



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UN SISTEMA
PARA EL CONTROL Y GRAFICACION DE
PLANOS DEL INSTITUTO MEXICANO DEL
PETROLEO

T E S I S
PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A:

JORGE ALBERTO MORENO HERNANDEZ

DIRECTOR DE TESIS:
M. C. JESUS SOSA IGLESIAS

CO-DIRECTOR DE TESIS:
DR. JESUS SAVAGE CARMONA

MEXICO, D. F.

2002



TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIAS

A mis padres, Estela y Miguel Angel, por todos los consejos y por todo el apoyo que me han entregado de forma incondicional en toda mi vida, agradeciendo todos los sacrificios que realizaron para lograr la culminación de este trabajo. Siempre estaré agradecido con ustedes.

A mis tíos, María Elena y Luis Manuel, por brindarme su confianza y darme la oportunidad para poder desarrollar este trabajo de tesis.

A mi novia, Georgina, por todo el amor y todo el apoyo que he recibido de su parte en innumerables ocasiones. Gracias.

RECONOCIMIENTO

Agradezco el apoyo brindado de mi director de tesis el M.C. Jesús Sosa Iglesias, por la ayuda, las correcciones y las observaciones mostradas durante la realización del presente trabajo de tesis y por todo el conocimiento que obtuve por parte de él.

Índice

DEDICATORIAS.....	i
RECONOCIMIENTO.....	ii
ÍNDICE DE FIGURAS.....	v
INTRODUCCIÓN.....	vi
1. JUSTIFICACIÓN.....	1
RESUMEN.....	1
1.1. EL PROCESO DE GRAFICACIÓN ACTUAL DEL IMP.....	2
1.2. DEFICIENCIAS DEL PROCESO DE GRAFICACIÓN ACTUAL DEL IMP.....	3
1.3. PROPUESTA DE SOLUCIÓN.....	4
1.4. VENTAJAS DE UN NUEVO SISTEMA DE GRAFICACIÓN Y CONTROL.....	4
2. ANÁLISIS Y METODOLOGÍA DE DESARROLLO.....	5
RESUMEN.....	5
2.1. LISTA DE REQUERIMIENTOS.....	6
2.2. RECURSOS A EMPLEAR.....	7
2.3. TECNOLOGÍA A UTILIZAR.....	7
2.4. ARQUITECTURA DEL SISTEMA.....	8
3. DISEÑO DEL SISTEMA DE CONTROL Y GRAFICACIÓN.....	10
RESUMEN.....	10
3.1. DISEÑO DEL SISTEMA.....	11
3.1.1. MÓDULOS Y ACTORES.....	12
3.2. DISEÑO DE LA BASE DE DATOS.....	12
3.2.1. DIAGRAMA ENTIDAD-RELACIÓN.....	12
3.3. DIAGRAMAS DEL SISTEMA.....	14
3.3.1. DIAGRAMA DE CASOS DE USO.....	14
3.3.2. DIAGRAMA DE CLASES.....	18
3.3.3. DIAGRAMAS DE SECUENCIAS.....	20
3.4. DISEÑO DE INTERFACES GRÁFICAS DE USUARIO.....	32

4. CONSTRUCCIÓN DEL SISTEMA DE CONTROL Y GRAFICACIÓN.....	33
RESUMEN	33
4.1. CREACIÓN DE LA BASE DE DATOS	34
4.2. PROGRAMACIÓN DE LAS INTERFACES GRÁFICAS DE LA APLICACIÓN	35
4.3. VALIDACIÓN DE FORMULARIOS HTML.....	49
4.4. CONSTRUCCIÓN DE LAS CLASES DE OBJETOS DEL SISTEMA.....	51
4.4.1. DESCRIPCIÓN DE LA CLASE ADMINISTRADOR	51
4.4.2. DESCRIPCIÓN DE LA CLASE CONSULTAS.....	52
4.4.3. DESCRIPCIÓN DE LA CLASE DESCRIPCION.....	53
4.4.4. DESCRIPCIÓN DE LA CLASE DESPACHADOR	54
4.4.5. DESCRIPCIÓN DE LA CLASE GRAFICAPLANO.....	55
4.4.6. DESCRIPCIÓN DE LA CLASE ORDEN.....	56
4.4.7. DESCRIPCIÓN DE LA CLASE USUARIO SISTEMA.....	58
5. PRUEBAS FINALES E INSTALACIÓN.....	59
RESUMEN	59
5.1. PRUEBAS AL SISTEMA DE CONTROL Y GRAFICACIÓN	60
5.2. INSTALACIÓN DEL NUEVO SISTEMA DE CONTROL Y GRAFICACIÓN.....	69
6. CONCLUSIONES.....	74
RESUMEN	74
6.1. CONCLUSIONES	75
6.2. TRABAJOS FUTUROS.....	77
BIBLIOGRAFIA.....	79

Índice de Figuras

Pág.

Figura 2.1	Arquitectura del Sistema de Control y Graficación.....	8
Figura 3.1	Diseño de Módulos y Actores del Sistema.....	12
Figura 3.2	Diagrama Entidad-Relación.....	13
Figura 3.3	Diagrama de Casos de Uso.....	15
Figura 3.4	Diagrama de Clases del Sistema.....	18
Figura 3.5	Diagrama de Secuencias del Caso de Uso Autenticar.....	22
Figura 3.6	Diagrama de Secuencias del Caso de Uso Registrar Usuario.....	24
Figura 3.7	Diagrama de Secuencias del Caso de Uso Crear una Orden de Graficación.....	26
Figura 3.8	Diagrama de Secuencias del Caso de Uso Generar Reportes de Graficación.....	28
Figura 3.9	Diagrama de Secuencias del Caso de Uso Graficar Planos de Ordenes Pendientes.....	31
Figura 4.1	Archivo del esquema de la base de datos.....	34
Figura 4.2	Página de Inicio del Sistema de Control y Graficación.....	36
Figura 4.3	Página de autenticación del usuario.....	37
Figura 4.4	Selección de competencia.....	38
Figura 4.5	Selección de nombre del empleado.....	38
Figura 4.6	Escritura de contraseña.....	39
Figura 4.7	Comienzo de elaboración de una Orden de Graficación.....	40
Figura 4.8	Características de un plano solicitado.....	41
Figura 4.9	Estado de la orden de graficación.....	41
Figura 4.10	Indicación del número de orden.....	42
Figura 4.11	Opciones de trabajo del Administrador.....	43
Figura 4.12	Presentación de los criterios de búsqueda.....	44
Figura 4.13	Asignación de valor a los criterios seleccionados.....	45
Figura 4.14	Resultados de la búsqueda procesada.....	45
Figura 4.15	Presentación de ordenes de graficación pendientes.....	46
Figura 4.16	Descripción de orden de graficación seleccionada.....	47
Figura 4.17	Pregunta del Sistema sobre el proceso de graficación.....	48
Figura 4.18	Cambio de la columna Estado en la descripción de la orden.....	48
Figura 4.19	Envío de correo electrónico.....	49
Figura 4.20	Validación de datos con JavaScript del lado del Cliente.....	50
Figura 4.23	Diagrama de Clase <i>Administrador</i>.....	52
Figura 4.24	Diagrama de Clase <i>Consultas</i>.....	53
Figura 4.25	Diagrama de Clase <i>Descripcion</i>.....	54
Figura 4.26	Diagrama de Clase <i>Despachador</i>.....	55
Figura 4.27	Diagrama de Clase <i>graficaPlano</i>.....	56
Figura 4.28	Diagrama de Clase <i>Orden</i>.....	57
Figura 4.29	Diagrama de Clase <i>UsuarioSistema</i>.....	58
Figura 6.1	Arquitectura del Sistema de Control y Graficación de Planos.....	78
Figura 6.2	Arquitectura propuesta.....	78

Introducción

El objetivo del presente trabajo de tesis es la construcción de un sistema automatizado de graficación de planos para el Instituto Mexicano del Petróleo (IMP), documentar el sistema y proveer de un servicio de graficación automatizado para el IMP.

En el Capítulo 1 se hace una justificación del trabajo de ésta tesis, donde se mencionan las características del antiguo sistema de graficación y sus deficiencias, planteando un objetivo y la importancia de realizar un nuevo sistema de graficación que proporcione una solución a los problemas que actualmente se presentan al graficar planos que fueron generados electrónicamente.

En el Capítulo 2 se inicia propiamente la etapa de análisis de requerimientos para el desarrollo del nuevo sistema de graficación, se habla de las necesidades que se deben cubrir, los recursos con los que se cuenta, se establece un plan de trabajo y se mencionan las herramientas de desarrollo para el nuevo sistema.

En el Capítulo 3 se muestra el modelado de las clases y objetos del sistema, el diseño de la base de datos y la documentación del sistema mediante diagramas, para que en la etapa de construcción de la aplicación esté lo suficientemente documentada.

En el Capítulo 4 se describen los procesos importantes que el sistema automatiza durante la etapa de construcción del sistema, se construye una base de datos, y se hace la programación de las graficas y objetos.

En el Capítulo 5 se reportan los resultados correspondientes a las pruebas finales al sistema, a la instalación y configuración del mismo, se describen los distintos sistemas periféricos que apoyan al sistema de graficación en su operación, así como la importancia que tienen para mejorar el servicio.

Finalmente, el Capítulo 6 está dedicado a las conclusiones, en donde se mencionan los logros obtenidos y los trabajos a futuro que de este trabajo se pueden obtener.

1

Justificación

Resumen

En este capítulo se presenta una descripción de la problemática a resolver con este trabajo de tesis y se rescatan los problemas asociados a la forma no automatizada de graficación, se plantea la necesidad de un sistema automático de graficación y monitoreo de planos para el Instituto Mexicano del Petróleo y el objetivo de construir dicho sistema.

1.1. El proceso de graficación actual del IMP

Dentro de las instalaciones del Instituto Mexicano del Petróleo (IMP) existe la competencia de Ingeniería Asistida por Computadora, dentro de la cual se encuentra ubicada el área de graficación de planos, donde cualquier empleado del Instituto puede solicitar que se le grafique uno o varios planos con cargo a los proyectos vigentes. El método que se utiliza para la graficación de planos se describe a continuación:

- 1) La persona que solicita planos para ser graficados debe llenar una solicitud de graficación por escrito, la solicitud se divide en dos partes, en la primera establece datos que describen la competencia en donde labora, la gerencia a la que pertenece, la división, el departamento, etc. En la segunda parte de la solicitud la persona proporciona la descripción del plano o de los planos que desea sean graficados, como no. del plano, nombre del archivo a graficar, tamaño, escala, cantidad de duplicados, material, no. de proyecto, etc., además, la solicitud contiene la fecha y el nombre del solicitante.
- 2) Posteriormente al llenado de la solicitud de graficación, el solicitante proporciona el medio magnético donde se encuentran los archivos que desea sean graficados, y cuyas características se establecieron en la solicitud de graficación.
- 3) Una vez que se proporcionaron todos los datos por parte del solicitante, existe una persona que es la encargada de procesar todas las ordenes de graficación de planos que se solicitan, revisando cada orden de graficación y su correspondiente medio magnético. Se verifican las características de los planos, su localización en el medio magnético, se ajustan los parámetros del graficador manualmente y se envían los archivos al graficador para obtener el plano.
- 4) Finalmente, una vez realizado el proceso de graficación, el solicitante pasa al área de graficación a recoger sus planos.
- 5) Mensualmente se elabora un reporte manual del número de planos graficados por proyecto.

1.2. Deficiencias del proceso de graficación actual del IMP

La forma en que se realiza el proceso de graficación tiene algunos puntos problemáticos de los cuales se citan algunos a continuación:

- Por la característica de que el proceso actual es un sistema manual, no se lleva un control real que cubra todas las características de los planos que se solicitan graficar, por ejemplo, los proyectos a los que pertenecen dichos planos, nunca se verifica si los planos graficados corresponden a proyectos reales vigentes, la cantidad de planos graficados por proyecto, características descriptivas de los planos, así como todas las características administrativas del personal, como nombre, proyecto, jefe de proyecto, etc. Con la utilización de una base de datos se pretende realizar un control estricto y la facilidad de elaborar reportes en base a cualquiera de las características antes señaladas.
- Es problemática la forma en que se realiza la solicitud de graficación, principalmente por el tiempo perdido en realizar la solicitud, debido al desplazamiento que el personal tiene que hacer hasta el área de graficación. Con la realización de un nuevo sistema, accesible desde la Intranet del IMP, cualquier persona podrá elaborar una solicitud de graficación en muy poco tiempo y sin tener que realizar algún desplazamiento, pudiéndolo hacer desde su navegador de Web.
- El proceso actual está limitado solo para la gente que está laborando dentro de las instalaciones del IMP, pero existe gente que pertenece al IMP que se encuentra de comisión, es decir, que labora en un proyecto fuera de las instalaciones del Instituto, que no tiene oportunidad de hacer uso del sistema de graficación con el que se cuenta. Con la realización de un nuevo sistema de control y graficación accesible desde la Intranet, el sistema se vuelve accesible automáticamente desde Internet, siempre y cuando sea personal del IMP el que intente acceder.
- La autenticación de usuario solo es validada por la firma del jefe inmediato.

1.3. Propuesta de solución

En virtud de la problemática anterior se hace necesario plantear el diseño y la construcción de un sistema automatizado de graficación de planos, que pueda ser utilizado a través de la Intranet del IMP por cualquier empleado del Instituto y que valide la autenticidad de los proyectos a los que se hacen cargos por graficación, además, que guarde un registro de la cantidad de planos procesados y graficados para poder generar diferentes tipos de reportes periódicamente.

Y, como objetivo de este trabajo de tesis, se tiene al Análisis, Diseño y Construcción de este sistema automatizado.

1.4. Ventajas de un nuevo sistema de graficación y control

- Validación de proyectos.
- Con la utilización de una base de datos se pretende realizar un control estricto y la facilidad de elaborar reportes en base a cualquiera de las características antes señaladas.
- Con la realización de un nuevo sistema, accesible desde la Intranet del IMP, cualquier persona podrá elaborar una solicitud de graficación en muy poco tiempo y sin tener que realizar algún desplazamiento, pudiéndolo hacer desde su navegador de web.
- Gracias a la Intranet del IMP, el sistema se vuelve accesible automáticamente desde Internet, siempre y cuando sea personal del Instituto el que intente acceder.

2

Análisis y metodología de desarrollo

Resumen

En este capítulo se describen las bases del diseño y los requerimientos que deberá de cumplir el sistema de graficación, se mencionan los recursos con los que se cuenta, se propone una arquitectura de desarrollo para el nuevo sistema y se describen las herramientas de software que se utilizan para el desarrollo del sistema de graficación y control de planos.

2.1. Lista de requerimientos

Las bases del diseño y los requerimientos del sistema de graficación incluyen:

- El proceso de solicitud de graficación, puede ser realizado por cualquier persona que labora dentro de las instalaciones del IMP, debe ser totalmente automatizado por el sistema, a través de la Intranet con que cuenta la Institución. El proceso de validación de usuario será acorde a los empleados registrados en el ERP (Enterprise Resource Planning) de SAP [SAP].
- Reducir los problemas de transferencia de archivos al momento de solicitar una orden de graficación.
- El sistema de graficación debe tener la capacidad de llevar un control estricto de todos los planos graficados de acuerdo a cada una de sus características, como pueden ser: número de copias graficadas, tamaño de los planos, proyectos a los que pertenecen, nombre de la persona solicitante, jefe de proyecto, fecha de graficación y características descriptivas de los planos.
- Modalidad de administración para poder realizar diferentes tipos de reportes de acuerdo a cada una de las características antes mencionadas.
- Accesibilidad del sistema de graficación desde el interior y el exterior del Instituto Mexicano del Petróleo por medio de un navegador Web, con soporte de seguridad para el acceso a la aplicación.

Para cumplir con estos requerimientos el sistema deberá contemplar:

- Alimentar una base de datos con las operaciones diarias de solicitud y procesamiento de ordenes de graficación con el nivel de detalle especificado en los requerimientos.
- Integración de la aplicación con información real de empleados y proyectos de acuerdo a la información del IMP contenida en SAP.

De esta forma los requerimientos que necesita satisfacer el IMP serán automatizados por el sistema, para apoyar al área de graficación en la carga de trabajo diario.

2.2. Recursos a emplear

Para evitar gastos excesivos en el licenciamiento de software en la operación del sistema, se optó por utilizar tecnologías de Internet de distribución libre. A continuación se presentan los principales recursos para el desarrollo del sistema.

- Acceso a una instancia de un Manejador de Bases de Datos Relacional.
- JDK 1.3.1(Java Development Kit) para el desarrollo de la aplicación [Jso02].
- Jakarta Tomcat 3.2.3 como contenedor de Servlets y JSP [JtoA01].
- Apache Web Server como servidor de web [Apa01].
- Together-J como herramienta utilizada para el modelado del sistema con UML (Lenguaje de Modelado Unificado) [Tog02].
- Se cuenta con un graficador CalComp DrawingMaster Modelo 600 para el proceso de graficación de planos, conectado mediante el puerto paralelo a una computadora tipo PC.
- Bibliografía especializada en Java, Java Servlets, JSP, JDBC, SQL, HTML, JavaScript, Tomcat y Apache Web Server.
- Computadora Pentium III con Linux o Windows NT, 128 MB de memoria y 4 GB en disco duro.

2.3. Tecnología a utilizar

Las tecnologías a utilizar dentro del desarrollo del sistema son:

Java Servlets [Hala01], puesto que los servlets tienen un marco de trabajo similar al lenguaje de programación Java, son independientes de la plataforma y están soportados en la mayoría de servidores web, los Servlets, una vez que son llamados por primera vez, quedan activos en la memoria del servidor, logrando con esto una mayor eficiencia en cuanto a tiempo de respuesta hacia el usuario del sistema, además, los Servlets se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, permitiendo con esto el desarrollo de aplicaciones distribuidas y fácilmente escalables. De igual forma, los servlets permiten redireccionar peticiones de servicios a otros servlets, en la misma máquina o en una máquina remota. Debido a todas las ventajas que aportan los

Servlets, el programa que realice la función de controlador dentro de nuestro sistema será un Servlet.

JavaServer Pages (JSP) [HalB01], la tecnología de las páginas JSP es transportable entre sistemas operativos y servidores web, la utilización de páginas JSP nos permite la posibilidad de construir las interfaces gráficas de usuario mediante formularios HTML, acceso a bases de datos u objetos distribuidos del lado del servidor. Debido a la facilidad con que se puede manejar HTTP y formularios con las páginas JSP, todas las interfaces que sean presentadas ante el usuario serán implementadas con la utilización de esta tecnología.

JavaBeans [HalC01], es la tecnología de Java para la creación de componentes de software reutilizables, además de que el uso de JavaBeans junto con Servlets y JSP nos permite el desarrollo de un sistema que separe la lógica de la aplicación de su presentación, logrando con esto una forma más limpia de codificación y de más fácil mantenimiento.

2.4. Arquitectura del sistema

La Figura 2.1. muestra el modelo de la arquitectura del sistema, la cual se basa en el patrón de diseño Modelo/Vista/Controlador (MVC), ó Modelo 2 de Programación [Paw01].

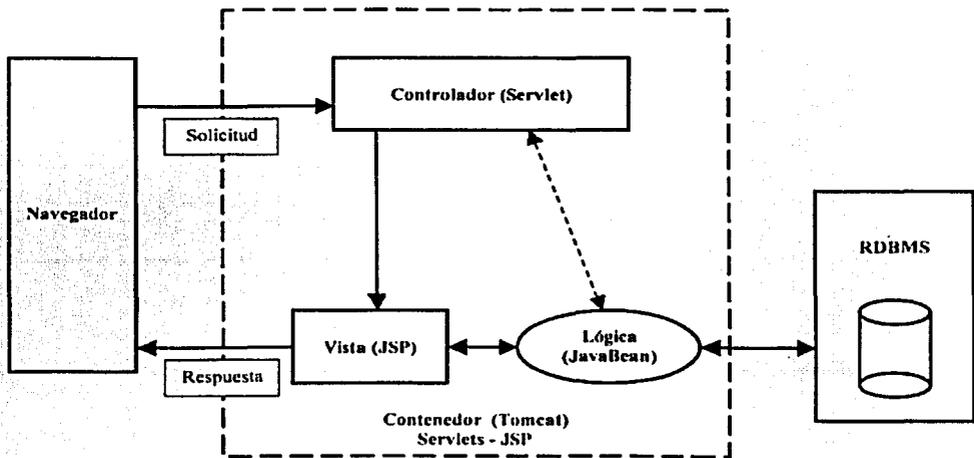


Figura 2.1. Arquitectura del Sistema de Control y Graficación

Nuestra arquitectura consta de un navegador web, el cual le permite al usuario enviar solicitudes de graficación hacia un Servlet, que es el controlador del sistema. El controlador procesa todas las solicitudes HTTP y decide, dependiendo de las acciones del usuario, que componente de presentación (Vista), reenviará hacia el navegador, en ciertos casos solo realiza una sencilla redirección hacia un componente de presentación, que es una página JSP que genera la respuesta HTML que determina la interfaz de usuario, en otras ocasiones realiza procesos de lógica de la aplicación, creando JavaBeans (Modelo), los cuales son utilizados por los componentes de presentación y que interactúan con un recurso externo, como lo es la base de datos relacional.

Los componentes de presentación (Vistas), que en nuestra arquitectura son páginas JSP, usan métodos de acceso para recobrar datos desde un JavaBean, y que se utilizarán para la construcción de la interfaz de usuario.

Tomcat, cumple la función de contenedor de Servlets y de páginas JSP, permitiendo que el sistema se ejecute correctamente y que esté disponible desde cualquier navegador a través del servidor web utilizado.

Las ventajas que se pueden citar de la utilización de esta arquitectura es que el procesamiento de la lógica de la aplicación no se mezcla con el componente de la presentación, con esto se facilita la recuperación de objetos o instancias de cualquier componente JavaBean que se puedan crear previamente por el controlador. El tener una separación claramente definida, de la presentación y de la lógica de la aplicación, nos trae como consecuencia una mayor facilidad de mantenimiento, uniformidad en la presentación y una mayor seguridad del sistema.

3

Diseño del Sistema de Control y Graficación

Resumen

En este capítulo se describen los elementos utilizados para la ingeniería y el diseño del sistema, incluyendo el diseño de la base de datos con el uso de un diagrama entidad-relación, y el diseño del funcionamiento del sistema con el uso de diagramas de casos de uso, diagramas de clases y diagramas de secuencias.

3.1. Diseño del sistema

El diseño del sistema de control y graficación es la etapa que nos muestra la forma en la que el sistema terminado cumplirá con los requerimientos identificados durante la fase de análisis.

El proceso de diseño del sistema de control y graficación resulta ser una de las etapas de más importancia para el desarrollo del sistema, ya que principalmente se enfoca en traducir los requerimientos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la etapa de codificación. Principalmente por este motivo, el diseño del sistema de control y graficación se enfoca sobre cuatro atributos distintos del programa, que son: la estructura de los datos, la arquitectura del software, el detalle de procedimientos y la caracterización de la interfaz.

3.1.1. Módulos y Actores

El desarrollo del sistema de control y graficación estará desarrollado básicamente para el manejo de dos tipos de usuario, el primero es el usuario con permiso de Solicitante, que será la persona que realice las solicitudes de graficación de planos a través de cualquier navegador web. El segundo tipo de usuario es el que presenta un permiso de Administrador del sistema, que será la persona encargada de procesar todas las ordenes de graficación que se encuentren en un estado pendiente, además de esto, el Administrador podrá elaborar reportes de acuerdo a cada una de las características de las solicitudes de graficación procesadas.

De acuerdo con los dos tipos de usuario que podrán hacer uso del sistema, se ha determinado dividir el funcionamiento del mismo en dos grandes módulos, los cuales serán dos controladores Servlets que procesen las peticiones de usuario dependiendo del permiso asignado. De esta forma se tendrá un módulo de Solicitudes, que será implementado por un Servlet, el cual controle y otro módulo de Administración, que será un Servlet distinto, el cual procesará las y procese todas las peticiones de un usuario con permiso de Solicitante peticiones de un usuario con permiso de Administrador. En la Figura 3.1. se observan los dos módulos diseñados para el funcionamiento del sistema.

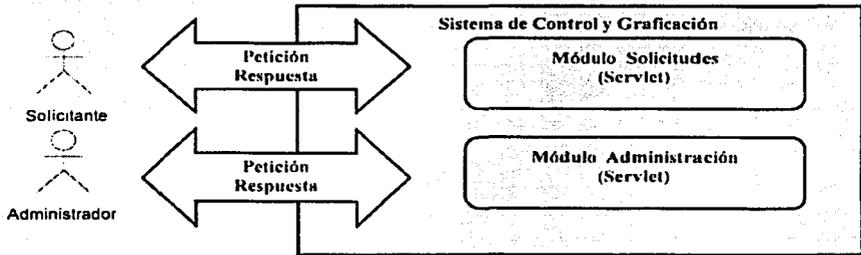


Figura 3.1. Diseño de Módulos y Actores del Sistema

3.2. Diseño de la base de datos

Para realizar el diseño de la base de datos se hace uso de una herramienta gráfica como lo es el diagrama Entidad-Relación (E-R), en el que se describe lógicamente las entidades que forman parte del sistema.

3.2.1. Diagrama entidad-relación

La Figura 3.2. muestra el diagrama E-R para el desarrollo del sistema de control y graficación.

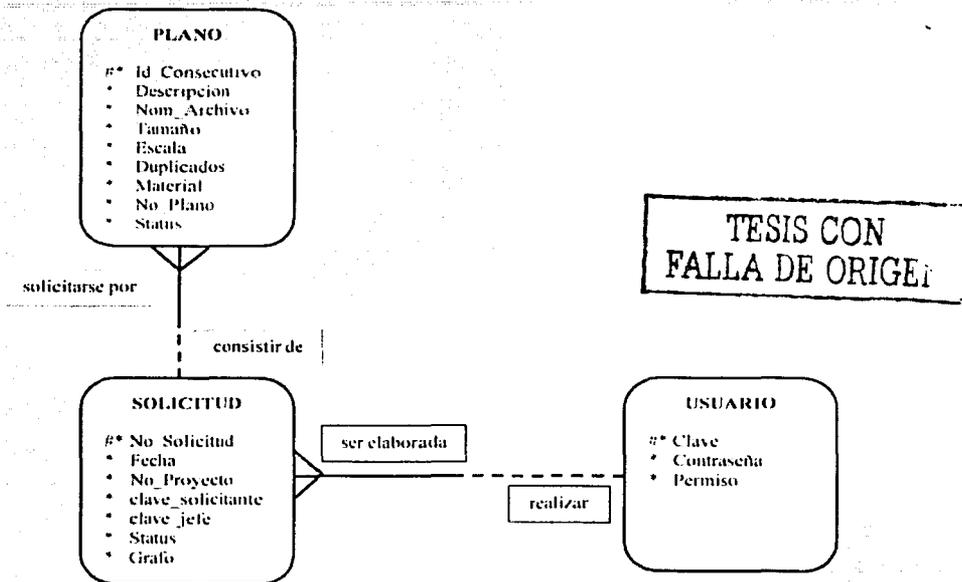


Figura 3.2. Diagrama Entidad-Relación

La Tabla 3.1. describe las tablas que se utilizan en la base de datos del sistema de control y graficación de planos.

Entidad	Descripción
SOLICITUD	Esta tabla almacena la información de la persona que realiza la solicitud de graficación, en ella se guardan el numero de solicitud de graficación, la fecha de la realización, el numero de proyecto a la que pertenece dicha solicitud de graficación, la clave de empleado del solicitante, la clave del jefe del solicitante, el status que nos indica si la solicitud ya ha sido procesada o no, y el grafo a la que pertenece la solicitud de graficación.

USUARIO	Se guardan los datos que distinguen a cada usuario como son su clave, su contraseña y el permiso que tiene asignado dentro del sistema.
PLANO	Aquí se almacenan las características principales que representan a cada uno de los planos que pueden formar una solicitud de graficación.

Tabla 3.1. Descripción de los elementos del diagrama E-R

3.3. Diagramas del sistema

Los diferentes diagramas del Sistema de Control y Graficación nos proporcionan una representación gráfica del funcionamiento del mismo, estableciendo las especificaciones del sistema y definiendo características para lograr que el producto cumpla con la calidad deseada. El objetivo de los diagramas es la descripción del funcionamiento del sistema de graficación el cual está formado por la interacción de módulos independientes entre sí, logrando con esto la facilidad de mantenimiento de dichos módulos.

De acuerdo con la estructura que presenta el Modelo 2 de programación [Paw01], el desarrollo del Sistema de Control y Graficación estará basado en la idea de separar el manejo de la lógica, la interfaz gráfica y la comunicación con el Servidor.

A continuación se hace uso de los distintos tipos de diagramas de acuerdo al Lenguaje Unificado de Modelado (UML) [FoSA99], que nos permiten describir de forma gráfica el funcionamiento y la interacción entre los objetos y módulos que constituyen el Sistema de Control y Graficación.

3.3.1. Diagrama de Casos de Uso

Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo [FoSB99]. El objetivo de este tipo de diagramas es el de ejemplificar de manera detallada todas las acciones que el usuario, dependiendo del permiso asignado, puede realizar con el sistema de control y graficación.

En la Figura 3.3. podemos observar los diferentes casos de uso para cada tipo de usuario.

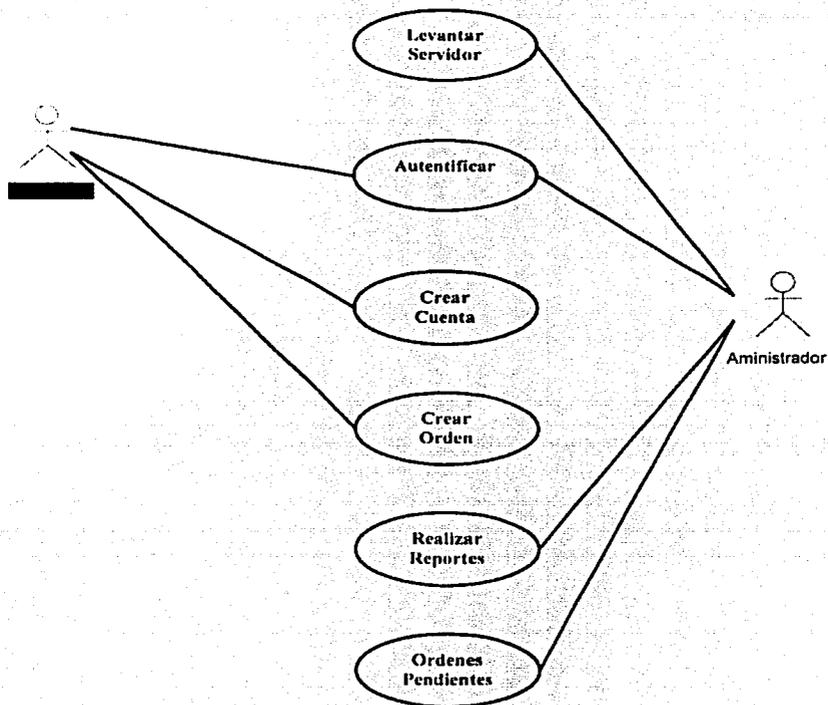


Figura 3.3. Diagrama de Casos de Uso.

A continuación se presenta una breve descripción de los distintos casos de uso que pueden realizar usuarios con permisos de Solicitante y de Administrador.

Caso de uso:

Actor:

Descripción:

Autenticar

Solicitante, Administrador

Para poder realizar una orden de graficación utilizando el sistema de control y graficación a través de un navegador web, es necesario proporcionar una clave de usuario y una contraseña, que serán validadas por el Servidor.

Caso de uso:

Actor:

Descripción:

Crear Cuenta

Solicitante, Administrador

Si el usuario del sistema no tiene una clave o contraseña que sean válidas por el Servidor, el usuario tiene la opción de poder crear una nueva cuenta que sea válida. Para la creación de la cuenta nueva, el usuario tiene que elegir la Competencia a la que pertenece dentro de la organización del IMP y su Nombre, elecciones que se realizan por medio de cuadros de selección presentados por el propio sistema.

Caso de uso:

Actor:

Descripción:

Crear Orden

Solicitante, Administrador

Después de realizar el proceso de autenticación correctamente, el sistema le permite al usuario que hace la solicitud la oportunidad de crear ordenes de graficación. Este proceso consiste en elegir proyectos y grafos, que sean válidos por el sistema de administración de proyectos (SAP) del IMP, por parte de la persona que está realizando la solicitud de graficación, se debe mencionar que solo será posible la creación de ordenes de graficación si existen estas dos características que pueden ser distintas para cada usuario. Posteriormente el usuario proporcionará las características de los planos que desea sean graficados, dichas características son las siguientes: proporcionar el nombre del archivo graficable, tamaño de graficación, escala, número de duplicados, número de plano, tipo de papel, y una descripción del plano a graficar, esta asignación de datos se debe realizar tantas veces como planos desee mandar graficar.

Caso de uso:

Actor:

Descripción:

Ordenes Pendientes

Administrador

Una de las dos opciones que tiene un usuario con permiso de Administrador es la de revisar la lista de órdenes de graficación que están pendientes. Dentro de ésta opción el sistema le presenta al Administrador una lista de todas las órdenes pendientes, de las cuales él podrá elegir, una a la vez, para realizar la graficación de los distintos planos que pueden incluirse en una orden, de acuerdo a las características de los mismos. Cuando el Administrador concluye con el proceso de una orden de graficación, el sistema le da la opción de seleccionar otra orden pendiente, para continuar con el mismo proceso con las demás órdenes restantes.

Caso de uso:

Actor:

Descripción:

Realizar Reportes

Administrador

La otra opción que tiene un usuario con permiso de Administrador es la posibilidad de poder realizar reportes sobre las distintas órdenes de graficación que se hayan procesado. Los reportes podrán realizarse de acuerdo a las siguientes características; por fecha, donde el graficador tendrá que proporcionar el rango de fechas en que se hayan procesado las órdenes solicitadas, por actividad de proyecto (grafo), en esta opción se indica el grafo por el cual se solicita la búsqueda; por proyecto, cuando se soliciten todas las órdenes procesadas que estuvieron relacionadas con un proyecto en específico; por solicitante, para obtener todas las órdenes de graficación solicitadas por un empleado en particular; y por solicitud, si se quisiera conocer todas las características de una orden en particular. Se debe mencionar que el sistema también permite la realización de reportes por la combinación de dos o más de las características antes mencionadas, con el objetivo de restringir la búsqueda y obtener de forma más rápida la orden o las características solicitadas.

Caso de uso:
Actor:
Descripción:

Levantar Servidor
Administrador

Un usuario con permiso de Administrador será la persona encargada de levantar el servidor web donde reside el sistema, con la finalidad de que el sistema sea accesible desde cualquier navegador web dentro y fuera del IMP. Cuando el servidor arranca, se crean los servlets principales del sistema y se ejecutan sus respectivos métodos `init()` con su codificación correspondiente.

3.3.2. Diagrama de Clases

Nuestro diagrama de clase describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones que existen entre ellos. Los diagramas de clase también muestran los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos [FoSC99].

Los módulos en que se ha dividido el funcionamiento del sistema están compuestos de varios tipos de objetos, los cuales permiten el procesamiento de datos y generan las respuestas correctas hacia el usuario, dependiendo del tipo de solicitudes que cada módulo recibe. La Figura 3.4 muestra el diagrama de clases y la relación que existe entre ellas, posteriormente se hace una descripción del funcionamiento general de los objetos que se han diseñado para el funcionamiento del sistema de control y graficación.

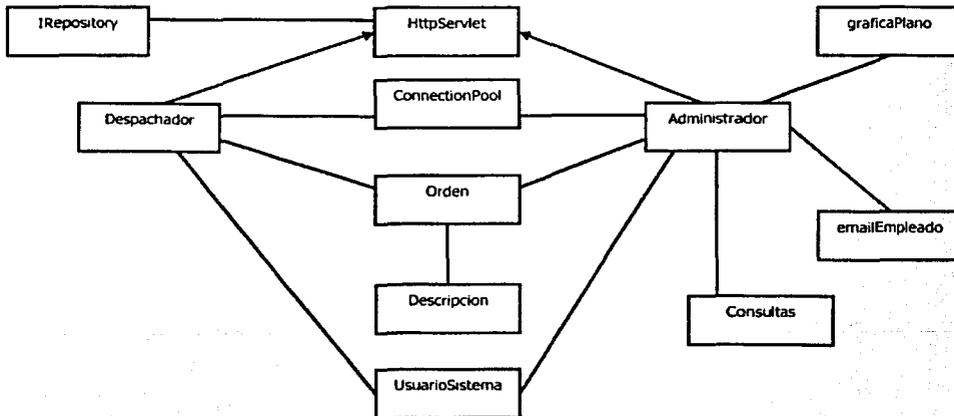


Figura 3.4. Diagrama de clases del sistema

-
- **Objeto ConnectionPool:** Debido a que en distintas etapas del funcionamiento del sistema es necesario la conexión y la manipulación de una base de datos, es de gran importancia la creación de un objeto que facilite dicho trabajo, es por esta razón que se diseña un objeto el cual nos permitirá asignar previamente las conexiones a la base de datos, administrar las conexiones disponibles, asignar nuevas conexiones, aguardar a que una conexión quede disponible o cerrar las conexiones cuando sea necesario, el uso de este objeto nos permite mantener un conjunto de conexiones abiertas a la base de datos para aumentar el desempeño de la aplicación.
 - **Objeto Consultas:** Una de las tareas que tiene un usuario con permiso de Administrador, es la de realizar reportes de acuerdo a las características de los ordenes de graficación procesadas, dichos reportes se traducen posteriormente en constantes consultas hacia una base de datos, para ello es necesario un objeto que construya, de forma dinámica, cada consulta que el Administrador desee realizar.
 - **Objeto Descripcion:** La creación de una orden de graficación, por parte de un usuario con permiso de Solicitante, puede estar constituida por uno o varios planos que se deseen graficar, de los cuales es necesario obtener diversas características como son tamaño, escala, duplicados, numero de plano, material, nombre del archivo, etc., el objetivo de construir un objeto de tipo Descripcion es la de recuperar en una sola entidad todos los detalles de estas características, que son diferentes según el plano a graficar, además, el objeto maneja un numero único (llave primaria), que permite identificar y relacionar a cada plano con sus respectivas características.
 - **Objeto graficaPlano:** Este objeto es el que permitirá a un usuario, con permiso de Administrador, realizar el proceso de graficación de cada uno de los planos que componen a una orden.
 - **Objeto Orden:** El objetivo principal de construir un objeto de este tipo es contar con una entidad lógica que represente una orden de graficación, conteniendo las características de la persona que realiza la solicitud de graficación como son clave del solicitante, proyecto, actividad dentro del proyecto, fecha, numero único (llave primaria), que identifique a cada una de las ordenes de graficación, y los detalles o características de los planos por graficar, que son objetos de tipo Descripcion.

- Objeto UsuarioSistema: Debido a la existencia de dos tipos de usuarios distintos, es necesario la creación de un objeto que realice el proceso de asignación de permisos, dependiendo del tipo de usuario que intente acceder al sistema, dicho objeto efectuará consultas a la base de datos y obtendrá el permiso del usuario por medio de una clave y una contraseña asignadas con anterioridad. De esta forma se determinará el perfil del usuario para construir el tipo de interfaz gráfica el cual tiene acceso.
- Objeto EmailEmpleado: objeto que será utilizado para el envío de un mensaje de aviso a cada usuario del sistema, indicándole que su solicitud de graficación ha sido procesada.

3.3.3. Diagramas de Secuencias

Nuestros diagramas de secuencia son modelos que describen la manera en que colaboran los grupos de objetos diseñados anteriormente [FoSD99]. A continuación se presentan los diagramas de secuencia que describen el funcionamiento del Sistema de Control y Graficación y la relación entre los principales objetos que conforman los casos de uso presentados con anterioridad.

3.3.3.1. Diagrama de Secuencias del Caso de Uso Autenticar

La Figura 3.5. muestra el diagrama de secuencia que corresponde al caso de uso Autenticar, el cual se describe a continuación: Cuando el usuario con permiso de Administrador levanta el Servidor web correspondiente, el servlet Despachador genera una instancia de un objeto connectionPool a través de su método `init()`. Debido a que el sistema realiza conexiones de forma repetida a la misma base de datos, es mejor utilizar un objeto del tipo `connectionPool`, el cual nos permite mantener un arreglo de conexiones a la base de datos y reciclarlas conforme el usuario las solicite, obteniendo con esto una ganancia en el tiempo de respuesta del sistema, ya que las conexiones se mantienen abiertas todo el tiempo.

De esta forma cuando un usuario con permiso de Solicitante desee crear una solicitud de graficación, utilizando el sistema de graficación, será necesario que indique la URL inicial del sistema por medio de un navegador web, cuando esto ocurre el servlet Despachador realiza un redireccionamiento hacia una página JSP que le permitirá realizar el proceso de autenticación, dicho proceso consiste en proporcionar una clave, una contraseña y pulsar el botón de Aceptar, permitiendo así que el servlet Despachador genere una instancia de objeto UsuarioSistema, el cual ejecuta el método `buscarUsuario()`, que realiza una consulta a la base de datos y

determina si el usuario tiene una cuenta válida para poder acceder al sistema, de ser así, el servlet Despachador genera una instancia de un objeto HttpSession, el cual nos permite buscar información asociada con la sesión en curso, almacenar información en una sesión o descartar las sesiones finalizadas o abandonadas, en nuestro caso este objeto le permite al servlet Despachador guardar la clave del usuario mediante el método setAttribute(). Posteriormente el servlet Despachador ejecuta el método irAPag(), el cual realiza el redireccionamiento hacia una página JSP que le permitirá al usuario comenzar con la creación de una solicitud de graficación.

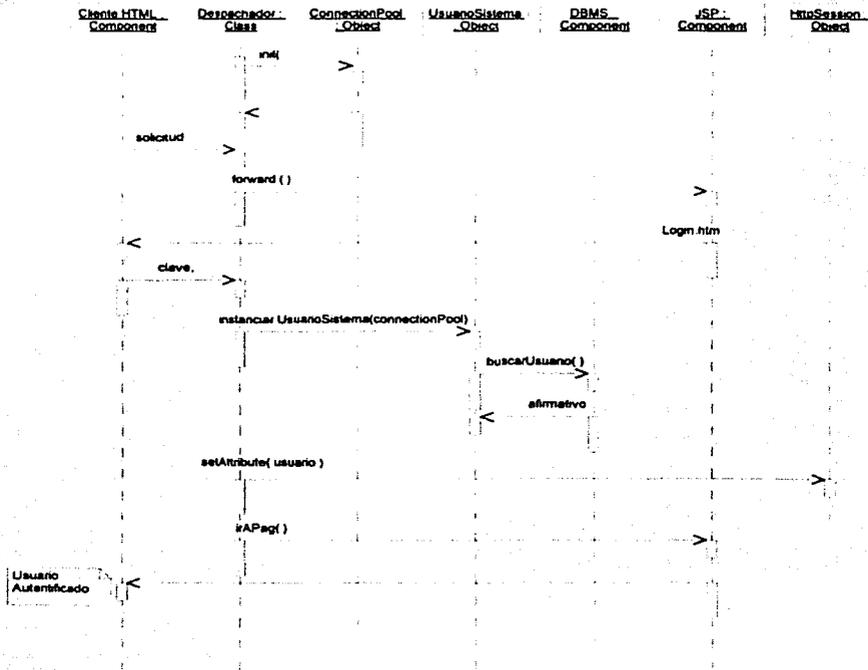


Figura 3.5. Diagrama de Secuencias del Caso de Uso Autenticar

3.3.3.2. Diagrama de Secuencias del Caso de Uso Registrar Usuario

En el caso en que la clave de usuario o la contraseña no sean válidas para el servidor, el servlet Despachador realiza un redireccionamiento hacia otra página JSP, la cual le permitirá registrar un usuario con una nueva cuenta que sea válida para utilizar el Sistema de Graficación.

Antes de presentar al usuario la página JSP que permite crear una cuenta nueva, ésta realiza algunas operaciones que se describen a continuación: genera Instancias de objetos del tipo IRepository, SapR3 y Competencias, dichos objetos pertenecen al API de conexión con SAP denominado Sap

Java Connector. Para mayor información sobre este API consultar "SAP Java Connector Tutorial" [JCO].

La instanciación de estos objetos le permite al sistema presentar una página JSP con un cuadro de selección donde se encontrarán todas las Competencias en que se encuentra dividido la organización interna del IMP, cuando el usuario seleccione la Competencia a la que pertenece se genera un nuevo cuadro de selección con los nombres de todos los empleados que pertenecen a dicha Competencia. Una vez que el usuario haya seleccionado la Competencia y su nombre, proporcionará su contraseña a través de los cuadros de edición de texto y seguidamente pulsará un botón de Aceptar, el cual permite una interacción con el servlet Despachador que a su vez genera una instancia de un objeto UsuarioInserta, con dicho objeto es posible ejecutar el método InsertarUsuario(), el cual realiza una operación de inserción en la base de datos con los datos proporcionados por el usuario.

Después del proceso de inserción en la base de datos, el servlet Despachador guarda la clave de usuario, dentro de un objeto del tipo HttpSession ejecutando el método setAttribute(), posteriormente el servlet Despachador ejecuta el método irAPag() el cual realiza una redirección hacia una nueva página JSP que permitirá la creación de una orden de graficación.

Para poder presentar al usuario una página JSP que permita la creación de una solicitud de graficación, la página crea una instancia de un objeto tipo IRepository, con el se construye un objeto tipo SapR3, el cual a su vez permite la creación de un objeto tipo Personal, un objeto WorkList y un objeto Proy. El objeto Personal contiene los métodos get_Nombre(), get_Posicion(), get_Organización() y get_BusArea(), con los cuales obtenemos el Nombre, el Puesto, la Competencia y la Dirección Ejecutiva del usuario que crea la orden de graficación. Con el objeto WorkList y su método get_Network().keys(), obtenemos una lista de las actividades que el usuario tiene asignadas para sus proyectos dentro del sistema SAP [SAP], denominadas Grafos. Con el objeto Proy se ejecuta el método get_Proys().keys(), que nos entrega una lista de los proyectos en los cuales está tripulado el usuario.

Una vez que se han obtenido los datos del usuario se puede presentar una página de solicitud de graficación al solicitante, la cual contendrá su datos y le presentará dos cuadros de selección, uno con la lista de grafos y el otro con la lista de proyectos que tenga asignados, con los cuales el usuario seleccionará el proyecto y la actividad a la cual se le harán cargos en la creación de la orden de graficación.

3.3.3.3. Diagrama de Secuencias del Caso de Uso Crear una Orden de Graficación

Para poder crear una orden de graficación el usuario debe seleccionar un Grafo y un Proyecto de la página que se le presenta para la creación de la orden, posteriormente debe pulsar sobre un botón de Aceptar, cuando esto ocurre la página JSP manda los parámetros seleccionados por el Usuario hacia el servlet Despachador, el cual recupera la clave del usuario solicitante del objeto HttpSession a través del método `getAttribute()`, y crea un objeto tipo Orden que inicialmente contiene la clave de usuario, el proyecto, el grafo y un objeto del tipo `connectionPool`. Posteriormente el servlet guarda el objeto Orden en un objeto HttpSession y ejecuta el método `irAPag()`, el cual realiza un redireccionamiento hacia una página JSP que le permitirá al usuario establecer las características de los planos que desee mandar a graficar, las características constan de nombre de plano, tamaño, escala, duplicados, número de plano, material y una pequeña descripción del plano a utilizar, estas características se establecen por parte del usuario a través de cajas de selección y cajas de edición de texto. Cuando el usuario ha establecido de forma correcta todas las características del plano a graficar, debe pulsar sobre un botón "Solicitar Orden", el cual manda las características establecidas por el usuario hacia el servlet Despachador, el cual los recupera y crea un objeto del tipo Descripción, recupera el objeto Orden del objeto HttpSession utilizando el método `getAttribute()` y ejecuta el método `addDescripción()` del objeto Orden, con el cual agrega el objeto Descripción al objeto Orden y posteriormente guarda el objeto Orden en el objeto HttpSession. Se debe mencionar que este proceso se puede realizar de manera repetitiva, dependiendo de la cantidad de planos que el usuario quiera graficar, al final el objeto Orden contendrá los datos del usuario solicitante y tantos objetos del tipo Descripción como planos se deseen graficar.

Posteriormente el servlet Despachador ejecuta el método `irAPag()` que realiza la redirección hacia una página JSP en donde se le muestra al usuario el estado completo de su solicitud de graficación, en cual, si el usuario está de acuerdo, pulsará sobre un botón de "Terminar Orden", con lo cual el servlet Despachador recupera el objeto Orden del objeto HttpSession y ejecuta el método `mandarOrden()`, con lo que se inserta el objeto Orden en la base de datos. Una vez insertado el objeto Orden, el servlet ejecuta nuevamente el método `irAPag()`, con lo que el sistema le muestra al usuario una página JSP que le indica el número de orden que le fue asignado a su solicitud de graficación y que recibirá un correo electrónico cuando su orden haya sido procesada por parte del Administrador del Sistema.

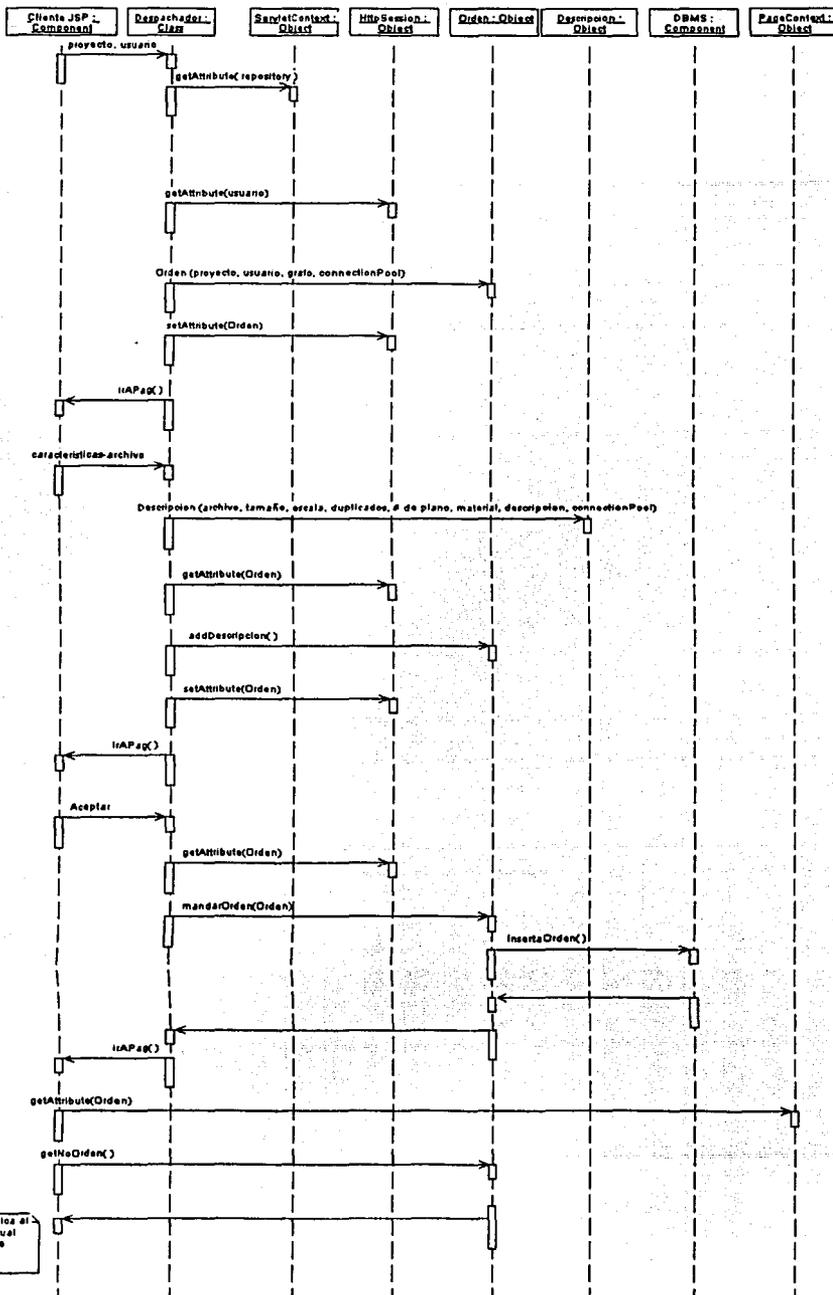


Figura 3.7. Diagrama de Secuencias del Caso de Uso Crear una Orden de Graficación

3.3.3.4. Diagrama de Secuencias del Caso de Uso Generar Reportes de Graficación

Cuando un usuario realiza el proceso de autenticación, el sistema determinará el permiso del usuario. Para el caso del permiso de Administrador, el sistema presenta una página JSP distinta a la que presenta a un usuario con permiso de Solicitante, la página de un usuario con permiso de Administrador presenta dos opciones, la primera opción permite crear reportes y es la que se ilustra con el diagrama de secuencias de la Figura 3.8, la segunda opción se explicará en la siguiente sección de este capítulo.

Para seleccionar la opción de crear reportes se tiene que pulsar el botón correspondiente, con lo cual el servlet Administrador ejecuta el método `irAPag()`, cuya finalidad es presentarle al Administrador una página JSP que contiene los criterios de selección de búsqueda que puede elegir, los criterios son presentados por medio de cuadros de selección y pueden ser: por rango de fechas, por grafo, por proyecto, por solicitante o por número de solicitud y el Administrador puede seleccionar uno o más criterios según lo requiera.

Después de que el Administrador elige los criterios de búsqueda oprime un botón de Continuar, que permite al servlet Administrador obtener el nombre de los criterios por los cuales el Graficador desea realizar la búsqueda y con ellos crea un objeto de tipo `Consultas`, posteriormente ejecuta el método `setAttribute` del objeto `HttpSession` para guardar el objeto `Consultas`. El servlet Administrador invoca su método `irAPag()` y redirecciona hacia una página JSP que tiene como objetivo el obtener y dirigir hacia el servlet Administrador los valores de los criterios de búsqueda que el Administrador proporcionó. Cuando el servlet Administrador obtiene los valores enviados por la página JSP, recupera el objeto `Consultas` del objeto `HttpSession` y realiza métodos del tipo `SetXxx()` para establecer dichos valores en propiedades del mismo objeto. El servlet ejecuta el método `builtQuery()` del objeto `Consultas`, que forma una consulta SQL de forma dinámica y que es válida para la base de datos, ejecuta después el método `ejecutaQuery()`, que hace una conexión con la base de datos y realiza la consulta. Los resultados obtenidos por la acción del método `ejecutaQuery` quedan guardados en una propiedad de tipo `cursor` del objeto `Consultas`.

Después de que el servlet Administrador ejecutó el método `ejecutaQuery`, realiza otra redirección con el método `irAPag()`, que le presenta al Administrador una página JSP que recupera al objeto `Consultas` del objeto `HttpSession`, y que ejecuta varios métodos del tipo `getXxx()` con el fin de presentar los resultados de la consulta en una página de reporte.

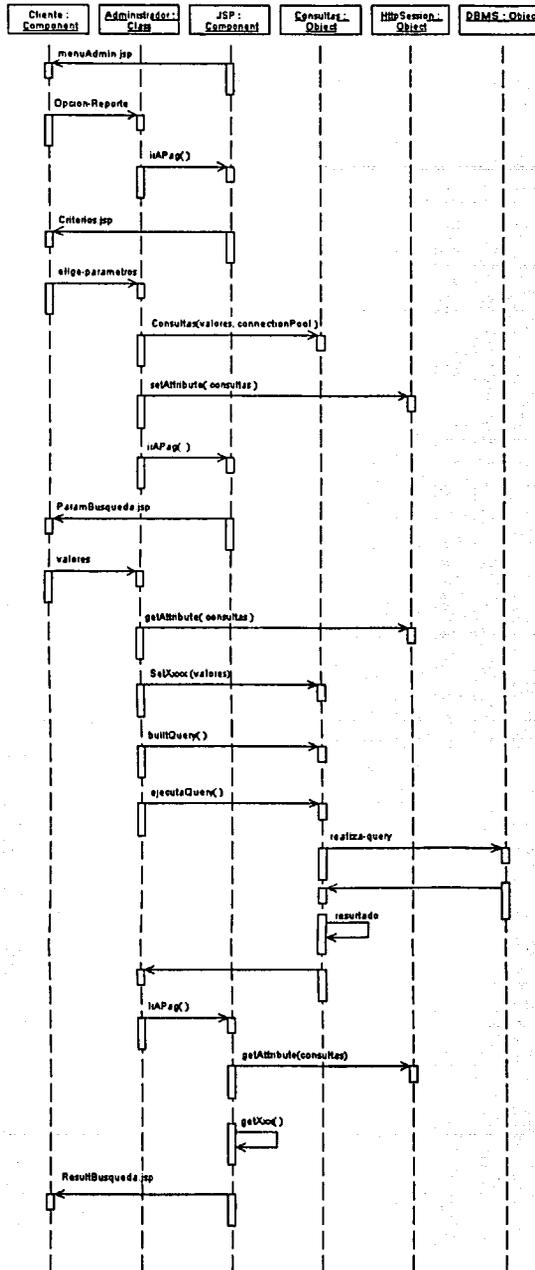


Figura 3.8. Diagrama de Secuencias del Caso de Uso Generar Reportes de Graficación

3.3.3.5. Diagrama de Secuencias del Caso de Uso Graficar Planos de Ordenes Pendientes

La segunda opción a elegir por un usuario con un permiso de Administrador es la de revisar y procesar las ordenes de graficación pendientes, y esto lo hace pulsando el botón correspondiente, con lo cual el servlet Administrador crea una instancia de un objeto Orden para guardarlo dentro de un objeto HttpSession utilizando el método `setAttribute()`, posteriormente el servlet Administrador realiza un redireccionamiento, por medio de su método `irAPag()`, hacia una página JSP que recupera el objeto Orden utilizando el método `getAttribute()` del objeto HttpSession, con este objeto la página JSP ejecuta el método `obtenerSolicitudes()`, que realiza una consulta a la base de datos y obtiene todas las ordenes de graficación que se encuentran en estado pendiente o sin procesar. Cuando la página recibe la entrega del método `obtenerSolicitudes()`, realiza varios métodos del tipo `getXxx()`, con el propósito de construir una página al Administrador que le muestre todas las ordenes de graficación pendientes.

De esta forma el Administrador tiene en una página un listado de todas las ordenes de graficación que deberá procesar. Cuando el Administrador seleccione una orden de graficación, el servlet Administrador guardará el número que identificará a dicha orden en un objeto HttpSession y ejecutará el método `irAPag()`, que llevará al Graficador hacia otra página JSP, la cual antes de presentarse en el navegador, recuperará el número de orden del objeto HttpSession, creará una instancia de un objeto Descripcion y ejecutará el método `obtenerDescripcion()` de dicho objeto, indicándole la descripción que solicita de acuerdo al número de orden recuperado. El método `obtenerDescripcion()` realiza una consulta a la base de datos y encuentra todos los objetos del tipo Descripcion que corresponden al número de orden. Posteriormente la página realiza varios métodos del tipo `getXxx()`, con el fin de recuperar las características que se encuentran en cada objeto Descripcion obtenido. Además de generar una instancia de un objeto Descripcion, la página crea un objeto del tipo `graficaPlano`, con el cual ejecuta el método `consultaStatus()`, que también realiza una consulta a la base de datos con el fin de determinar si una descripcion dentro del objeto Orden ya ha sido procesada o se encuentra pendiente. Este proceso de consultas a la base de datos se repite según el número de objetos Descripcion que existan en un objeto Orden.

Después de realizar las consultas anteriores, la página JSP presenta la orden de graficación que no ha sido procesada con todos los detalles de todos los objetos Descripción que pudiera tener. Dentro de ésta página se tienen dos opciones para el Administrador, la primera, graficar uno por uno los planos que se encuentran dentro del objeto Orden seleccionado, y que están representados por objetos del tipo Descripción, y la segunda, que es terminar la orden de graficación, la cual cambiará el estado de una orden de graficación pendiente a una orden procesada.

Para el desarrollo de la primera opción el servlet Administrador crea una instancia de un objeto graficaPlano y ejecuta el método buscaCaracteristicas(), con el fin de obtener las características del plano a graficar, como escala y número de duplicados, crea una instancia de un objeto ParallelPort, el cual prepara la salida física por la cual transmitirá los datos hacia el dispositivo "Graficador", y posteriormente ejecuta el método graficarArchivo(), con el cual se lee el archivo a graficar y se transmite su contenido hacia el puerto paralelo para su procesamiento.

Después de que el plano ha sido graficado, el servlet Administrador utiliza el método irAPag(), con el cual redirecciona al Administrador hacia una página JSP que pregunta si la graficación del plano ha sido correcta, de ser así, le permite la opción de seguir procesando otros planos de la misma orden si los hubiera, en caso contrario, tendrá la opción de volver a repetir la operación de graficación o terminar de procesar la orden si el Administrador así lo decide por haber errores en el archivo graficado.

Cuando el Administrador decide terminar de procesar la orden el servlet Administrador ocupa la instancia generada del objeto Orden y ejecuta el método actualizaStatusOrden(), el cual realiza una actualización sobre la base de datos para cambiar el estado de una orden de graficación pendiente por una orden de graficación procesada. Después de esto, el servlet Administrador envía una notificación por correo electrónico al usuario que solicitó la orden de graficación, indicándole que su solicitud ha sido procesada.

3.4. Diseño de Interfaces Gráficas de Usuario

Dentro de la arquitectura de desarrollo de nuestro sistema, las interfaces gráficas de usuario o capas de presentación, se traducen en páginas web dinámicas programadas con JavaServer Pages (JSP) [HalB01], las cuales, de acuerdo a Modelo 2 de programación, invocan servicios de Java Beans [HalC01], con los que se realizan la lógica de la aplicación y todos los procesos definidos en el modelado de clases.

Todas las páginas JSP deberán estar instaladas dentro de un contexto válido para el contenedor de Servlets Tomcat[JtoA01]. Las gráficas de usuario se han diseñado de acuerdo al role que una persona puede asumir al utilizar el Sistema de Control y Graficación, estableciéndose diversos tipos de botones de control dentro de las páginas, dependiendo el tipo de usuario que utilice el sistema. El nombre de cada una de las páginas se ha asignado de acuerdo al proceso que permiten realizar dentro del sistema.

Todo los elementos del contexto deben estar incluidos en un directorio raíz, el cual debe presentar una estructura de directorios específica para el funcionamiento del Sistema, en nuestro caso el directorio raíz recibe el nombre de "Alberto" y el nombre del contexto es "jsp". Debe existir un subdirectorio con el nombre *web-inf*, el cual contiene un subdirectorio con el nombre de *classes*, que es donde se encuentran todos los archivos que corresponden a los Servlets o a los JavaBeans utilizados por el sistema.

Las páginas JSP, que representan a todas las interfaces gráficas de usuario de nuestro sistema, se encuentran instaladas en un subdirectorio de nombre *jsp*, que también se encuentra dentro del directorio raíz. La declaración del contexto de nuestro sistema, se realiza editando el archivo *server.xml*, el cual se encuentra en el directorio *conf* de Tomcat. El proceso de declaración del contexto se ilustra en el capítulo 5 de este trabajo de tesis.

4

Construcción del Sistema de Control y Graficación

Resumen

En este capítulo se describirán los detalles de la construcción y codificación de la base de datos del sistema, la creación de las interfaces gráficas de usuario mediante formularios HTML, la validación de datos en formularios HTML utilizando JavaScript del lado del cliente y la codificación de las clases de los objetos involucrados en el sistema como Servlets y componentes JavaBeans.

4.1. Creación de la base de datos

La creación de la base de datos se realizó mediante la ejecución del script que contiene las instrucciones SQL con el esquema de la base de datos relacional a utilizar. La Figura 4.1. muestra el archivo utilizado, en el cual se observan las sentencias utilizadas para la creación de las tablas necesitadas según la etapa de diseño.

```
drop table planos cascade constraints;
create table planos
(
  ID_Consecutivo NUMBER PRIMARY KEY,
  Descripción varchar2(60) NOT NULL,
  Nom_Archivo varchar2(30) NOT NULL,
  Tamano char(1) NOT NULL,
  Escala NUMBER NOT NULL,
  Duplicados number NOT NULL,
  Graficados number NOT NULL,
  Status varchar(20) NOT NULL,
  Material char(1) NOT NULL CHECK (Material IN('P','B')),
  No_Plano varchar2(20) NULL,
);
drop table solicitudes cascade constraints;
create table solicitudes
(
  No_Solicitud NUMBER PRIMARY KEY,
  Fecha date NOT NULL,
  No_Proyecto varchar2(8) NOT NULL,
  ID_Consecutivo NUMBER REFERENCES planos(ID_Consecutivo) NOT NULL,
  clave_solicitante NUMBER NOT NULL,
  clave_jefe NUMBER NOT NULL,
  Grafo number NOT NULL,
  FOREIGN KEY (ID_Consecutivo) REFERENCES planos
);
create table Contraseñas
(
  Clave varchar(20) NOT NULL PRIMARY KEY,
  Password varchar(10) NOT NULL,
  Admin varchar(1) NOT NULL,
);
drop sequence No_solicitud;
CREATE SEQUENCE No_solicitud START WITH 1 INCREMENT BY 1;
drop sequence ID_Consecutivo;
CREATE SEQUENCE ID_Consecutivo START WITH 1 INCREMENT BY 1;
```

Figura 4.1. Archivo del esquema de la base de datos

Se usa la tabla PLANOS para almacenar todas las características que representan los planos, se puede observar que todos los datos son del tipo no-nulo, a excepción del campo No_Plano, el cual puede o no contener datos. La tabla planos lleva como llave primaria el valor que se encuentra en el campo ID_Consecutivo.

Se usa la tabla SOLICITUDES para guardar todos los datos correspondientes a una orden de graficación, en ella podemos ver que ningún dato puede ser nulo, y que lleva por llave primaria al valor del campo No_Solicitud, con el cual se identifica a cada una de las ordenes de graficación. Además, la tabla cuenta con una llave foránea que es a su vez la llave primaria de la tabla PLANOS.

Se usa la tabla CONTRASENIAS para guardar todos los datos correspondientes a la clave de empleado, la contraseña y el tipo de cuenta asignado a cada usuario, todos los datos son del tipo no-nulo y se asigna el campo Clave como llave primaria de la tabla.

Se crean adicionalmente dos secuencias, para generar las claves primarias de las tablas PLANOS y SOLICITUDES.

4.2. Programación de las interfaces gráficas de la aplicación

Las interfaces gráficas de la aplicación fueron programadas a través de formularios HTML y tecnología JavaServer Pages (JSP) [HalB01], la construcción de éstas se realizó mediante el uso de un editor de texto simple y están clasificadas a continuación en función de los casos de uso.

4.2.1. Interfaces Gráficas del Caso de Uso Autenticar

La Figura 4.2 es la interfaz gráfica que se le presenta al usuario cuando accesa el sistema, para ello, el usuario deberá acceder desde su navegador web la dirección electrónica de la página de inicio del sistema, <http://proy.imp.mx/jsp/jsp/Index.jsp>, y posteriormente selecciona el botón correspondiente, ya sea para comenzar con la elaboración de una orden de graficación, a través del proceso de autenticación o para la creación de una nueva cuenta de usuario. En la Figura 4.2 se indica la opción del botón *Login*.

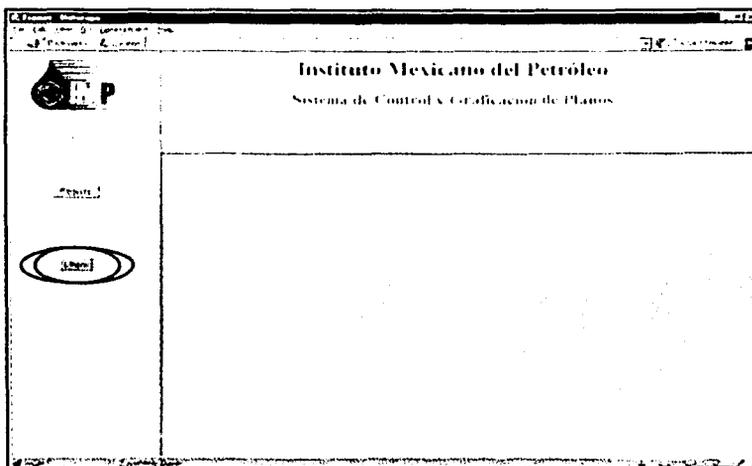


Figura 4.2. Página de inicio del Sistema de Control y Graficación

La Figura 4.3. muestra la interfaz gráfica del sistema después de haber seleccionado la opción *Login*. El objetivo de esta interfaz gráfica, es capturar los parámetros del proceso de autenticación, solicitando la cuenta y contraseña de usuario.

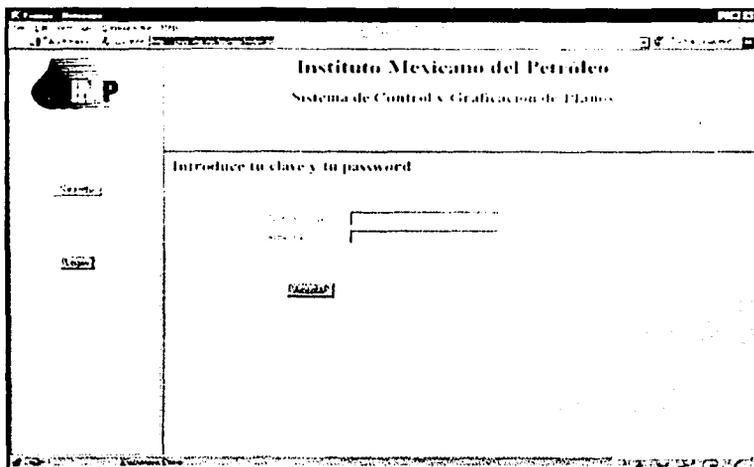


Figura 4.3. Página de autenticación de usuario

4.2.2. Interfaces Gráficas del Caso de Uso Crear Cuenta

Si el usuario no tiene una cuenta válida para acceder el sistema, deberá elegir la opción del botón *Registro* de la interfaz gráfica inicial Figura 4.2, de esta forma, el usuario podrá crear una cuenta nueva para acceso al sistema.

El proceso de creación de una cuenta nueva consiste en seleccionar la Competencia a la que el empleado que solicita el registro, pertenece, la Figura 4.4. muestra la interfaz de la selección de la competencia. Posteriormente, el usuario tendrá que seleccionar su nombre, véase Figura 4.5, con esto, el sistema obtiene la clave de empleado, que es un número único con el cual cada persona se distingue dentro del IMP, y finalmente, el usuario proporciona la contraseña y la verifica. En la Figura 4.6 se muestra la interfaz gráfica de este proceso.

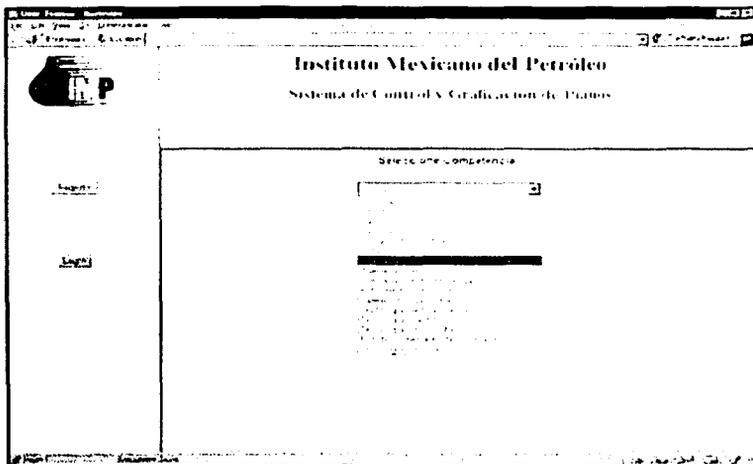


Figura 4.4. Selección de Competencia

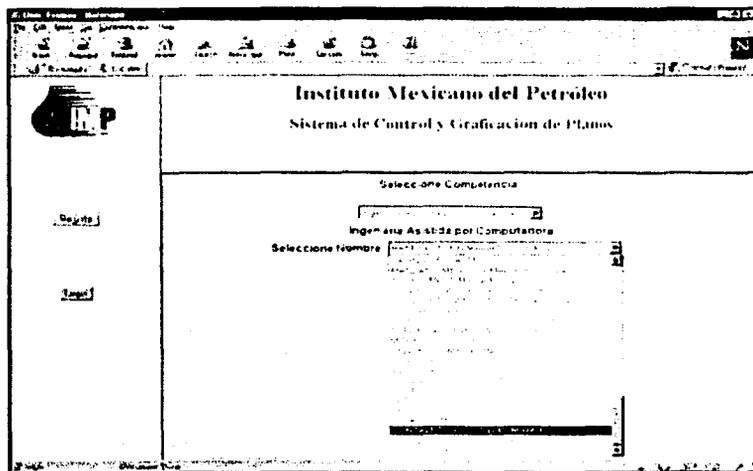


Figura 4.5. Selección de nombre de empleado

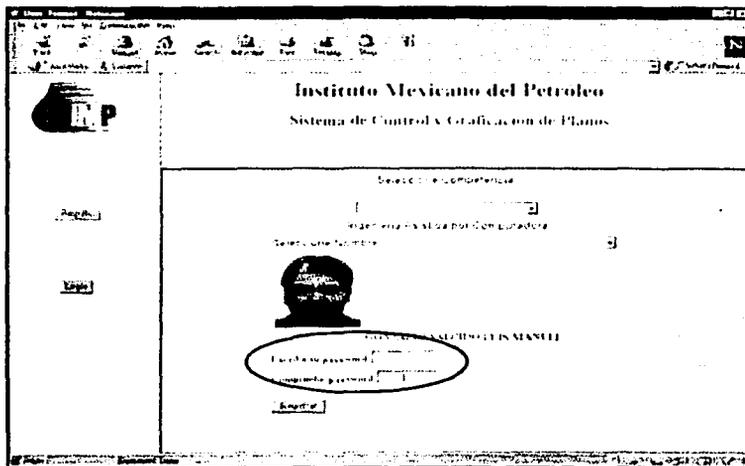


Figura 4.6. Escritura de contraseña

4.2.3. Interfaces Gráficas del Caso de Uso Crear Orden

Cuando el usuario accesa al sistema con una cuenta válida, podrá crear una o varias ordenes de graficación.

La figura 4.7. muestra la interfaz gráfica que el sistema presenta después de que el usuario ha pasado con éxito el proceso de autenticación, en ella, se muestra un reporte de los datos generales del usuario autenticado, en esta misma interfaz el usuario tendrá que indicar el *Proyecto* y la actividad (*Grafo*) a la cual estará relacionada la orden de graficación. Pulsando el botón *Aceptar* el usuario podrá continuar con la creación de una orden.

En la interface gráfica aparece también el botón *Logout*, el cual permite al usuario salir de la aplicación en el momento en que así lo requiera.

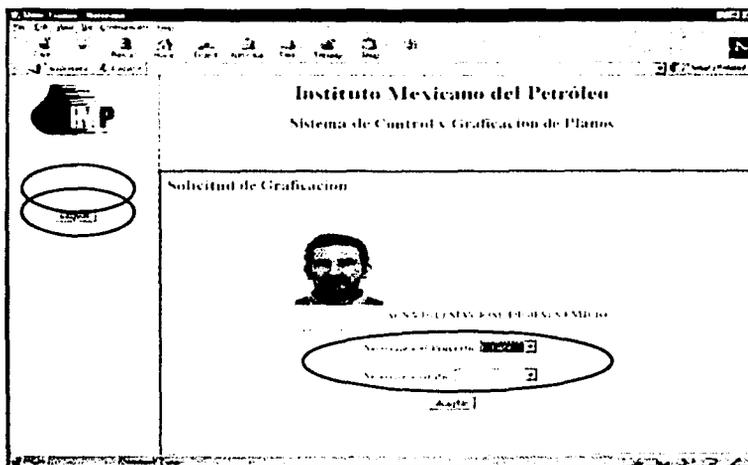


Figura 4.7. Comienzo de elaboración de una Orden de Graficación

La figura 4.8. muestra la interfaz gráfica en la cual el usuario deberá establecer todas las características de los planos a graficar para un orden de graficación. Si el usuario no adiciona las características de los planos la orden de graficación no se creará.

Una vez indicadas las características de los planos a graficar el usuario deberá activar el botón *Solicitar Orden*, con lo cual se despliega la interfaz gráfica de la Figura 4.9, donde se muestra el estado de la orden de graficación, en ella se indica la cantidad de planos que un usuario desea graficar y sus respectivas características. Además, la interfaz gráfica, permite al usuario seguir anexando varios planos, eliminar planos solicitados dentro de la lista de los ya existentes o terminar con la solicitud de graficación.

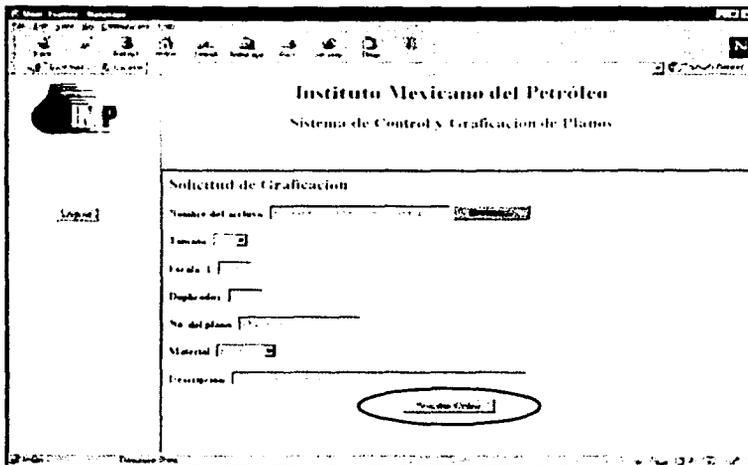


Figura 4.8. Características de un plano solicitado

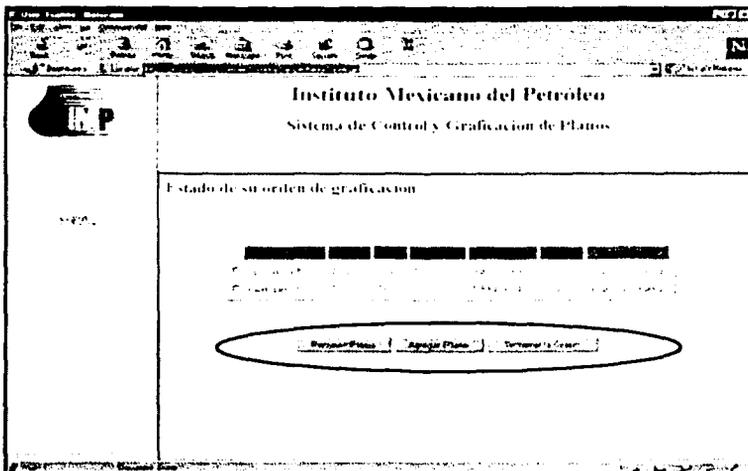


Figura 4.9. Estado de la orden de graficación

Cuando el usuario termina de anexar planos a la orden lo indicará a través del uso del botón *Terminar Orden*, el sistema entonces muestra la interfaz gráfica de la Figura 4.10, en la cual se indica el número de orden asignado por el sistema a la solicitud del usuario, con este número podrá identificar su orden para posterior referencia.

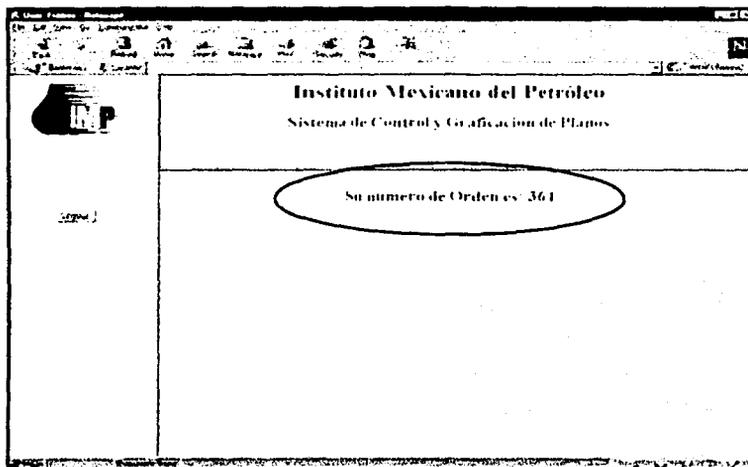


Figura 4.10. Indicación del numero de orden

4.2.4. Interfaces Gráficas del Caso de Uso Realizar Reportes

Cuando el sistema es accesado por un usuario con permiso de Administrador, la interfaz gráfica inicial es la que se muestra en la Figura 4.11, cuyo principal objetivo es presentarle al Administrador las opciones que puede realizar, esto lo hace a través de los botones *Generar Reporte* y *Ordenes Pendientes*, las cuales permitirán al Administrador generar distintos tipos de reportes o procesar todas las ordenes de graficación pendientes.

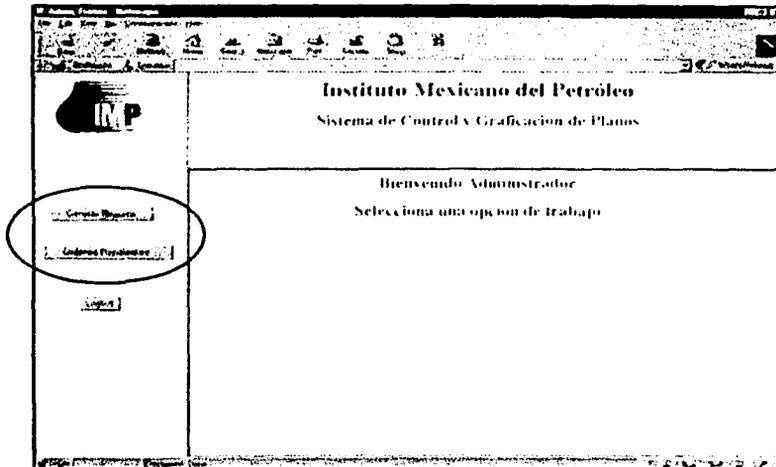


Figura 4.11. Opciones de trabajo del Administrador

Cuando un Administrador pulsa sobre el botón *Generar Reporte* el sistema le presenta la interfaz gráfica de la Figura 4.12, donde se puede indicar diferentes tipos de búsqueda para las ordenes ya procesadas.

La interfaz presenta varios criterios por los cuales el Administrador podrá realizar las búsquedas, además de un botón *Continuar*, con el cual el sistema permite capturar los parámetros correspondientes al criterio de búsqueda elegido por el Administrador. La Figura 4.13 muestra un ejemplo de parámetros para un tipo de búsqueda por clave de empleado y número de solicitud.

Para obtener el reporte de la búsqueda seleccionada, el Administrador deberá pulsar sobre el botón *Continuar*, con el cual el sistema mostrará un reporte similar al de la Figura 4.14. Adicionalmente presenta un botón *Nueva Búsqueda*, con el cual se tendrá la opción de seguir realizando más búsquedas en la base de datos del sistema.

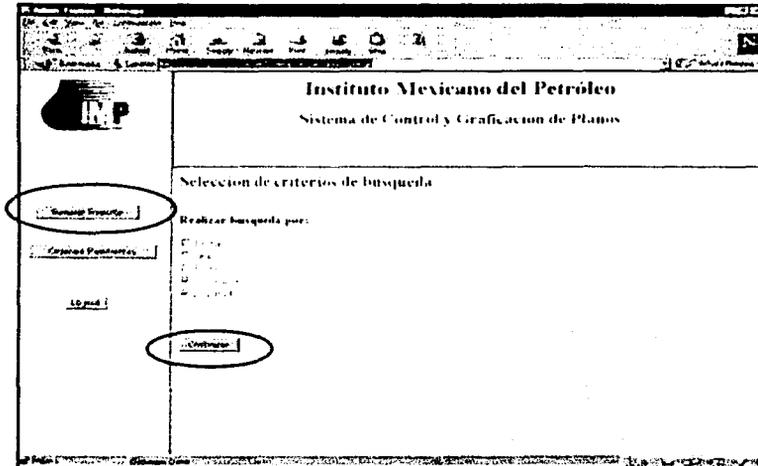


Figura 4.12. Presentación de los criterios de búsqueda

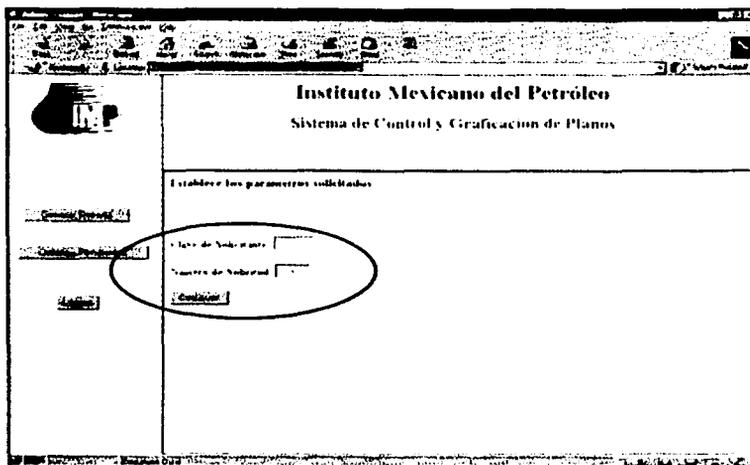


Figura 4.13. Asignación de valor a los criterios seleccionados

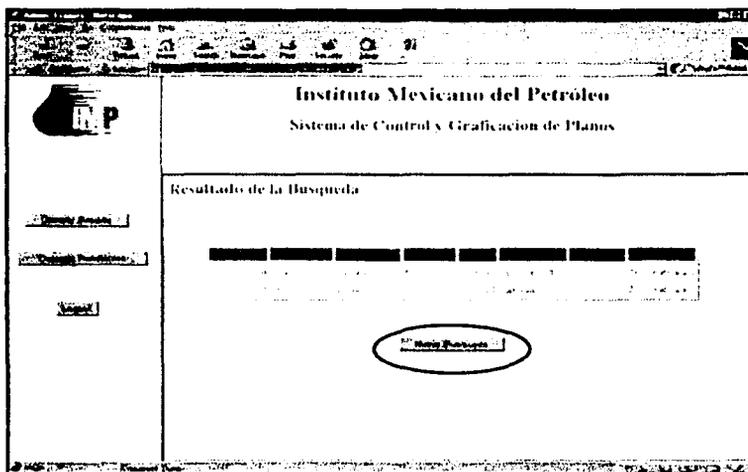


Figura 4.14. Resultados de la búsqueda procesada

4.2.5. Interfaces Gráficas del Caso de Uso Ordenes Pendientes

El Administrador deberá procesar las ordenes de graficación con estatus de pendientes para pasarlas a un estatus de procesadas, y así obtener la graficación de los planos en papel. Pulsando el botón *Ordenes Pendientes*, de la Figura 4.11, al Administrador se le mostrará la interfaz de la Figura 4.15, la cual contiene una tabla con todas las ordenes pendientes y sus respectivas características.

Una de las columnas que presenta la tabla de la interfaz gráfica de la Figura 4.15, nos indica el numero de orden que el sistema asigna a cada solicitud de graficación, éste numero está representado mediante una liga que hace referencia a los detalles de ese número de orden, por ejemplo si se pulsa sobre la liga de la orden número 364, se pueden observar los detalles de esta orden como lo muestra la Figura 4.16.

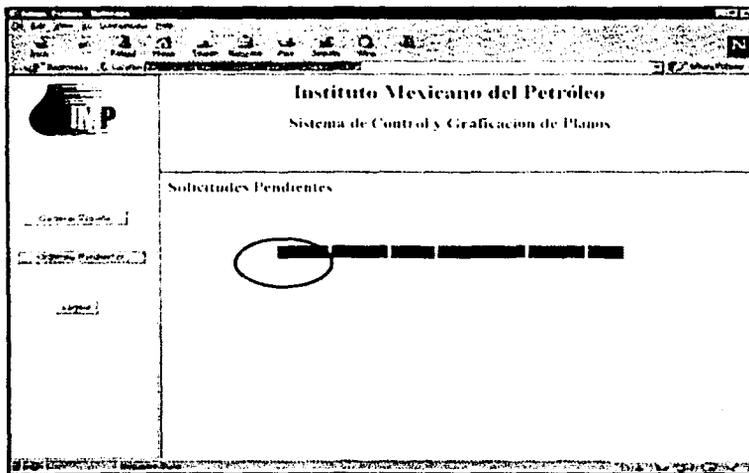


Figura 4.15. Presentación de ordenes de graficación pendientes

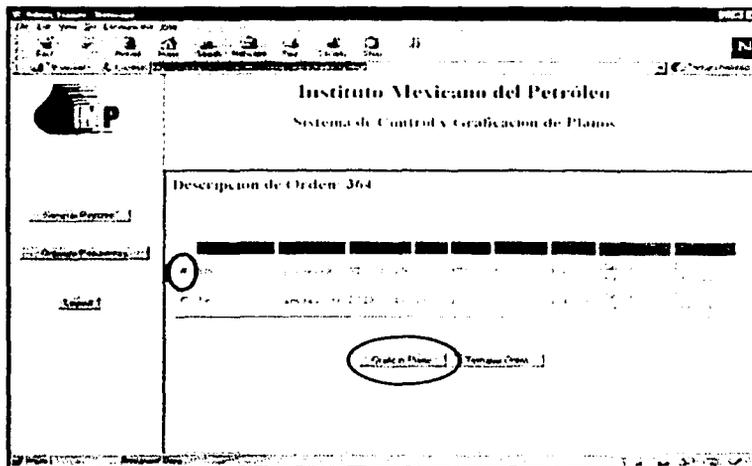


Figura 4.16. Descripción de orden de graficación seleccionada

Para que el Administrador pueda graficar los planos que componen una orden de graficación, deberá seleccionar uno por uno los planos que desea graficar y pulsar el botón *Graficar Plano*.

Cuando el Administrador pulsa sobre el botón *Graficar Plano*, el sistema envía el archivo que corresponde al plano seleccionado al puerto donde se encuentra conectado el graficador, y presenta la interfaz gráfica de la Figura 4.17, que solicita respuesta del Administrador para saber si la graficación del plano fue correcta, en caso afirmativo, el sistema presenta la interfaz gráfica de la Figura 4.18, donde se puede ver que la columna Estado ha cambiado de valor, indicando que el plano ha sido graficado. En caso negativo el sistema presenta la interface gráfica de la Figura 4.16, para que el Graficador tenga la opción de poder intentar nuevamente la graficación del plano. Este proceso se puede repetir varias veces hasta que el Administrador haya terminado de graficar todos los planos que componen la orden de graficación, o hasta que el Administrador pulse sobre el botón *Terminar Orden*.

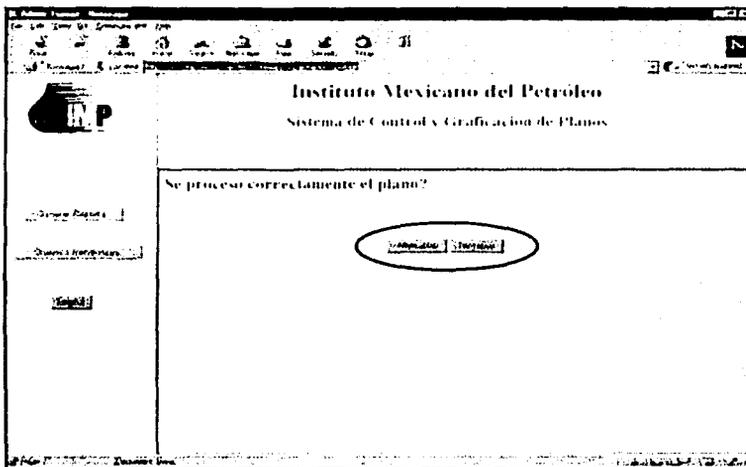


Figura 4.17. Pregunta del sistema sobre el proceso de graficación

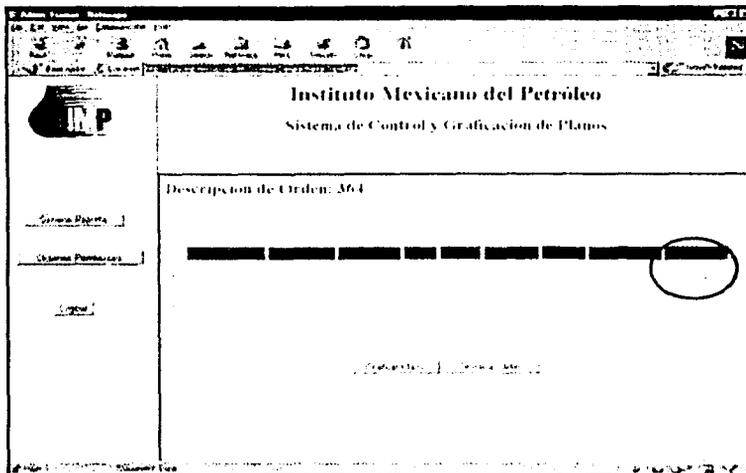


Figura 4.18. Cambio de la columna Estado en la descripción de la orden

Cuando el Administrador pulsa sobre el botón *Terminar Orden* el sistema presenta la interfaz de la Figura 4.19, la cual le indica al Administrador que un correo electrónico ha sido enviado a la persona que realizó la solicitud de graficación, mediante el cual se le avisa al usuario que su orden ha sido procesada. Adicionalmente en la misma interfaz gráfica se presenta el botón *Continuar*, el cual permite al Administrador seguir con el procesamiento de otras ordenes de graficación.

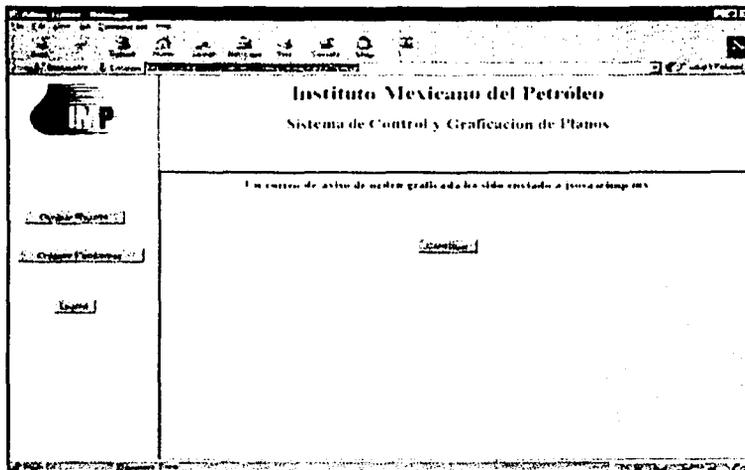


Figura 4.19. Envío de correo electrónico

4.3. Validación de Formularios HTML

Cuando se realiza una solicitud de graficación utilizando el Sistema de Control y Graficación, existen varios formularios HTML en los que el usuario tiene que proporcionar información para que la solicitud se procese correctamente, en estos formularios se hace necesaria una validación de los datos proporcionados por el usuario, verificando principalmente que el tipo de datos correspondan con los solicitados.

Para realizar esta validación de datos se tienen dos alternativas: validar la información utilizando los servlets controladores del sistema, o validar los datos haciéndolo del lado del cliente utilizando JavaScript [Dan96].

Por resultar un proceso más sencillo, se ha decidido validar la información proporcionada por el usuario utilizando JavaScript del lado del cliente. De esta forma, cualquier dato que el usuario proporcione y que no corresponda con el tipo de dato solicitado, hará que el sistema muestre un mensaje de aviso indicándole al usuario el posible error que ha cometido al teclear la información solicitada por el sistema, además del aviso mostrado, el sistema impedirá que se envíen los datos al servidor si no son validados correctamente.

La Figura 4.20, muestra un ejemplo de la validación de datos que el sistema realiza cuando un usuario proporciona datos de diferente tipo al solicitado, mostrando un mensaje del error cometido por parte del usuario para que se pueda corregir el valor y hasta que los valores estén correctamente validados puedan ser enviados al servidor.

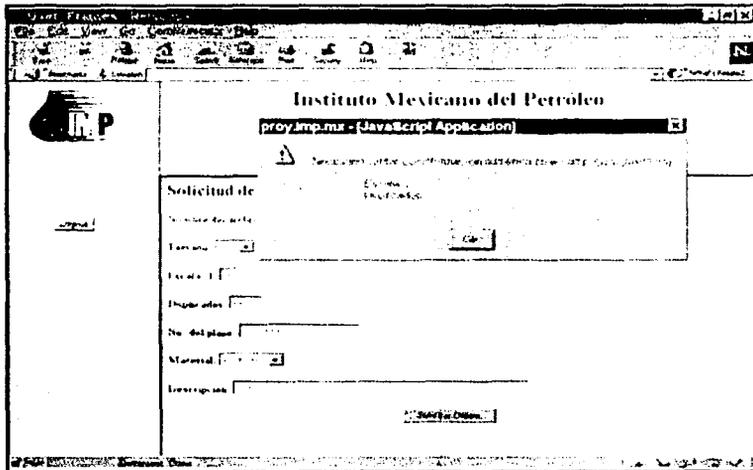


Figura 4.20. Validación de datos con JavaScript del lado del Cliente

La validación de los datos se hace de una manera similar en todos los formularios HTML en donde un usuario tenga que proporcionar datos.

4.4. Construcción de las Clases de Objetos del Sistema

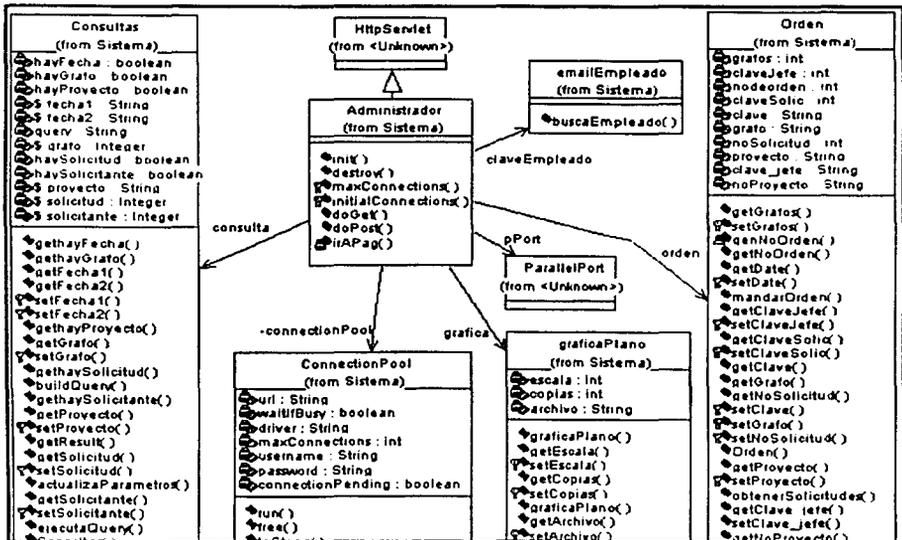
En esta parte se muestra la construcción y se describe el funcionamiento de las clases que representan a los objetos del Sistema de Control y Graficación.

4.4.1. Descripción de la Clase *Administrador*

La clase *Administrador* es el Servlet encargado de procesar todas las opciones que tiene un usuario con un permiso de *Administrador*, como son las opciones de generación de reportes o la de procesamiento de ordenes de graficación pendientes. Dependiendo de las opciones que el usuario elija, el Servlet captura todos los parámetros indicados o seleccionados, y de acuerdo con los valores de dichos parámetros, el Servlet redirecciona el valor de los mismos hacia diferentes objetos, que son los encargados de realizar los cálculos o la lógica de la aplicación, y que devuelven valores al Servlet para que éste realice nuevamente otra redirección.

Por tratarse de un Servlet, la clase *Administrador* tiene que extender de la clase *HttpServlet* y sustituir los métodos *doGet()* y *doPost()*, esto con la finalidad de poder manejar datos que sean enviados mediante peticiones GET o POST a través de un navegador web. En el método *init()* se establecen los parámetros necesarios para poder instanciar un objeto del tipo *connectionPool*, que es el que nos permitirá tener varias conexiones con la base de datos del sistema, el método *init()* se ejecuta siempre que un usuario con permiso de *Administrador* levanta el Servidor.

Los métodos *initialConnections()* y *maxConnections()* se refieren a la cantidad de conexiones abiertas que se tendrán con la base de datos. El método *irAPag()* es el responsable de realizar todas las redirecciones que el Servlet ejecuta de acuerdo con el procesamiento de los datos, y por último, el método *destroy()*, que cierra todas las conexiones existentes con la base de datos, este método se ejecuta cuando el Servidor destruye el Servlet y ocurre cuando un usuario con permiso de *Administrador* baja el Servidor.

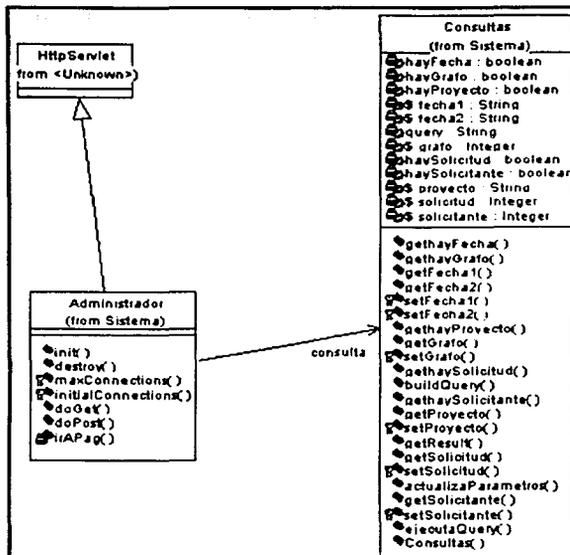


TESIS CON
 FALLA DE ORIGEN

Figura 4.23. Diagrama de la Clase Administrador

4.4.2. Descripción de la Clase Consultas

La clase *Consultas* es un Javabeen cuya función principal es la de formar un query de forma dinámica. Cuando un usuario con permiso de Administrador desea realizar un reporte es necesario que se le indique al sistema los parámetros con los cuales se realizará la búsqueda, debido a que dichos parámetros pueden cambiar dependiendo del reporte solicitado, es necesario un Javabeen que reconozca los parámetros solicitados y que con ellos pueda formar un query válido para la base de datos. Uno de los métodos más importantes dentro de la clase *Consultas* es *builtQuery()*, que se encarga de verificar las propiedades de clase y, con las propiedades solicitadas por el usuario, formar un query dinámico que represente la búsqueda que el usuario solicita. El método *ejecutaQuery()* es el encargado de realizar el query en la base de datos, existen además métodos del tipo *get()* y *set()*, que son característicos dentro del funcionamiento de un Javabeen, y que nos sirven para manipular los parámetros necesarios.



TESIS CON
 FALLA DE ORIGEN

Figura 4.24. Diagrama de la Clase Consultas

4.4.3. Descripción de la Clase *Descripcion*

La clase *Descripcion* es un JavaBean que se encarga de manipular los parámetros que describen todas las características de un archivo de graficación, como son: nombre del archivo, tamaño, escala, duplicados, material, etc., esta manipulación se realiza a través de métodos del tipo *get()* y *set()*, característicos de todos los JavaBean. La clase cuenta además con un método llamado *genIDDescripcion()*, que genera un número único de *Descripcion* y que identifica a cada archivo de graficación. Se tiene además un método llamado *obtenerDescripcion()*, que realiza una consulta a la base de datos y obtiene los objetos del tipo *Descripcion* solicitados, existe también el método *descripcionSiguiente()*, que devuelve los objetos del tipo *Descripcion* obtenidos por el método anteriormente descrito.

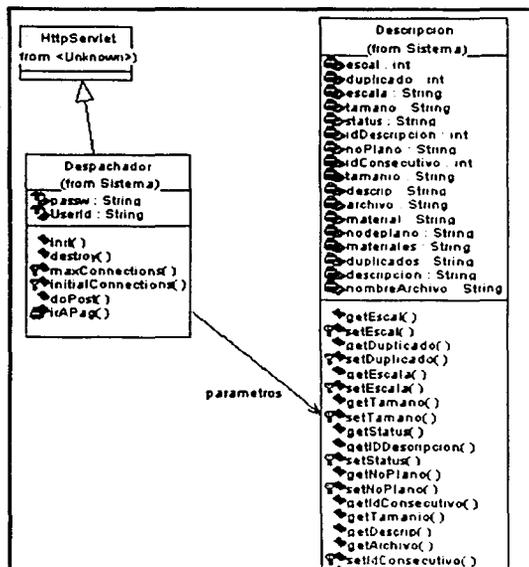


Figura 4.25. Diagrama de la Clase Descripcion

4.4.4. Descripción de la Clase *Despatchador*

La clase *Despatchador* es el Servlet encargado de todas las funciones de un usuario con permiso de Solicitante, su primera función es la autenticación del usuario, asignándole una interfaz al usuario de acuerdo con el permiso que tenga su cuenta. Posteriormente se encarga de obtener todos los parámetros para formar el objeto *Descripcion* y al objeto *Orden*.

La clase *Despatchador* realiza varias redirecciones para poder guiar al usuario hasta obtener un objeto *Orden* que sea válido para poder ser procesado por el sistema. Al igual que la clase *Administrador* el Servlet *Despatchador* extiende de *HttpServlet* y cuenta con los métodos *doGet()* y *doPost()*, con la finalidad de poder manejar datos que sean enviados mediante peticiones GET o POST a través de un navegador web. Sustituye el método *init()* de la misma forma que lo hace el Servlet *Administrador*, estableciendo todos los parámetros que permitan la instancia de un objeto

connectionPool, cuenta con los mismos métodos *initialConnections()* y *maxConnections()*, que se refieren a la cantidad de conexiones que se establecerán con la base de datos.

Se tiene también el mismo método *irAPag()*, ya que también realiza constantes redirecciones dentro del sistema, y por último, cuenta con el método *destroy()* que es característico de todos los Servlets y que se encarga de cerrar todas las conexiones que se establecieron con la base de datos.

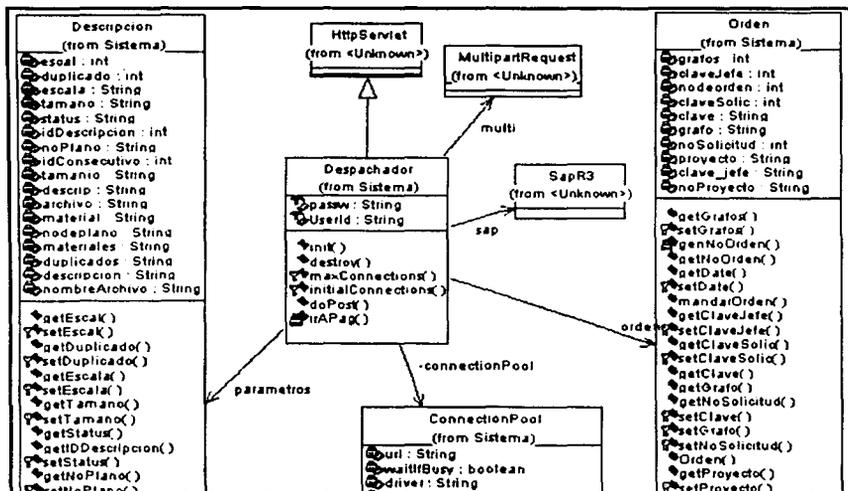


Figura 4.26. Diagrama de la Clase Despachador

4.4.5. Descripción de la clase *graficaPlano*

La clase *graficaPlano* es un JavaBean que contiene tres métodos importantes, el primero llamado *consultaStatus()*, que realiza una consulta hacia la base de datos con el fin de indicar si un archivo de graficación ya ha sido procesado o se encuentra en estado pendiente, el siguiente método llamado *graficarArchivo()*, que manda a graficar el archivo seleccionado hacia la unidad de graficación cuando un usuario ha accedido al sistema con un permiso de Administrador.

Una vez que el archivo ha sido graficado correctamente por el sistema, el objeto *graficaPlano* se encarga de cambiar el status al archivo de graficación, esta acción la realiza por medio del método *modificaStatus()*. La clase contiene además métodos del tipo *get()* y *set()*, con el propósito de manipular los parámetros recibidos por parte del Servlet *Administrador* y que son necesarios para poder ejecutar los métodos de consulta y de modificación del status de un archivo de graficación.

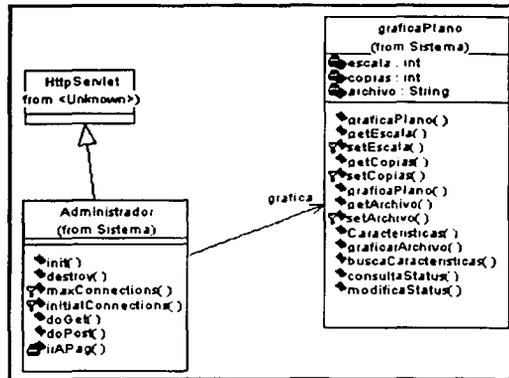


Figura 4.27. Diagrama de la Clase *graficaPlano*

4.4.6. Descripción de la Clase *Orden*

La clase *Orden* es un JavaBean cuya función principal es generar objetos que representen un orden de graficación válida para el sistema. Un objeto *Orden* esta compuesto de uno o más objetos del tipo *Descripcion*, que proporciona todas las características de un archivo de graficación, consta también del nombre del proyecto al cual se asignará la orden de graficación, de la clave de la persona que elabora la orden de graficación, de la fecha de la elaboración de la orden y del grafo, que es un número que hace referencia a una actividad dentro del proyecto al cual se está asignando la orden de graficación. Además, un objeto *Orden* contiene un número que es único para cada orden de graficación, y que servirá para identificar a todas y a cada una de las solicitudes de graficación procesadas por el sistema, dicho número es generado por el método llamado *genNoOrden()*.

La clase cuenta con métodos del tipo *get()* y *set()*, con los que se realiza la manipulación de todos los elementos que conforman un objeto *Orden*. Existen además los métodos *mandarOrden()*, que realiza una inserción en la base de datos de todos los elementos que conforman al objeto, *actualizaStatusOrden()*, que cambia el status que el objeto tiene dentro de la base de datos cuando ésta ha sido procesada, *obtenerSolicitudes()* y *solicitudSiguiente()*, que son consultas que se ejecutan cuando un usuario, con permiso de Administrador, selecciona la opción de procesar las solicitudes pendientes.

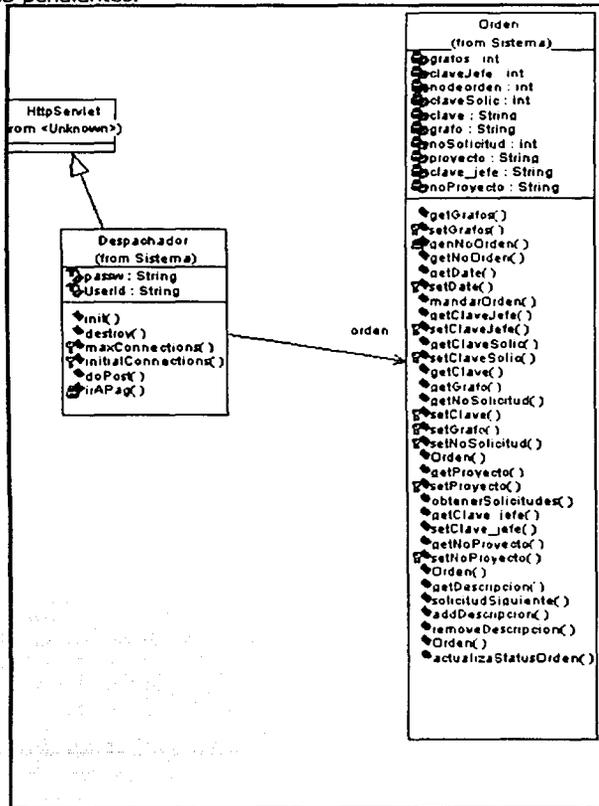


Figura 4.28. Diagrama de la Clase Orden

4.4.7. Descripción de la Clase *UsuarioSistema*

La clase *UsuarioSistema* es un JavaBean que es el encargado de realizar el proceso de autenticación del usuario a través de su método *buscarUsuario()*, el cual realiza una consulta a la base de datos y verifica en primer término que el usuario exista dentro de la misma y, posteriormente, revisa el permiso que tiene dicho usuario dentro del sistema. Del resultado de este método depende la interfaz que se le mostrará al usuario cuando éste haga uso del Sistema de Control y Graficación.

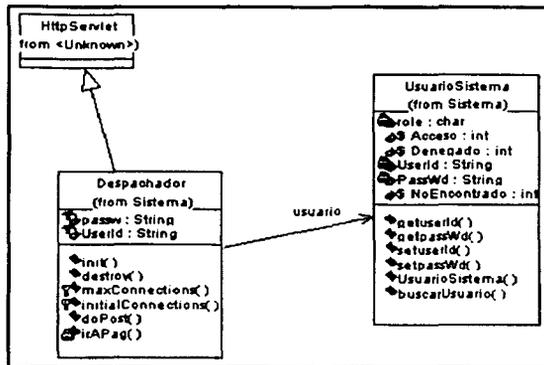


Figura 4.29. Diagrama de la Clase *UsuarioSistema*

5

Pruebas finales e Instalación

Resumen

En este capítulo se describe el plan de pruebas que se programó para verificar la correcta funcionalidad del sistema, el desempeño, los resultados que se han obtenido y los detalles de instalación del sistema de control y graficación de planos.

5.1. Pruebas al Sistema de Control y Graficación

Se realizó un plan de pruebas para cada caso de uso presentado en la parte de diseño del sistema, de esta forma se trata de evaluar cada procedimiento que el sistema ejecuta conforme a las peticiones de cualquier usuario. Posteriormente a cada plan de prueba, se han diseñado los procedimientos de prueba y evaluación, para que finalmente se verifique la ejecución de las pruebas diseñadas para cada caso de uso de acuerdo al plan de la prueba.

5.1.1. Prueba de Caso de Uso "Autentificar"

Objetivo: Verificar la funcionalidad del sistema para permitir el acceso al sistema.

Plan de Prueba:

Introducción:	Realizar el proceso de autenticación de un usuario del sistema.
Estrategia:	Autenticación del usuario considerando los siguientes casos: <ul style="list-style-type: none">- Usuario con una cuenta válida.- Usuario con cuenta no-válida (clave de empleado incorrecta).- Usuario con cuenta no-válida (contraseña incorrecta).- Autenticación con datos nulos.
Programa de Actividades:	<ul style="list-style-type: none">- Petición de la dirección de inicio del sistema a través de un navegador web.- Pulsar la opción de Login.- Establecer los datos de clave de empleado y contraseña conforme a la estrategia de prueba.

Diseño de la Prueba:

Caso de Prueba	Procedimiento de Prueba
Cuenta válida	<ul style="list-style-type: none">- Capturar clave de empleado y contraseña válidos.- Pulsar botón Continuar.
Cuenta no-válida (clave de empleado incorrecta)	<ul style="list-style-type: none">- Capturar clave de empleado incorrecta.- Capturar contraseña correcta.- Pulsar botón Continuar.
Cuenta no-válida	<ul style="list-style-type: none">- Capturar clave de empleado correcta.

(contraseña incorrecta)	- Capturar contraseña incorrecta. - Pulsar botón Continuar.
Datos Nulos	- Pulsar botón Continuar.

Ejecución de Pruebas:

Caso de Prueba	Exito
Cuenta válida	X
Cuenta no-válida (clave de empleado incorrecta)	X
Cuenta no-válida (contraseña incorrecta)	X
Datos Nulos	X

5.1.2. Prueba de Caso de Uso "Crear Cuenta"

Objetivo: Registrar un nuevo usuario del sistema.

Plan de Prueba:

Introducción:	Realizar el proceso de creación de una cuenta de usuario del sistema.
Estrategia:	Creación de una cuenta de usuario considerando los siguientes casos: <ul style="list-style-type: none"> - El usuario no tiene cuenta válida del sistema. (Cuenta Nueva) - El usuario posee una cuenta de usuario válida del sistema. (Duplicación de Cuenta)
Programa de Actividades:	<ul style="list-style-type: none"> - Petición de la dirección de inicio del sistema a través de un navegador web. - Pulsar la opción de Registro. - Seleccionar los datos de Competencia y Nombre del empleado. - Proporcionar la contraseña. - Comprobar la existencia o no existencia de la cuenta del usuario que se quiera registrar en la base de datos del sistema.

Diseño de la Prueba.

Caso de Prueba	Procedimiento de Prueba
Cuenta Nueva	<ul style="list-style-type: none"> - Seleccionar el nombre de Competencia. - Seleccionar el nombre de empleado. - Establecer contraseña. - Confirmar contraseña.
Duplicación de Cuenta	<ul style="list-style-type: none"> - Seleccionar el nombre de Competencia. - Seleccionar el nombre de empleado. - Establecer contraseña. - Confirmar contraseña. - Envío de un correo electrónico a la persona que registra indicándole que ya se había registrado con anterioridad, enviando también la contraseña en caso de que el usuario la haya olvidado. Con esto no se permite la duplicación de usuarios registrados.

Ejecución de Pruebas:

Caso de Prueba	Éxito
Cuenta Nueva	X
Duplicación de Cuenta	X

5.1.3. Prueba de Caso de Uso "Crear Orden"

Objetivo: Registrar una orden de graficación para solicitar la graficación de planos.

Plan de Prueba:

Introducción:	Realizar el proceso de creación de una orden de graficación.
Estrategia:	Creación de una orden de graficación considerando los siguientes casos: <ul style="list-style-type: none"> - Hacer uso de la opción Login, con el objetivo de abrir una sesión con el sistema. - Hacer uso de la opción Crear Orden para registrar una nueva solicitud de graficación. - Hacer uso de la opción Agregar Plano. Agregando varios planos a graficar. - El usuario proporciona datos nulos al

	<p>establecer los datos de un plano por graficar.</p> <ul style="list-style-type: none"> - El usuario proporciona datos incorrectos al establecer los datos de un plano por graficar. - Hacer uso de la opción Remover Plano. Remover todos los planos agregados a la orden. - Hacer uso de la opción Terminar Orden. Registrando en la base de datos la orden y todos sus detalles de planos a graficar. - Hacer uso de la opción Logout para terminar la sesión con el sistema.
Programa de Actividades:	<ul style="list-style-type: none"> - Hacer uso de la opción de Login, para abrir la sesión con el sistema. - Autenticarse con una cuenta válida en el sistema. - Creación de varias ordenes de graficación de acuerdo a la estrategia de prueba.

Diseño de la Prueba:

Caso de Prueba	Procedimiento de Prueba
Datos Nulos	<ul style="list-style-type: none"> - En la opción de Agregar Planos a una orden, verificar que el usuario haya asignado valores a los campos para inserción. (Validación con JavaScript desde el Navegador).
Datos Incorrectos	<ul style="list-style-type: none"> - En la opción de Agregar planos a una orden, verificar que el usuario haya asignado valores a los campos para inserción y que los valores sean del tipo correcto de acuerdo a la base de datos. (Validación con JavaScript desde el Navegador).
Crear una orden sin planos a graficar	<ul style="list-style-type: none"> - No se puede terminar una orden si no se ha agregado al menos un plano a graficar.
Crear una orden con varios planos	<ul style="list-style-type: none"> - Construir una orden con varios planos a graficar y terminar la orden. Verificar la orden junto con todos sus planos a graficar en la base de datos.
Crear una orden agregando y removiendo planos a graficar	<ul style="list-style-type: none"> - Crear una orden utilizando las opciones Agregar Plano y Remover Plano de manera repetitiva y en distinto orden.

<p>Crear varias ordenes de graficación</p>	<ul style="list-style-type: none"> - Utilizar la opción de crear orden varias veces y verificar que el usuario es notificado con su numero de orden. Y también que las ordenes han sido registradas en la base de datos del sistema.
--	---

Ejecución de Pruebas:

Caso de Prueba	Exito
Datos Nulos	X
Datos Incorrectos	X
Crear una orden sin planos a graficar	X
Crear una orden con varios planos	X
Crear una orden agregando y removiendo planos a graficar	X
Crear varias ordenes de graficación	X

5.1.4. Prueba de Caso de Uso "Realizar Reportes"

Objetivo: Generar reportes del uso del sistema, en sus diferentes opciones.

Plan de Prueba:

<p>Introducción:</p>	<p>Realización de reportes utilizando todos los criterios de búsqueda de forma independiente y de forma conjunta.</p>
<p>Estrategia:</p>	<p>Realización de varios tipos de reportes considerando los siguientes casos:</p> <ul style="list-style-type: none"> - Hacer uso de la opción Login utilizando una cuenta de Administrador, con el objetivo de abrir una sesión con el sistema. - Hacer uso de la opción Generar Reporte para que se permita la realización de consultas a la base de datos del sistema. - Hacer uso de la opción Nueva Búsqueda de forma repetida, con el objetivo de realizar distintas consultas a la base de datos, cambiando el numero y el orden de los criterios de búsqueda seleccionados para la realización de la consulta. <ul style="list-style-type: none"> - Seleccionar cada uno de los criterios de búsqueda de forma independiente, y realizar las consultas con cada uno de

	<p>ellos.</p> <ul style="list-style-type: none"> - Seleccionar dos criterios de búsqueda distintos y realizar varias consultas. - Seleccionar tres criterios de búsqueda distintos y realizar varias consultas. - Seleccionar cuatro criterios de búsqueda distintos y realizar varias consultas. - Seleccionar todos los criterios de búsqueda que el sistema presenta y realizar varias consultas. <p>- En cada una de las consultas realizadas, el usuario:</p> <ul style="list-style-type: none"> - Proporciona datos nulos al establecer los valores de los criterios de búsqueda empleados en cada consulta. - Proporciona datos incorrectos al establecer los valores de los criterios de búsqueda empleados en cada consulta. <p>- Hacer uso de la opción Logout para terminar la sesión con el sistema.</p>
Programa de Actividades:	<ul style="list-style-type: none"> - Hacer uso de la opción de Login, para abrir la sesión con el sistema. - Autenticarse en el sistema con una cuenta válida de Administrador. - Realización de varias consultas a la base de datos del sistema de acuerdo a la estrategia de prueba.

Diseño de la Prueba:

Caso de Prueba	Procedimiento de Prueba
Consultas con un criterio de búsqueda	<ul style="list-style-type: none"> - En la opción Nueva Búsqueda, verificar que el sistema realiza correctamente distintas consultas en la base de datos, utilizando solamente un criterio de búsqueda.
Consultas utilizando dos criterios de búsqueda	<ul style="list-style-type: none"> - En la opción Nueva Búsqueda, verificar que el sistema realiza correctamente las consultas solicitadas utilizando dos de los criterios de búsqueda permitidos.
Consultas utilizando tres criterios de búsqueda	<ul style="list-style-type: none"> - En la opción Nueva Búsqueda, verificar que el sistema realiza correctamente las consultas solicitadas utilizando tres de los criterios de búsqueda permitidos.

Consultas utilizando cuatro criterios de búsqueda	- En la opción Nueva Búsqueda, verificar que el sistema realiza correctamente las consultas solicitadas utilizando cuatro de los criterios de búsqueda permitidos.
Consultas utilizando todos los criterios de búsqueda	- En la opción Nueva Búsqueda, verificar que el sistema realiza correctamente las consultas solicitadas utilizando todos los criterios de búsqueda permitidos.
Datos Nulos	- En la opción Nueva Búsqueda, verificar que el usuario haya seleccionado por lo menos uno de los criterios de búsqueda permitidos para la realización de consultas. (Validación con JavaScript desde el Navegador).
Datos Incorrectos	- En la opción Nueva Búsqueda, verificar que el usuario haya asignado valores a los campos de criterios de búsqueda y que los valores sean del tipo correcto de acuerdo a la base de datos. (Validación con JavaScript desde el Navegador).

Ejecución de Pruebas:

Caso de Prueba	Exito
Consultas con un criterio de búsqueda	X
Consultas con dos criterios de búsqueda	X
Consultas con tres criterios de búsqueda	X
Consultas con cuatro criterios de búsqueda	X
Consultas con todos los criterios de búsqueda	X
Datos Nulos	X
Datos Incorrectos	X

5.1.5. Prueba de Caso de Uso "Ordenes Pendientes"

Objetivo: Graficar los planos de las solicitudes de graficación pendientes.

Plan de Prueba:

Introducción:	Realizar el procesamiento de las ordenes de graficación pendientes.
Estrategia:	<p>Procesar varias solicitudes de graficación considerando los siguientes casos:</p> <ul style="list-style-type: none">- Hacer uso de la opción Login utilizando una cuenta con permiso de Administrador, con el objetivo de abrir una sesión con el sistema.- Hacer uso de la opción Solicitudes Pendientes para procesar una orden de graficación.- Seleccionar una orden de graficación.- Hacer uso de la opción Graficar Plano. Graficando todos los planos que comprenden la orden de graficación.<ul style="list-style-type: none">- El usuario no realiza ninguna selección de los planos presentados.- Hacer uso de la opción Graficar Plano. Graficando solo algunos de los planos que comprenden a la orden de graficación.- Hacer uso de la opción Terminar Orden. Registrando en la base de datos la orden procesada y todos los detalles de los planos graficados.- Hacer uso de la opción Logout para terminar la sesión con el sistema.
Programa de Actividades:	<ul style="list-style-type: none">- Hacer uso de la opción de Login, para abrir la sesión con el sistema.- Autenticarse con una cuenta de Administrador en el sistema.- Procesamiento de varias ordenes de graficación de acuerdo a la estrategia de prueba.

Diseño de la Prueba:

Caso de Prueba	Procedimiento de Prueba
Sin selección de planos	- En la opción de Graficar Plano, verificar que el usuario haya seleccionado alguno de los planos que comprenden la orden de graficación. (Validación con JavaScript desde el Navegador).
Graficando los planos de la solicitud de graficación	- Verificar que el estado del plano seleccionado haya cambiado después de haber hecho uso de la opción Graficar Plano.
Terminando el proceso de la solicitud de graficación	- Verificar en la base de datos del sistema, que el estado de la solicitud de graficación haya cambiado después de haber hecho uso de la opción Terminar Orden.

Ejecución de Pruebas:

Caso de Prueba	Exito
Sin selección de planos	X
Graficando los planos de la solicitud de graficación	X
Terminando el proceso de la solicitud de graficación	X

Esta forma de realización de pruebas, nos permitió detectar los problemas que el sistema de control y graficación presentaba y al mismo tiempo corregirlos, teniendo al termino del ciclo de realización de pruebas, un sistema depurado y listo para su funcionamiento.

5.2. Instalación del nuevo Sistema de Control y Graficación

El proceso de instalación del Sistema de Control y Graficación fue realizado de la siguiente forma:

1. Instalación de jakarta-tomcat-3.2.3 en la máquina destino de la aplicación [JtoB01].
2. Configuración de un contexto válido de desarrollo dentro del archivo jakarta-tomcat-3.2.3/conf/server.xml. En este caso el contexto definido recibe el nombre "jsp" y todos los documentos correspondientes al contexto se encuentran en el subdirectorio Alberto del disco local F.

```
<Context path="/jsp"
docBase="F:/Alberto"
crossContext="false"
debug="0"
reloadable="true"
/>
```

3. Configuración del juego de conexiones para SAP, realizada mediante la generación del archivo web.xml y la utilización de una Servlet, ambos instalados dentro del contexto definido en jakarta-tomcat-3.2.3

```

<servlet>
  <servlet-name>
    SapConnect
  </servlet-name>
  <servlet-class>
    sap.imp.ServletSapPool
  </servlet-class>

  <init-param>
    <param-name>poolid</param-name>
    <param-value>R3</param-value>
  </init-param>
  <init-param>
    <param-name>maxconn</param-name>
    <param-value>10</param-value>
  </init-param>
  <init-param>
    <param-name>client</param-name>
    <param-value>350</param-value>
  </init-param>
  <init-param>
    <param-name>userid</param-name>
    <param-value>impjmflores</param-value>
  </init-param>
  <init-param>
    <param-name>password</param-name>
    <param-value>jflores</param-value>
  </init-param>
  <init-param>
    <param-name>language</param-name>
    <param-value>EN</param-value>
  </init-param>
  <init-param>
    <param-name>appserv</param-name>
    <param-value>192.168.137.2</param-value>
  </init-param>
  <init-param>
    <param-name>system</param-name>
    <param-value>02</param-value>
  </init-param>

  <load-on-startup></load-on-startup>
</servlet>

```

-
4. Configuración del juego de conexiones para la base de datos, realizada mediante la generación del archivo web.xml y la utilización de una Servlet, ambos instalados dentro del contexto definido en jakarta-tomcat-3.2.3

```
<servlet>
  <servlet-name>
    Despachador
  </servlet-name>
  <servlet-class>
    Sistema.Despachador
  </servlet-class>

  <init-param>
    <param-name>driver</param-name>
    <param-value>oracle.jdbc.driver.OracleDriver</param-value>
  </init-param>
  <init-param>
    <param-name>url</param-name>
    <param-value>jdbc:oracle:thin:@192.168.143.226:1521:ora8i</param-value>
  </init-param>
  <init-param>
    <param-name>username</param-name>
    <param-value>amh</param-value>
  </init-param>
  <init-param>
    <param-name>password</param-name>
    <param-value>amh</param-value>
  </init-param>

  <load-on-startup></load-on-startup>
</servlet>
```

-
5. Configuración del puerto paralelo utilizado para la graficación, realizada por medio del archivo web.xml y el uso de una Servlet instalada dentro del contexto definido en jakarta-tomcat-3.2.3

```
<servlet>
  <servlet-name>
    HabilitaPuerto
  </servlet-name>
  <servlet-class>
    Sistema.HabilitaPuerto
  </servlet-class>

  <init-param>
    <param-name>puerto</param-name>
    <param-value>LPT1</param-value>
  </init-param>
  <init-param>
    <param-name>mensaje</param-name>
    <param-value>Escritor en paralelo</param-value>
  </init-param>
  <init-param>
    <param-name>tiempo</param-name>
    <param-value>30000</param-value>
  </init-param>

  <load-on-startup></load-on-startup>
</servlet>
```

6. Instalación de archivos ejecutables (*.java), y archivos compilados (*.class), dentro del subdirectorio F:/Alberto/Web-inf/classes/Sistema, el cual pertenece al contexto válido configurado con anterioridad.
7. Instalación de interfaces gráficas de usuario, páginas JSP, dentro del contexto válido para la aplicación. Todas la interfaces deberán estar instaladas dentro del subdirectorio F:/Alberto/jsp, el cual pertenece al contexto configurado.
8. Instalación de los controladores JDBC dentro del directorio jakarta-tomcat-3.2.3/lib, los cuales permitirán la comunicación y la realización de transacciones con la base de datos.

-
9. Generación del archivo `Graficacion.war`, el cual contiene toda la aplicación dentro de un contexto configurado para su funcionamiento.

```
f:\jsp>jar -cvf f:\Alberto\Graficacion.war .
```

6

Conclusiones

Resumen

En este capítulo se presentan las conclusiones a que se llegó con el trabajo desarrollado, los logros obtenidos y las posibles actualizaciones que pudiera tener el sistema de control y graficación.

6.1. Conclusiones

El presente trabajo de tesis ha contribuido al mejoramiento de la forma de graficación de planos y del control de la información relacionada con ésta actividad dentro del área de Graficación del Instituto Mexicano del Petróleo. Se han reducido los tiempos de operación, mejorando los procesos que antes se realizaban manualmente, ya que los usuarios ahora pueden usar la red para enviar sus archivos a graficar, almacenándose en una base de datos toda la información asociada a los proyectos en los cuales tiene un impacto económico la graficación de planos, se ha incorporado la capacidad de poder realizar reportes de acuerdo a diferentes características con el propósito de poder tener estadísticas y gastos en el uso de materiales como papel, y poder hacer cargos más fidedignos a los proyectos, además de obtener un sistema accesible desde cualquier lugar dentro y fuera de la Institución a través de Internet o Intranet, el sistema cuenta con un mecanismo de seguridad para su acceso y permite el uso de diferentes permisos o perfiles de usuario para su operación, adicionalmente integra datos reales de personal y proyectos, con los datos actuales almacenados en el sistema ERP (Enterprise Resource Planning) de SAP.

En la etapa de diseño del Sistema de Control y Graficación de Planos se ha desarrollado haciendo uso de la notación gráfica del Lenguaje de Modelado Unificado (UML), describiendo el funcionamiento del sistema por medio de diagramas de Casos de Uso, diagramas de Clases y diagramas de Secuencias.

El Sistema de Control y Graficación de Planos fue desarrollado utilizando el Modelo 2 de Programación de JSP, consiguiendo con esto, la separación de la lógica del sistema de la presentación del mismo. Factor importante al momento de realizar cambios o depuraciones al sistema entre diferentes entidades, como lo son la de diseño gráfico y la de programadores, facilitando el mantenimiento.

El uso de Servlets, JavaServer Pages y JavaBeans, para la etapa de diseño y construcción del sistema, ha permitido poder utilizar el patrón de diseño Modelo/Vista/Controlador (MVC). Con los Servlets controlamos las actividades de los usuarios, con los JavaBeans realizamos la lógica de la aplicación y con los JavaServer Pages construimos las vistas o presentación de la aplicación a través del uso de formularios HTML y páginas de resultados. Por lo anterior podemos decir, que las herramientas de desarrollo seleccionadas, resultaron adecuadas para el propósito planteado en este trabajo de tesis.

En cuanto a la lógica de control de acceso, se desarrollaron dos Servlets, una para todos los usuarios que realizan solicitudes de graficación, y otra para los usuarios administradores, encargados de realizar el procesamiento de las solicitudes, de tal forma que el Sistema de Control y Graficación nos permita controlar las transacciones de todos los usuarios, garantizando la privacidad de ciertos datos a los que solo determinados usuarios pueden tener acceso.

El sistema se ha desarrollado utilizando Tomcat, que es un motor de Servlets y JSP, el cual permite la escalabilidad del sistema, ya que pueden conectarse uno o varias instancias de Tomcat con diferentes servidores web.

Para mejorar la eficiencia de las transacciones entre la base de datos y el Sistema de Control y Graficación, se ha utilizado un objeto de múltiples conexiones a la base de datos, previamente establecido, *ConnectionPool*, para administrar las conexiones disponibles y reciclarlas conforme un usuario haga uso del sistema. Debido al uso de la clase *ConnectionPool*, se ha logrado reducir el tiempo de operación del sistema de forma significativa.

En cuanto a la configuración del sistema se ha optado por utilizar el archivo de configuración *web.xml*, instalado dentro del subdirectorio *Web-inf* del contexto configurado en Tomcat, con el fin de establecer los parámetros necesarios para realizar las conexiones con la base de datos, los parámetros utilizados para la conexión con SAP y los parámetros que se necesitan para la configuración de la unidad de graficación y del puerto paralelo. De esta forma, los valores de los parámetros establecidos a través del *ServletContext* están disponibles para todos los Servlets del servidor Tomcat, proporcionando una mayor facilidad de mantenimiento y de configuración del sistema.

En la etapa de pruebas se diseñó un conjunto de pruebas para comprobar la confiabilidad del sistema y garantizar la calidad del software, mediante este conjunto de pruebas se pudieron detectar varios errores en la programación y se pudieron corregir sistemáticamente. Por ejemplo, con la interfaz *ResultSet*, se presentó un error de programación que consumía todos los cursores con la base de datos hasta agotarse, la razón era porque solo se abrían pero nunca se cerraban.

6.2. Trabajos futuros

A pesar de haber resuelto muchos problemas en operación de la graficación de planos, este sistema aún es susceptible de mejoras, y a continuación se listan algunas de las que considero importantes a realizar en trabajos futuros:

- Incrementar la seguridad del sistema mediante el uso de encriptación de datos o manejo de certificados SSL [SSL98].
- Transformar los JavaBeans desarrollados en nuestro sistema, en componentes Enterprise JavaBeans [EJB02], con el objetivo de aumentar la escalabilidad y la confiabilidad del sistema.
- Hacer que el Sistema de Control y Graficación de Planos funcione como un servicio web, con el fin de realizar todas las transacciones con la base de datos por medio de archivos en formato XML.
- Para aumentar la escalabilidad del sistema se podría distribuir la forma del proceso de graficación. La Figura 6.1 muestra la arquitectura del Sistema de Control y Graficación de Planos, el cual se compone de un servidor web relacionado con una instancia de Tomcat, dicha instancia debe estar instalada en la misma máquina en que se encuentra conectado el graficador, haciendo que las peticiones de graficación sean atendidas una por una. Sería recomendable utilizar una arquitectura como la mostrada en la Figura 6.2, donde varias instancias de Tomcat instaladas en diferentes máquinas, se comuniquen, por medio de un socket, con un nuevo Servlet encargado de recibir todos los archivos de graficación, el cual podría tener configuradas varias unidades de graficación a través de distintos puertos, logrando con esto, que se puedan atender la graficación de varios planos al mismo tiempo.

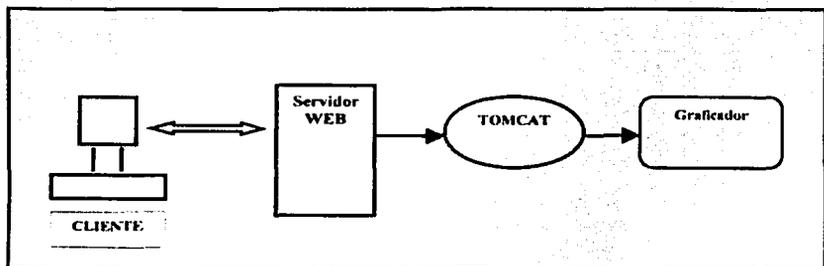


Figura 6.1. Arquitectura del Sistema de Control y Graficación de Planos

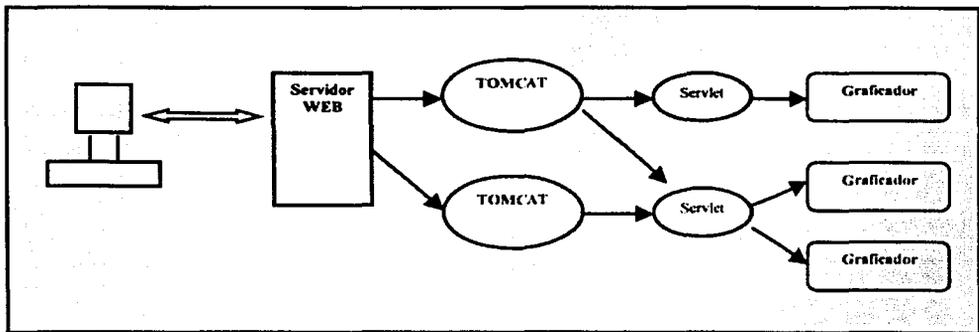


Figura 6.2. Arquitectura propuesta

Bibliografía

- [Apa01]** Apache HTTP Server Project. "Apache 1.3"
<http://httpd.apache.org>
- [Dan96]** Arman Danesh, "JavaScript in a week", Sams Net, 1996.
- [EJB02]** Enterprise JavaBeans Technology. 2002
<http://java.sun.com/products/ejb/docs.html>
- [FoSA99]** Martin Fowler, Kendall Scott, "UML gota a gota", PEARSON 1999.
- [FoSB99]** Martin Fowler, Kendall Scott, "UML gota a gota", capítulo 3, PEARSON 1999.
- [FoSC99]** Martin Fowler, Kendall Scott, "UML gota a gota", capítulos 4-5, PEARSON 1999.
- [FoSD99]** Martin Fowler, Kendall Scott, "UML gota a gota", capítulo 6, PEARSON 1999.
- [HalA01]** Marty Hall, "Servlets y JavaServer Pages", Prentice Hall, capítulos 1-9, 2001.
- [HalB01]** Marty Hall, "Servlets y JavaServer Pages", Prentice Hall, capítulos 10-12, 2001.
- [HalC01]** Marty Hall, "Servlets y JavaServer Pages", Prentice Hall, capítulos 13-14, 2001.
- [JCO]** "Java Connector Tutorial". Thomas G. Schuessler, 2000.
<http://www.sap.com/australia/teched/tuesday/HO3FewsterJCOTutorial.pdf>
- [JDBC02]** JDBC Data Access API. 2002
<http://java.sun.com/products/jdbc/>
- [Jso02]** JavaSoft, Sun Microsystem. "Java Development Kit JDK 1.3.1",
<http://java.sun.com/j2se/1.3/>

-
- [JtoA01]** The Jakarta Project. "Jakarta-Tomcat-3.2.3",
<http://jakarta.apache.org/tomcat/index.html>
- [JtoB01]** The Jakarta Project. "User's Guide",
<http://jakarta.apache.org/tomcat/tomcat-3.2-doc/uguide/tomcat-ug.html>
- [Paw01]** Introduction to the J2EE Platform.
<http://developer.java.sun.com/developer/technicalArticles/J2EE/Intro/index.html#arch>
- [SAP]** Systemanalyse und Programmentwicklung ("Systems Analysis and Program Development").
<http://www.sapfans.com/sapfans/saphist.htm>
- [SSL98]** Introduction to SSL.
<http://developer.netscape.com/docs/manuals/security/sslin/contents.htm>
- [Tog02]** TogetherSoft.
<http://www.togethersoft.com/downloads/index.jsp>