el usuario indique que fecha de trabajo desea usar.
- Texto con cantidad de elemento PAE (entre paréntesis) y fecha de trabajo.
- Tabla con nombres y los registros capturados en la fecha de trabajo

Diagrama de Secuencia:



La interacción con el calendario se muestra en cambiar fecha.

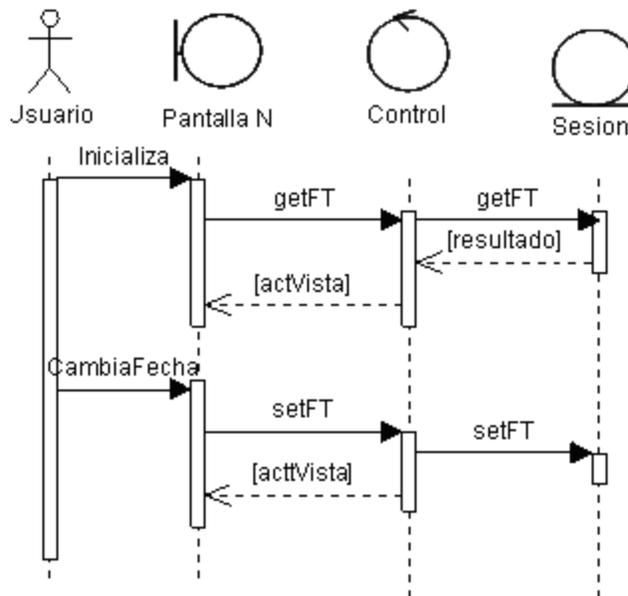
IV- Cambiar fecha.

Septiembre de 2006						
D	L	M	M	J	V	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Controles:

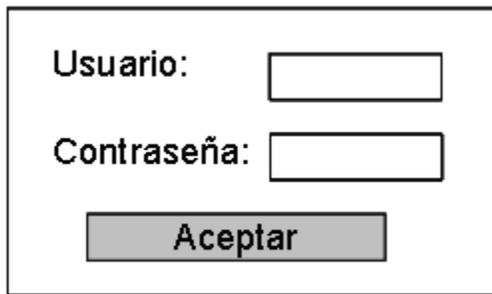
- Calendario: Permite al usuario cambiar la fecha de trabajo*

Diagrama de secuencia.



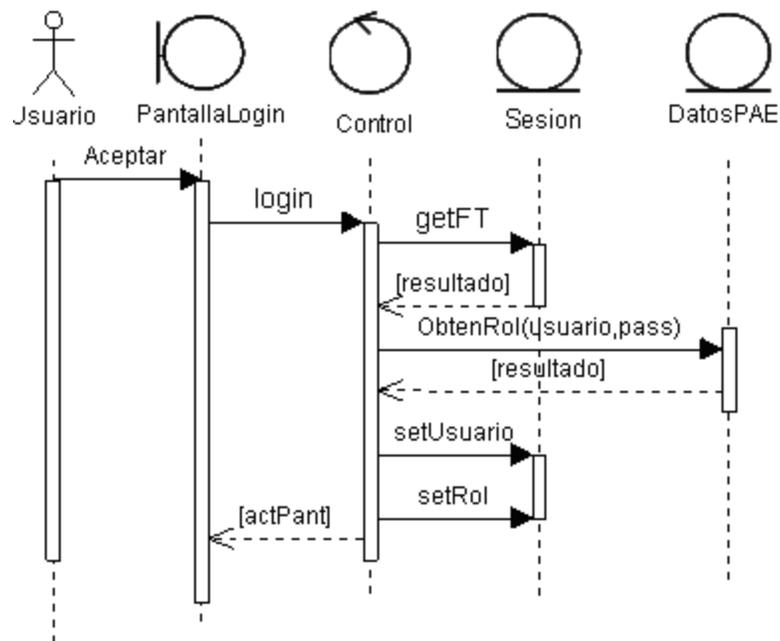
**Este control se encuentra en varias pantallas, por lo que la Actualización de la vista (actVista) dependerá de los controles existentes.*

V Login.



- Controles:
- Caja de Texto: Usuario
 - Caja de Texto: Contraseña
 - Boton: Aceptar.

Diagrama de secuencia.



VI-C2 Registrar entrada.

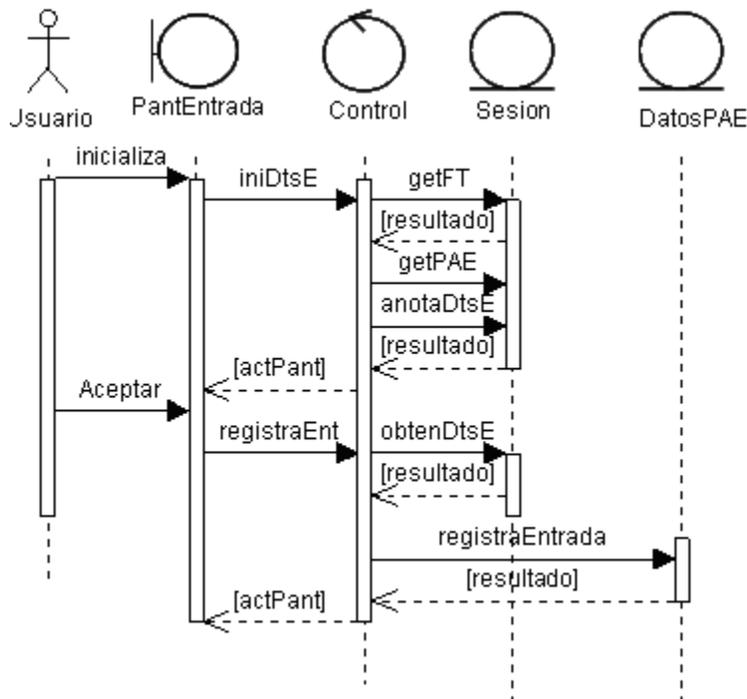
Septiembre de 2006						
D	L	M	M	J	V	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

HECTOR MANUEL GARCIA MORALES
Cia : SERVINEXT
ENTRADA : 02:50 p.m.

Controles:

- Calendario.
- Texto: elemento PAE, compañía, hora de entrada.
- Botón aceptar.

Diagrama de secuencia:



Otros prototipos y diagramas de secuencias.

VII –C3 Registrar salida.

VIII- C4 Registrar entrada y salida.

IX – C5 Ver reporte mensual

X – C6 Ver reporte semanal

XI -C7 Ver reporte diario

XII- C8 Alta de PAE.

XIII- C9 Baja de PAE.

XIV – C10 Modificar PAE.

XV -C11 Ayuda ver documentación

XVI- C12 Baja de operador.

XVII- C13 NUEVO: Dar de alta en el sistema a un operador.

XVIII- C14 NUEVA CIA: Dar de alta en el sistema a una compañía.

XIX- C15 ELIMINAR CIA.: Dar de baja en el sistema a una compañía.

XX - C16 MODIFICA CIA.: Modifica una o más características de una compañía previamente dada de alta.

XXI- C17 E/S VIERNES: Redefine horario de trabajo, para los viernes de una compañía previamente dada de alta.

XXII- C18 RECALCULA: Recalcula horas trabajadas debido a cambios hechos en registros para una compañía en particular.

XXIII- C19 Bitácora: Mostrar registros de movimientos.

Anexo III: Diagramas de clases

A continuación se muestran los diagramas de las principales clases y al final los nombres de todas las Clases del CAPAE:

ClsControl: La usan los controles de usuario, (ver directorio 'webusercontrol').

ClsControl
<pre> -m_sesion : System.Web.SessionState.HttpSessionState -m_Aplicacion : System.Web.HttpApplicationState -m_pagina : System.Web.UI.Page +rol : string +dirDatos : string +usuario : string +paes : string +cia : string +ft : DateTime +ciaOp : string +comando : string +PAEm : string +PAEbaja : string +tipodeReporte : string +ciaModifica : string +ciaModEntSal : string +ciaModVie : string </pre>
<pre> +ClsControl(Session : HttpSessionState, Page : System.Web.UI.Page) +actualizaCias(comando : string, cia : string, horario : string) : string +actualizaPersonas(comando : string, nombre : string, ciaPAE : string) : string +anotaDtsS(fecha : DateTime, PAE : string, hrsal : string) : void +anotaDtsE(fecha : DateTime, PAE : string, hrent : string) : void -notaEnLog(generator : string, comando : string, fecha : string, Nombre : string, info : string) : void +bitacoraTabla() : DataTable +ciadePAE(PAE : string) : string -datosActuales() : capae4.ClsDatosPAE -convES(horario : string, entsal : string []) : void -filelog() : string +generaRepMensual(Table1 : Table) : string +generaRepHrsDiario(mesCompleto : bool) : DataTable +listaCia() : ArrayList +listaPAE() : ArrayList +listaPAE(Cia : string) : ArrayList +login(usuario : string, pass : string, ip : string) : bool +nombreCorto(nom : string) : string +obtenerRegistro(PAE : string, f : DateTime, entrada : string, salida : string, lapso : string) : void +obtenRegsDia() : DataTable +obtenDtsE(fecha : DateTime, nombre : string, hrent : string) : string +obtenDtsS(fecha : DateTime, nombre : string, hrsal : string) : string -obtenHorario(cia : string, ent : System.DateTime [], sal : System.DateTime []) : void +obtenHorario(cia : string, hhmam : string, viernes : string) : void +operadores() : ArrayList +paginaComando(comando : string) : string +permiso(comando : string) : bool +registro(comando : string, fecha : DateTime, PAE : string, hrent : string, hrsal : string) : string +totHorasCia(cia : string, mes : bool) : double +titHorario() : string +verificaRegistro(comando : string, FT : DateTime, PAE : string, hrent : string, hrsal : string) : string </pre>

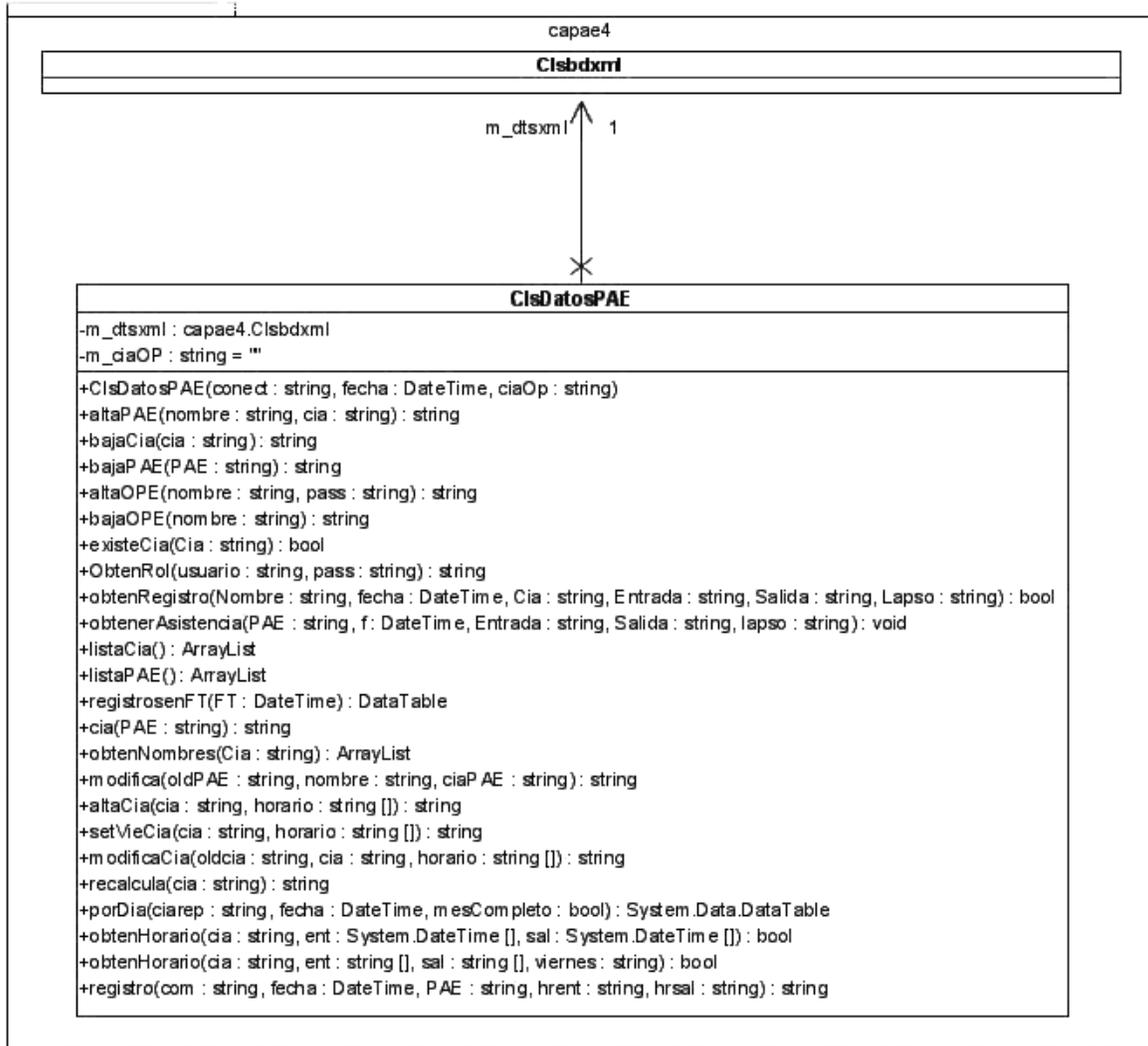
ClsBitacora: Para el manejo de bitácora.

ClsBitacora
-m_pathlog : string = "" -m_verbose : bool = false +verbose : bool
+ClsBitacora(pathlog : string) +registra(mensaje : string) : void +contenido() : string

ClsRepMensualXDia: Clase auxiliar para reportes.

ClsRepMensualXDia
-_fecha : DateTime -_fecha1runes : DateTime -_dia : string +Encabezado : string
+ClsRepMensualXDia(f : DateTime, Cia : string) +generaRepMensual(Table1 : Table, dtzPae : capae4.ClsDatosPAE) : string -CreaCelda(tipo : string, sthmm : string, dthmm : DateTime, hayhorario : bool) : System.Web.UI.WebControls.TableCell +encabezado1(semmana : int, columna : int) : string +encabezado2(columna : int) : string +asistencias(Nombre : string, semana : int, obpae : capae4.ClsDatosPAE) : string []

ClsDatosPAE: Para el manejo de la persistencia de los datos.



Clsbdxml: Maneja los datos en formato xml.

Clsbdxml
<pre> -mArchivodatos : string -mdocxml : XmlDocument mWarning : string = "" -mpathDir : string = "" mFechaDts : DateTime -mExisteMensual : bool = false -mrootxml : string = "<?xml version='1.0' encoding='iso-8859-1'?><Registro/>" +warning : string +Clsbdxml() +abreDocumentoXML(fecha : DateTime, pathDir : string) : bool actualizaDtsAsistencia(tipo : string, f : DateTime, datos : string []) : bool actualizaDtsCia(funcion : string, Cia : string, datos : string []) : bool actualizaDtsPersona(funcion : string, Nombre : string, datos : string []) : bool +actualizaHoras(cia : string) : int -calculaLapso(Nombre : string, s : DateTime, e : DateTime, c : string, diasem : System.DayOfWeek) : TimeSpan -creaAtributo(attr : XmlAttributeCollection, nombre : string, valor : string) : void +creaClave(Nombre : string) : string -creaNodoAsistencia(Nombre : string, NodoPersona : XmlNode, fwork : DateTime, Entrada : string, Salida : string) : XmlNode -creaNodoCia(cia : string, ent : string, sal : string, vienesEnt : string, vienesSal : string) : XmlNode -creaNodoPAE(Nombre : string, clave : string, Cia : string) : XmlNode -creaNuevoMensual(mrootxml : string) : bool -diasem(f : DateTime) : string +nombreCorto(nom : string) : string obtenAsistencia(Nombre : string, f : DateTime, Entrada : string, Salida : string, Lapso : string, c : string) : void obtenHorario(cia : string, ent : System.DateTime [], sal : System.DateTime []): bool +obtenHorario(cia : string, ent : string [], sal : string [], vienes : string) : bool -obtenNodoAsistencia(miNodo : XmlNode, f : DateTime) : XmlNode -obtenNodoAsistencia(Nombre : string, f : DateTime) : XmlNode -obtenNodoX(tipo : string, nombre : string) : XmlNode +obtenNombres(omiteCia : bool, Cia : string) : ArrayList +obtenNombresCias(sinCia : string) : System.Collections.ArrayList +obtenRegistro(Nombre : string, cia : string, clave : string) : bool +obtenRegistro(Nombre : string, Cia : string, fecha : DateTime, Entrada : string, Salida : string, Lapso : string) : bool porDia(cia : string, f : DateTime, mesCompleto : bool) : DataTable -salva() : bool +tablaAsistencia(omiteCia : string) : DataTable </pre>

Controles de Usuario.

Los siguientes controles se encuentran bajo la carpeta ‘webusercontrols’ (a excepción de los que inician con ‘Master’), se cargan en los “placeholder” de las páginas maestras (ver al final MasterPage).

MasterContentsControl: Pantalla Inicial.

MasterContentsControl
#Calendar1 : System.Web.UI.WebControls.Calendar #DataGrid1 : System.Web.UI.WebControls.DataGrid #lblmsg : System.Web.UI.WebControls.Label
-Page_Load(sender : object, e : System.EventArgs) : void -inicializa() : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -Calendar1_SelectionChanged(sender : object, e : System.EventArgs) : void

login: Registra usuario para identificarlo.

login
#Button1 : System.Web.UI.WebControls.Button #txtpass : System.Web.UI.WebControls.TextBox #txtcuenta : System.Web.UI.WebControls.TextBox #lbltitulo : System.Web.UI.WebControls.Label
-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -Button1_Click(sender : object, e : System.EventArgs) : void

ActPers: Actualizan datos del personal.

actpers
<pre>#panelPAE : System.Web.UI.WebControls.Panel #btnAceptar : System.Web.UI.WebControls.Button #lblCia : System.Web.UI.WebControls.Label #txtNombre : System.Web.UI.WebControls.TextBox #txtNombreOP : System.Web.UI.WebControls.TextBox #txtPass : System.Web.UI.WebControls.TextBox #panelOp : System.Web.UI.WebControls.Panel #listaOp : System.Web.UI.WebControls.DropDownList #lblMensaje : System.Web.UI.WebControls.Label</pre>
<pre>-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -inicializa() : void -btnAceptar_Click(sender : object, e : System.EventArgs) : void</pre>

admcia: Se usa en los comandos que actualizan datos de las compañías del PAE.

admcia
<pre>#btnAceptar : System.Web.UI.WebControls.Button #lblmsg : System.Web.UI.WebControls.Label #txtCia : System.Web.UI.WebControls.TextBox #smpm : System.Web.UI.WebControls.DropDownList #mmsal : System.Web.UI.WebControls.DropDownList #hhsal : System.Web.UI.WebControls.DropDownList #eam pm : System.Web.UI.WebControls.DropDownList #mment : System.Web.UI.WebControls.DropDownList #panelhorario : System.Web.UI.WebControls.Panel #Panel1 : System.Web.UI.WebControls.Panel #gridbitacora : System.Web.UI.WebControls.DataGrid #hhent : System.Web.UI.WebControls.DropDownList</pre>
<pre>-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -inicializa() : void -obtenEntSal() : string -btnAceptar_Click(sender : object, e : System.EventArgs) : void -actlzaCtrlES(viernes : string) : void -actlzaCtrlES(ent : string [], sal : string []) : void</pre>

menú: Aparece en la parte superior de casi todas las páginas.

menu
#mnualta : System.Web.UI.WebControls.LinkButton #mnusalida : System.Web.UI.WebControls.LinkButton #mnulogin : System.Web.UI.WebControls.LinkButton #mnuentSal : System.Web.UI.WebControls.LinkButton #mnurepMes : System.Web.UI.WebControls.LinkButton #mnuentrada : System.Web.UI.WebControls.LinkButton #mnurepDia : System.Web.UI.WebControls.LinkButton #mnubaja : System.Web.UI.WebControls.LinkButton #mnumodifica : System.Web.UI.WebControls.LinkButton #mnubajaOp : System.Web.UI.WebControls.LinkButton #mnualtaOp : System.Web.UI.WebControls.LinkButton #mnualtaCia : System.Web.UI.WebControls.LinkButton #mnubajaCia : System.Web.UI.WebControls.LinkButton #mnumodCia : System.Web.UI.WebControls.LinkButton #mnuevieCia : System.Web.UI.WebControls.LinkButton #mnubitacora : System.Web.UI.WebControls.LinkButton #mnuinicio : System.Web.UI.WebControls.LinkButton #mnurepSem : System.Web.UI.WebControls.LinkButton
-Page_Load(sender : object, e : System.EventArgs) : void -inicializa() : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -LinkButton1_Click(sender : object, e : System.EventArgs) : void -mnuayuda_Click(sender : object, e : System.EventArgs) : void

menuaux: Se usa en forma constante en casi todas las páginas.

menuaux
#listaPae : System.Web.UI.WebControls.DropDownList #listaCIA : System.Web.UI.WebControls.DropDownList #Label1 : System.Web.UI.WebControls.Label
-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -inicializa() : void -listaPae_SelectedIndexChanged(sender : object, e : System.EventArgs) : void -listaCIA_SelectedIndexChanged(sender : object, e : System.EventArgs) : void

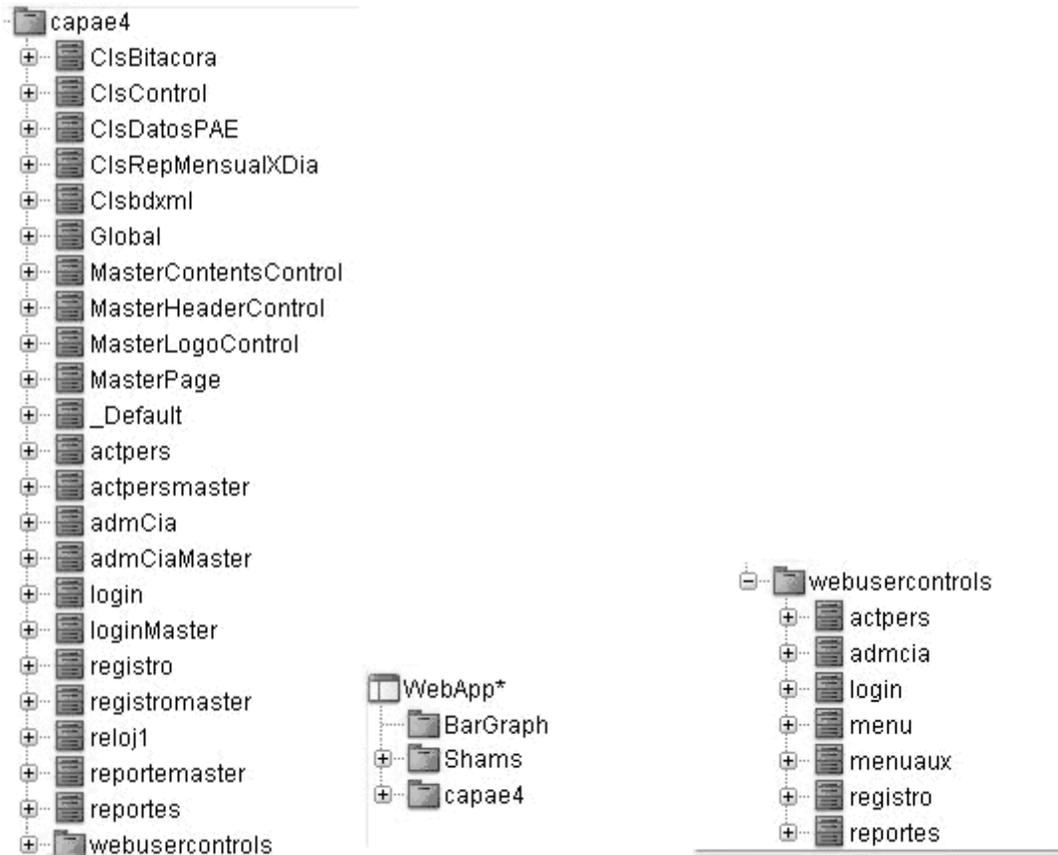
Registro: Se usa en comandos que modifican la asistencia.

registro
<pre>#Calendar1 : System.Web.UI.WebControls.Calendar #lblmensaje : System.Web.UI.WebControls.Label #panelES : System.Web.UI.WebControls.Panel #hhent : System.Web.UI.WebControls.DropDownList #mmment : System.Web.UI.WebControls.DropDownList #eampm : System.Web.UI.WebControls.DropDownList #sampm : System.Web.UI.WebControls.DropDownList #mmsal : System.Web.UI.WebControls.DropDownList #hhsal : System.Web.UI.WebControls.DropDownList #btnAceptar : System.Web.UI.WebControls.Button</pre>
<pre>-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -inicializa() : void -actualizaControles(ctrl : capae4.ClsControl, comando : string) : void -muestraControles(comando : string, FT : DateTime) : void -btnAceptar_Click(sender : object, e : System.EventArgs) : void -iniCtrlsTiempo(tiempo : string, hh : ListControl, mm : ListControl, ampm : ListControl) : void -Calendar1_SelectionChanged(sender : object, e : System.EventArgs) : void</pre>

Reportes: Visualiza reportes pedidos.

reportes
<pre>#lblmsg : System.Web.UI.WebControls.Label #panelsup : System.Web.UI.WebControls.Panel #BarGraphC1 : BarGraph.BarGraphC #DataGrid1 : System.Web.UI.WebControls.DataGrid #Table1 : System.Web.UI.WebControls.Table -ctrl : capae4.ClsControl #Table2 : System.Web.UI.WebControls.Table</pre>
<pre>-Page_Load(sender : object, e : System.EventArgs) : void #OnInit(e : EventArgs) : void -InitializeComponent() : void -creaReporte(comando : string) : void -grafica(tm : DataTable, usuarios : ArrayList, totmeshrs : double, mensual : bool, fecha : DateTime) : void -colorsel(n : int) : string</pre>

Todas las clases usadas en el CAPAE.



Anexo IV: Código relevante

MasterPageUserControl.ascx

```

<%@ Control Inherits="Shams.Web.UI.MasterPages.PageUserControl" %>
<asp:panel id="panelMain" HorizontalAlign="Left" Width="100%"
Height="26.2%" runat="server">
<TABLE id="Table1" height="18" cellSpacing="0" cellPadding="0"
width="100%" bgColor="#ffffcc" border="0">
  <TR>
    <TD width="1" bgColor="yellow" colSpan="1">
<asp:Placeholder id="placeholderLogo"
runat="server"></asp:Placeholder></TD>
    <TD bgColor="#66cc66">
      <asp:Placeholder id="placeholderHeader"
runat="server"></asp:Placeholder></TD>
    </TR>
  </TABLE>
<TABLE id="Table2" height="1" cellSpacing="1" cellPadding="1"
width="100%" bgColor="khaki" border="0">
  <TR>
    <TD bgColor="#ffccff" colSpan="1" height="1">
      <asp:Placeholder id="placeholderMenu"
runat="server"></asp:Placeholder></TD>
    </TR>
  </TABLE>
<TABLE id="Table3" height="1" cellSpacing="1" cellPadding="1"
width="100%" bgColor="khaki" border="0">
  <TR>
    <TD bgColor="#ffffcc" height="1">
      <asp:Placeholder id="placeholderSubMenu"
runat="server"></asp:Placeholder></TD>
    </TR>
  </TABLE>
<TABLE id="Table4" height="702" cellSpacing="1" cellPadding="0"
width="100%" border="0">
  <TR>
    <TD width="1" bgColor="white" height="100%">
      <asp:Panel id="panelNavigator" runat="server" Height="100%"
Width="100%" HorizontalAlign="Left" BackColor="#C0C0FF"
Wrap="False">
        <asp:Placeholder id="placeholderNavigation"
runat="server"></asp:Placeholder> </asp:Panel></TD>
    <TD bgColor="#ccffff" height="100%">
      <asp:Placeholder id="placeholderContents"
runat="server"></asp:Placeholder></TD>
    </TR>
  <TR>
    <TD width="1" bgColor="white" height="18">
      <asp:Placeholder id="placeholderSiteCounter"
runat="server"></asp:Placeholder></TD>
    <TD bgColor="#ffcc33" height="18">
      <asp:Placeholder id="placeholderFooter"
runat="server"></asp:Placeholder></TD>
    </TR>
  </TABLE>
</asp:panel>

```

Registro.ascx.cs (algunos procedimientos)

```

private void inicializa()
{
    // Inicializa segun comando (entrada, salida. entSal)
    ClsControl ctrl = new ClsControl(Session, Page);
    string comando = "";
    if (Request.QueryString["comando"] != null) {
        comando = Request.QueryString["comando"].ToString();
        if
        (!ctrl.permiso(comando)) {Response.Redirect(ctrl.paginaComando("inicio"), true);}
    }
    ctrl.comando = comando;
    actualizaControles(ctrl,comando);
}
private void actualizaControles(ClsControl ctrl, string comando)
{
    string err = "";
    string PAE = ctrl.pae;
    if (PAE.Length == 0)
    {
        err = "Falta seleccion de PAE";
        comando = "";
    }
    else
    {
        DateTime FT = ctrl.ft;
        string hract = DateTime.Now.ToShortTimeString();
        string Cia = ctrl.ciadePAE(PAE);
        lblmensaje.Text = FT.ToLongDateString() + "<br>";
        lblmensaje.Text += "<b>" + PAE + "</b><BR>CIA: " + Cia ;
        switch (comando)
        {
            case "entrada":
                err = ctrl.verificaRegistro(comando,FT,PAE,hract,"");
                if (err.Length == 0)
                {
                    ctrl.anotaDtsE(FT,PAE,hract);
                    lblmensaje.Text += "<BR>ENTRADA:<b> " + hract +
" </b>";
                }
                break;
            case "salida":
                err = ctrl.verificaRegistro(comando,FT,PAE,"",hract);
                if (err.Length == 0)
                {
                    ctrl.anotaDtsS(FT,PAE,hract);
                    lblmensaje.Text += "<BR>SALIDA:<b> " + hract +
" </b>";
                }
                break;
            case "entSal":
                string he;
                string hs;
                string lapso;
                ctrl.anotaDtsE(FT,PAE,hract);
                ctrl.obtenerRegistro(PAE,FT,out he,out hs,out lapso);
                if (he.Length > 0)
                { // inicializa entrada 00:00 a.m.
                    iniCtrlsTiempo(he,hhent,mment,eampm);
                    if (hs.Length > 0)
                    { // inicializa salida hr mm ampm

```

CAPAE

```
                iniCtrlsTiempo(hs,hhsal,mmsal,sampm);
            }
        }
        lblmensaje.Text += "<br>Indique Hora de Entrada, Hora
de Salida.";
        lblmensaje.Text += "<br>Y oprima boton Aceptar. ";
        break;
        default :
            err = "El llamado de este Página es incorrecto (falta
comando)";
            break;
    }
}
muestraControles(comando,ctrl.ft);
if (err.Length > 0)
{
    lblmensaje.Text = err;
    lblmensaje.ForeColor = System.Drawing.Color.Red;
    btnAceptar.Enabled = false;
}
else
{
    lblmensaje.ForeColor = System.Drawing.Color.Black;
    btnAceptar.Enabled = true;
}
ctrl = null;
}

private void muestraControles(string comando,DateTime FT)
{
    panelES.Visible = false;
    Calendar1.SelectedDate = FT;
    Calendar1.VisibleDate = FT;
    switch (comando)
    {
        case "entrada": btnAceptar.Text ="Aceptar Entrada";
            break;
        case "salida": btnAceptar.Text ="Aceptar Salida";
            break;
        case "entSal": panelES.Visible = true;
            btnAceptar.Text ="Aceptar Entrada y Salida";
            break;
        default :
            Calendar1.Visible = false;
            btnAceptar.Visible = false;
            break;
    }
}

private void btnAceptar_Click(object sender, System.EventArgs e)
{
    // segun comando (entrada, salida. entSal)
    ClsControl ctrl = new ClsControl(Session, Page);
    string comando = ctrl.comando;
    string nombre;
    DateTime fecha;
    string horaE;
    string horaS;
    string err = "";
    Application.Lock();
    Application["actualizando"] = comando;
    switch (comando)
    {
        case "entrada":
```

```

err = ctrl.obtenDtsE(out fecha,out nombre,out horaE);
if (err.Length == 0)
{
    err = ctrl.registro(comando, fecha, nombre, horaE, "");
}
break;
case "salida":
err = ctrl.obtenDtsS(out fecha,out nombre,out horaS);
if (err.Length == 0)
{
    err = trl.registro(comando, fecha, nombre, "", horaS);
}
break;
case "entSal":
// leer datos
err = ctrl.obtenDtsE(out fecha,out nombre,out horaE);
horaE = hhent.Selected.Value.ToString() +
":" + mmment.Selected.Value.ToString() +
" " + eampm.Selected.Value.ToString();
horaS = hhsal.Selected.Value.ToString() +
":" + mmsal.Selected.Value.ToString() +
" " + sampm.Selected.Value.ToString();
if (err.Length == 0)
{
    fecha = ctrl.ft;
err =
ctrl.verificaRegistro(comando, fecha, nombre, horaE, horaS);
if (err.Length == 0)
{
    err =
ctrl.registro(comando, fecha, nombre, horaE, horaS);
}
}
break;
default :
err = "El llamado de este Página es incorrecto (falta
comando)";
break;
}
Application.UnLock();
if (err.Length > 0)
{
    lblmensaje.Text = err;
    lblmensaje.ForeColor = System.Drawing.Color.Red;
}
else
{
    ctrl.comando = "";
    Response.Redirect(ctrl.paginaComando("inicio"));
}
ctrl = null;
}

```

registromaster.cs

```

public class registromaster : Shams.Web.UI.MasterPages.MasterPageBase
{
    public registromaster()
    {
        base.pageController.PageTitle = "CAPAE-Registro de Asistencia";
    }
}

```

CAPAE

```
        override public void AddPlaceholderControls()
        {
            ClearPlaceholderControls();
            base.AddPlaceholderControls();
        }
        override public void AddPlaceholderHeader()
        {
            System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/MasterHeaderControl.ascx");

            base.userPageControl.PlaceHolderHeader.Controls.Add(userControl);
        }
        override public void AddPlaceholderMenu()
        {
            System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/menu.ascx");

            base.userPageControl.PlaceHolderMenu.Controls.Add(userControl);
        }
        override public void AddPlaceholderSubMenu()
        {
            System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/menuaux.ascx");

            base.userPageControl.PlaceHolderMenu.Controls.Add(userControl);
        }
        override public void AddPlaceholderNavigation()
        {
        }
        override public void AddPlaceholderLogo()
        {
            System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/MasterLogoControl.ascx");

            base.userPageControl.PlaceHolderLogo.Controls.Add(userControl);
        }
        override public void AddPlaceholderContents() // La
Diferencia
        {
            System.Web.UI.UserControl userControl =
LoadUserControl("WebUserControls/registro.ascx");

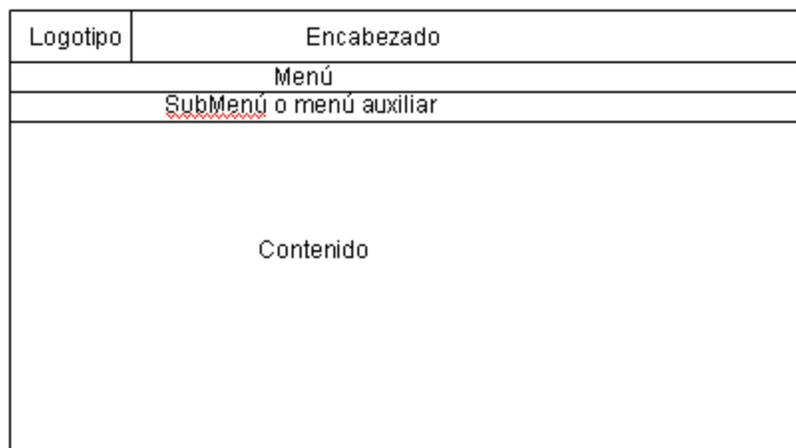
            base.userPageControl.PlaceHolderContents.Controls.Add(userControl);
        }
    }
```

Anexo V: Pantallas finales

En las siguientes imágenes se muestran algunas de las imágenes resultantes de usar los comandos del sistema. Para simplificar las imágenes, en algunas de ellas solo se puso la sección o detalle de la pantalla que cambia **con respecto a la pantalla inicial** (comando inicio), recuérdese que a tiempo de ejecución generalmente se verán, las siguientes **Secciones**:

- **Logotipo:** Con la imagen de la empresa.
- **Encabezado:** Texto con datos del proyecto y aplicación.
- **Menú:** Con los comandos (ligas) para que el usuario los use en cualquier momento.
- **Menú Secundario:** Para mostrar Usuario activo, la hora del sistema, y permitir la selección de elemento PAE y compañía.
- **Contenido:** Para mostrar los controles con los que interactúa el usuario y/o resultados de los comandos.

Prototipo General



La única excepción a lo anterior es la pantalla del comando “login”, en la que no se muestran los menús.

- login

The screenshot shows a web interface with a logo on the left and a title bar at the top. The title bar contains the text 'Captura de Asistencia del Personal Auxiliar Externo' and 'Tecnologías de la Información(LS)'. Below the title bar is a 'Login' form with two input fields for 'Usuario:' and 'Contraseña:', and an 'Aceptar' button.

- inicio (con Rol de Administrador).

The screenshot shows a web interface with a logo on the left and a title bar at the top. The title bar contains the text 'Captura de Asistencia del Personal Auxiliar Externo' and 'Tecnologías de la Información(LS)'. Below the title bar is a navigation menu with links: inicio, login, entrada, salida, entSal, repMes, repSem, repDia, alta, baja, modifica, ayuda, bajaOp, altaOp, altaCia, bajaCia, modCia, vieCia, bitacora. Below the navigation menu is a 'Hilario' section with a time display '02:12 p.m.', a 'PAE:' dropdown menu showing 'WENDY PAULIN MARTINEZ', and a 'SERVINEXT' dropdown menu. Below the 'Hilario' section is a date display '(11) Sábado, 24 de Febrero de 2007'. Below the date display is a table with columns: Nombre, Cia, Entrada, Salida, Lapso. The table contains 11 rows of data.

Febrero de 2007							Nombre	Cia	Entrada	Salida	Lapso
D	L	M	M	J	V	S					
28	29	30	31	1	2	3	WENDY PAULIN MARTINEZ	CEPRA			
	4	5	6	7	8	9	ISRAEL VILLA MORALES	CEPRA			
	11	12	13	14	15	16	JAVIER CEDILLO ITURIBE	CEPRA			
	18	19	20	21	22	23	GUILLERMO CASTRO BARBEYTO	CEPRA			
	25	26	27	28	1	2	FERNANDO GALLARDO ALONSO	LANWARE			
	4	5	6	7	8	9	ROBERTO ESPINOSA	LANWARE			
						10	YOLANDA PADILLA ANGELES	LANWARE			
							GUADALUPE IVÁN VELÁSQUEZ SÁMANO	CEPRA			
							AVRIL PEREZ MUÑOZ	UVM			
							OLMO CASTANEIRA RIOS	POLI			
							CHRISTIAN ARIZANDY RODRIGUEZ MEDINA	CUE			

- **inicio** (con Rol Operador detalle, obsérvese el menú.)

- **inicio** (con Rol Invitado detalle, obsérvese el menú)

- **entrada** (Secciones Menú Auxiliar y Contenido).

- **entrada** (Después de Clic a botón “Aceptar Entrada” muestra inicio con hora entrada modificada para elemento PAE, Detalle)

- **salida** (Secciones Menú Auxiliar y Contenido)

hilario **02:44 p.m.** PAE: WENDY PAULIN MARTINEZ

Febrero de 2007						
D	L	M	M	J	V	S
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	24
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>

Sábado, 24 de Febrero de 2007
WENDY PAULIN MARTINEZ
 CIA: CEPRA
 SALIDA: **02:44 p.m.**

- **salida** (Detalle, Después de “Aceptar Salida” muestra inicio con hora salida cambiada, Detalle)

hilario **02:46 p.m.** PAE: WENDY PAULIN MARTINEZ

(11) Sábado, 24 de Febrero de 2007

Febrero de 2007							Nombre	Cia	Entrada	Salida	Lapso
D	L	M	M	J	V	S					
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>	WENDY PAULIN MARTINEZ	CEPRA	02:26 p.m.	02:44 p.m.	00:18
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	ISRAEL VILLA MORALES	CEPRA			
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	JAVIER CEDILLO ITURIBE	CEPRA			

- **entSal** (Secciones Menú Auxiliar y Contenido)

hilario **02:50 p.m.** PAE: WENDY PAULIN MARTINEZ

Febrero de 2007						
D	L	M	M	J	V	S
<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>
<u>11</u>	<u>12</u>	<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>
<u>18</u>	<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	23	24
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>1</u>	<u>2</u>	<u>3</u>
<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>

Viernes, 23 de Febrero de 2007
WENDY PAULIN MARTINEZ
 CIA: CEPRA
 Indique Hora de Entrada, Hora de Salida.
 Y oprima boton Aceptar.

Entro: 08 : 30 a.m.
 Salio: 02 : 00 p.m.

Al “Aceptar Entrada y Salida” muestra pantalla inicial.

- **repMes**(Secciones Menú Auxiliar y Contenido)

hilario	02:57 p.m.	PAE:	WENDY PAULIN MARTINEZ	▼	LANWARE	▼
EN MES: Dic/2006 Horario de 08:00 a.m. a las 06:00 p.m.						
dd	0-WENDY	1-ISRAEL	2-JAVIER	3-GUILLERMO	4-GUADALUPE	
01-v	00:00	00:00	00:00	00:00	00:00	
04-l	09:07	04:00	09:07	10:00	04:00	
05-m	07:54	04:00	09:20	07:54	04:00	
06-m	10:00	04:00	10:00	09:53	04:00	
07-j	09:41	00:00	09:00	09:44	09:00	
08-v	09:13	00:00	09:13	09:30	00:00	
11-l	10:00	00:00	09:32	09:49	00:00	
12-m	10:00	00:00	09:50	10:00	10:00	
13-m	10:00	00:00	09:53	10:00	03:36	
14-j	09:58	00:00	09:31	00:00	09:00	
15-v	09:50	00:00	09:49	10:00	00:00	
18-l	09:45	00:00	09:45	10:00	09:00	
19-m	08:34	00:00	08:34	08:33	08:48	
20-m	09:47	00:00	09:20	09:50	09:30	
21-j	09:55	00:00	09:54	09:59	08:59	
22-v	03:58	00:00	03:59	03:58	04:00	
25-l	00:00	00:00	00:00	00:00	00:00	
26-m	00:00	00:00	00:00	00:00	00:00	
27-m	00:00	00:00	00:00	00:00	00:00	
28-j	00:00	00:00	00:00	00:00	00:00	
29-v	00:00	00:00	00:00	00:00	00:00	
TOT	137	12	136	129	83	

EN MES: Dic/2006	
WENDY	
ISRAEL	
JAVIER	
GUILLERMO	
GUADALUPE	
hrs-mes:210	

repDia (Secciones menú auxiliar y contenido)

horario **03:07 p.m.** PAE: WENDY PAULIN MARTINEZ CEPRA

Horario de 08:00 a.m. a las 05:00 p.m. y los Viernes: 08:00 a.m.-03:00 p.m.

ASISTENCIA DEL MES DE Diciembre DE 2006

NOMBRE	01 - Dic - 06		02 - Dic - 06		03 - Dic - 06		04 - Dic - 06		05 - Dic - 06	
.....	ENTRADA	SALIDA								
FERNANDO GALLARDO ALONSO									AUS	AUS
ROBERTO ESPINOSA									AUS	AUS
YOLANDA PADILLA ANGELES									AUS	AUS
NOMBRE	04 - Dic - 06		05 - Dic - 06		06 - Dic - 06		07 - Dic - 06		08 - Dic - 06	
.....	ENTRADA	SALIDA								
FERNANDO GALLARDO ALONSO	00:00	12:00	00:00	12:00	00:00	12:00	AUS	AUS	AUS	AUS
ROBERTO ESPINOSA	00:00	12:00	00:00	12:00	00:00	12:00	AUS	AUS	AUS	AUS
YOLANDA PADILLA ANGELES	08:04	17:19	08:06	17:49	08:00	18:24	08:01	18:33	08:09	17:13
NOMBRE	11 - Dic - 06		12 - Dic - 06		13 - Dic - 06		14 - Dic - 06		15 - Dic - 06	
.....	ENTRADA	SALIDA								
FERNANDO GALLARDO ALONSO	AUS	AUS								
ROBERTO ESPINOSA	AUS	AUS								
YOLANDA PADILLA ANGELES	08:22	17:10	08:06	17:20	08:07	17:06	08:06	17:20	08:16	15:25
NOMBRE	18 - Dic - 06		19 - Dic - 06		20 - Dic - 06		21 - Dic - 06		22 - Dic - 06	
.....	ENTRADA	SALIDA								
FERNANDO GALLARDO ALONSO	AUS	AUS								
ROBERTO ESPINOSA	AUS	AUS	AUS	AUS	AUS	AUS	08:20	18:00	08:22	17:00
YOLANDA PADILLA ANGELES	08:07	17:15	08:17	17:22	08:04	17:16	AUS	AUS	AUS	AUS
NOMBRE	25 - Dic - 06		26 - Dic - 06		27 - Dic - 06		28 - Dic - 06		29 - Dic - 06	
.....	ENTRADA	SALIDA								
FERNANDO GALLARDO ALONSO	AUS	AUS								
ROBERTO ESPINOSA	AUS	AUS								
YOLANDA PADILLA ANGELES	AUS	AUS	08:36	18:29	08:24	17:25	08:24	18:00	08:23	AUS

- **alta** (Secciones Menú Auxiliar y Contenido)

hilario **03:11 p.m.** PAE: WENDY PAULIN MARTINEZ ▼ CEPRA ▼

Ponga Nombre del PAE y Oprima botón Aceptar

Nombre:

Cia:

Al “Aceptar” se muestra la pantalla inicial, con los cambios ya reflejados.

- **ayuda** (abre nueva ventana)

[Introduccion](#)
[Pagina Principal](#)
[Comandos](#)
[Acerca de](#)

Acerca del CAPAE (v 4.0)

El Grupo de Paginas mejor conocidas como CAPAE(Captura de Asistencia del Personal Auxiliar Externo), fueronGeneradas con tecnología ASP.NET, inicialmente para uso del Laboratorio de Software de TI, a peticio del Ing. Roberto Flores.

Objetivo Principal: Supervisar la Asistencia del PAE.

Características Generales:

- Enfocado a Nivel de Proyecto.
- De una manera simple permite supervisar la asistencia del PAE al Centro de Trabajo.
- Por ser una aplicación ASP.NET es Accesible desde cualquier punto de la Intranet.
- Maneja Roles: Administrador, Capturista y Anónimo (solo consulta)
- Agrupa al PAE por empresa, Para Becarios y Personal de Honorariosse usaria un nombre ficticio..
- Internamente almacena los datos en Formato XML, para posterior explotacion.

Funciones:

- Consulta de Asistencia
- Registro de Entrada/Salida.
- Muestra Reportes Semanal o mensual de Horas Trabajadas.
- Validar Administrador o Capturista para la sesión.
- Actualizar Nombre y Empresa del Personal (Alta, Baja Modificar)
- Actualizar Datos las Empresas: Nombre y Horario (Alta, Baja Modificar)

Al usar cualquiera de los enlaces de la izquierda el contenido correspondiente es mostrado en el panel derecho.

- **altaOp** (Sección contenido)

Indique Nombre y contraseña del Operador y Clic a Aceptar	
Operador:	<input type="text" value="Raul"/>
Contraseña:	<input type="password" value="••••••"/>
<input type="button" value="Aceptar Alta Operador"/>	

- **altaCia** (Sección contenido)

Ponga el Horario de la Compañía, su Nombre y Oprima el boton Aceptar		
Entrada:	<input type="text" value="08"/> <input type="text" value="00"/> <input type="text" value="a.m."/>	
Salida:	<input type="text" value="05"/> <input type="text" value="00"/> <input type="text" value="p.m."/>	
Compañía:	<input type="text" value="ACME"/>	
<input type="button" value="Aceptar Nueva Cia"/>		

- **vieCia** (Secciones menú auxiliar y contenido)

Hilario	<input type="text" value="12:25 p.m."/>	P.AE:	<input type="text" value="ALEJANDRO RAMIREZ TAPIA"/>	<input type="text" value="ORSA"/>
Se Modificara al Oprimir el boton Aceptar				
Entrada:	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="a.m."/>			
Salida:	<input type="text" value="00"/> <input type="text" value="00"/> <input type="text" value="a.m."/>			
Compañía:	<input type="text" value="ORSA"/>			
<input type="button" value="Aceptar Horario Viernes"/>				

- bitácora (Sección contenido)

Activades registradas en el mes:	
fecha-hora	Registro
18/01/2007 08:18 a.m.	'Se Crea Archivo de Log'
18/01/2007 08:18 a.m.	'192.168.144.130,login,18/01/2007,INES,Operador'
18/01/2007 09:42 a.m.	'192.168.144.130,login,18/01/2007,INES,Operador'
18/01/2007 01:33 p.m.	'192.168.144.115,login,18/01/2007,hilario,Operador'
18/01/2007 01:38 p.m.	'192.168.144.115,login,18/01/2007,Hilario,Administrador'
19/01/2007 07:54 a.m.	'192.168.144.130,login,19/01/2007,INES,Operador'
22/01/2007 08:00 a.m.	'192.168.144.130,login,22/01/2007,INES,Operador'
22/01/2007 08:50 a.m.	'192.100.181.86,login,22/01/2007,Ireyna,Invitado'
22/01/2007 08:54 a.m.	'192.100.181.86,login,22/01/2007,,Invitado'
23/01/2007 08:11 a.m.	'192.168.144.130,login,23/01/2007,INES,Operador'
23/01/2007 08:18 a.m.	'192.100.181.86,login,23/01/2007,,Invitado'
23/01/2007 09:41 a.m.	'192.100.181.86,login,23/01/2007,Ireyna,Invitado'
24/01/2007 08:04 a.m.	'192.168.144.130,login,24/01/2007,INES,Operador'
24/01/2007 08:18 a.m.	'INES,entSal,23/01/2007,HECTOR MANUEL GARCIA MORALES,08:10 a.m.-06:00 p.m.'
24/01/2007 08:18 a.m.	'INES,entSal,23/01/2007,ALEJANDRO RAMIREZ TAPIA,08:00 a.m.-06:00 p.m.'
24/01/2007 08:18 a.m.	'INES,entSal,23/01/2007,OSWALDO ESPINOSA DE LOS MONTEROS MORALES,08:00 a.m.-06:00 p.m.'
24/01/2007 08:18 a.m.	'INES,entSal,23/01/2007,CESAR ALEJANDRO VILLEGAS CASTRO,08:11 a.m.-06:00 p.m.'
24/01/2007 08:19 a.m.	'INES,entSal,23/01/2007,SALVADOR MENDEZ RODRIGUEZ,08:17 a.m.-06:00 p.m.'
24/01/2007 08:20 a.m.	'INES,entSal,22/01/2007,SALVADOR MENDEZ RODRIGUEZ,08:03 a.m.-06:00 p.m.'
24/01/2007 08:20 a.m.	'INES,entSal,22/01/2007,OSWALDO ESPINOSA DE LOS MONTEROS MORALES,08:13 a.m.-06:00 p.m.'
24/01/2007 08:21 a.m.	'INES,entSal,22/01/2007,CESAR ALEJANDRO VILLEGAS CASTRO,08:24 a.m.-06:00 p.m.'
24/01/2007 08:21 a.m.	'INES,entSal,22/01/2007,HECTOR MANUEL GARCIA MORALES,08:11 a.m.-06:00 p.m.'
24/01/2007 08:21 a.m.	'INES,entSal,22/01/2007,ALEJANDRO RAMIREZ TAPIA,08:00 a.m.-06:00 p.m.'
24/01/2007 08:21 a.m.	'INES,entSal,19/01/2007,ALEJANDRO RAMIREZ TAPIA,07:54 a.m.-03:00 p.m.'

REFERENCIAS

[ARLOW1] *UML 2 and the Unified Process* 2da ED, Jim Arlow, Ila Neustadt . Addison Wesley, 2005, ISBN 0-321-32127-8

[BRAUDE1] *Ingeniería de Software*, Eric J Braude ,Alfaomega, 2003, ISBN 970-15-0851-3

[EVANS1] *Guía en 10 minutos HTML 4*, Tim Evans, Prentice Hall, 1998, ISBN 84-8322-071-7

[GUIANDIN1] Revista Colombiana de Computación Relaciones entre casos de uso de UML ,Roxana S Giandin, Vol.1 Num.1 Año 2000, http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r11_art4_c.pdf

[JACOBSON1] *Object Oriented Software Engineering*, Ivar Jacobson, ACM Press Addison Wesley, 1992, ISBN 0201544350

[JAVAHISPANO1]

http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/?menuId=&onlypath=true,
Procesos de Desarrollo, Alberto Molpeceres, 2003

[KIMMEL1] *Manual de UML*, Paul Kimmel, Mc Graw Hill , 2007, ISBN 970-10-5899-2

[MUKHTAR1] *An Extensible Master-Page Framework for ASP.NET 1.1 Using Pattern Oriented Design*, Shams Mukhtar, http://www.codeproject.com/aspnet/Frame_Work_Design.asp

[ORTIN1] *El Modelo de Negocio como base del modelo de Requisitos*, María José Ortín, Jesús García Molina, Begoña Moros, Joaquín Nicolás, Facultad de Informática Universidad de Murcia. Ponencia, 2000.

[ROSENBERG1] *Use Case Driven Object Modeling with UML*, Rosenberg et al, Addison Wesley, 1999, ISBN 0201432897

[WEBOMG1] www.omg.mda, 2007

[WEBIMP1] <http://www.imp.mx/acerca/>, Acerca del IMP, Instituto Mexicano del Petroleo,2005

[WIKIPEDIA1] http://es.wikipedia.org/wiki/Modelo_de_negocio/,2006

[WIKIPEDIA2] http://es.wikipedia.org/wiki/Beta_testing