



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**



**FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"**

**SISTEMA INTEGRAL DE SERVICIOS PARA LA
COMISIÓN FEDERAL DE MEJORA REGULATORIA
(COFEMER) SIICOFE.**

T E S I N A

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A:

FEDERICO SANCHEZ CONTRERAS

TUTOR: MTRO. EN C. MARCELO PÉREZ MEDEL



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.

AGRADECIMIENTOS.

A Dios.

Por darme la dicha de un momento muy especial en mi vida y gozar de salud y bienestar al lado de mi familia y seres queridos.

Mis padres.

Les agradezco por su confianza en mí y en el apoyo tanto humano como físico. Por su sabiduría y que gracias a ellos logre culminar mi carrera. Gracias padres míos les estoy eternamente agradecidos por todo lo que me han dado.

Mis profesores.

Que con su conocimiento transmitido han hecho de mí un buen estudiante apoyándome en lograr culminar con mi carrera y hacer de mí un hombre de bien.

A mi esposa (DIANA).

Le agradezco su paciencia y su confianza por haberme dado su apoyo y comprensión en la culminación de mi carrera. Gracias por darme la dicha tan grande de ser padre.

A mis hijos (FEDERICO, DAYANA YARELI y NATALY ANAHID).

Les brindo mi alegría y mi entrega para poder culminar satisfactoriamente este proyecto y con la firme propuesta de ser en ellos un ejemplo a seguir y superarme cada instante de mi vida.

Índice.	Pág.
Introducción.	I
I. Marco teórico.	
1.1 Introducción a los sistemas de información.	1
1.2 El enfoque de las Tecnologías de información.	1
1.3 Administración de los sistemas de información.	2
1.4 Requerimientos de los sistemas de información.	3
1.5 Ciclo de vida de un sistema de información.	4
1.6 Metodología de análisis y diseño de sistemas.	5
1.6.1 Modelo vista controlador.	5
1.6.2 Modelo UML.	6
1.6.3 Modelo RUP (IBM).	9
1.6.4 Modelo CMM.	9
1.7 Reglas de integridad.	10
1.7.1 Regla de integridad de entidades.	10
1.7.2 Regla de integridad referencial.	10
1.8 Hacia dónde vamos con JAVA.	11
1.9 Arquitectura de aplicaciones.	11
II Análisis del sistema.	
2.1 Definición y planteamiento del problema.	12
2.2 Definición de recursos.	15
2.2.1 Recursos humanos.	15
2.2.2 Recursos de hardware.	15
2.2.3 Recursos de software	15
2.2.4 Diagrama de actividades (GANT)..	15
2.3 Requerimientos de un sistema.	16
2.3.1 Requerimientos funcionales.	16
2.3.2 Requerimientos no funcionales.	16
2.4 El UML para el análisis del sistema.	17
2.5 Diagramas de caso de uso del sistema SIICOFÉ.	18
2.5.1 Caso de uso de sistema SIICOFÉ.	19
2.5.2 Diagramas de navegación para solicitar un Reporte.	24
2.5.3 Diagrama de secuencia para registrar un servicio.	25
2.5.4 Diagrama de Actividades.	26
2.5.5 Modelo Conceptual.	27
2.5.6 Diagrama general de clases.	28
2.5.7 Diagramas de clases a detalle.	29
2.5.8 Diagramas de clases de análisis.	30

	Pág.
III. Diseño del sistema SIICOFE	
3.1 Definición de Audiencia (Niveles de usuario)	31
3.2 Capa de Vista.	31
3.2.1 Mapas del sitio.	31
3.2.2 Árbol funcional.	32
3.2.3 Interface Web.	32
3.3 Capa de datos.	36
3.4 Capa de Negocio.	38
IV Implementación del Sistema Integral COFEMER (SIICOFE)	
4.1 ¿Qué probar antes de implantar el sistema?	42
4.2 Pruebas de interfaces y Contenidos.	42
4.2.1 Verificación de contenidos.	42
4.2.2 Sistema Automático.	42
4.2.3 Verificar las Interfaces.	43
4.3 Pruebas de funcionalidad y operación.	43
4.3.1 Validación de formularios.	43
4.3.2 Carga inicial.	44
4.3.3 Pruebas de Carga.	45
4.3.4 Seguridad.	45
CONCLUSIONES.	46
APÉNDICE 1.	47
APÉNDICE 2.	48
BIBLIOGRAFÍA.	55

INTRODUCCIÓN

El desarrollo de la TI (Tecnologías de Información), nos permite en la actualidad ir implementando sistemas a la medida de las necesidades de los usuarios finales. Con objetivos claros y concisos para optimizar tiempos y movimientos.

El SIICOFE (Sistema Integral COFEMER) permitirá enlazar dos áreas importantes como son: el área Administrativa y el área de Sistemas para optimizar tiempos de respuesta de las necesidades de los usuarios y por consiguiente mejorar el tiempo de atención de un usuario.

El sistema SIICOFE en una primera etapa abarcara el área de Soporte Técnico y el área Administrativa sin considerar la Nómina. En una segunda etapa se tomará en cuenta la Nómina y desarrollo de plantillas para su implementación de formatos.

El sistema SIICOFE permitirá tener un sistema de control de inventarios el cual debe de contener a todos los empleados y artículos de la COFEMER. Así como se tendrá un control de Servicios de Mantenimiento, Préstamo de equipo y cuentas de correo. Emitirá estadísticas para la adquisición de equipo en futuras licitaciones. Comparará con el SICORE la información que se tiene actual contra el sistema actual. Validará los datos del SICORE en formato XML con la información del sistema

El sistema reducirá costos en consumo de papel ya que muchas de las cosas que se elaboran utilizan mucho papel y de esta manera se tendrá la información en electrónico reduciendo horas de tiempo hombre y optimizando el consumo de papel a favor de una conciencia ambiental.

En el primer capítulo se dará un marco teórico de referencia para entender algunos de los conceptos más importantes de cómo se fue llevando a cabo el diseño del sistema SIICOFE.

En el segundo capítulo se llevará a cabo el análisis del sistema en base a información que se tiene del mismo. En el capítulo tres en base al análisis del capítulo 2 se desarrollará el sistema SIICOFE. Y ya por último en el capítulo 4 se realizarán unas pruebas de carga de información y pruebas del sistema antes de iniciar la operación del sistema.

Iniciemos como se llevó a cabo el proceso de elaboración del sistema SIICOFE. Y entenderemos algunos conceptos básicos que se deben de conocer sobre las Tecnologías de Información.

Marco teórico.

Para poder iniciar en los sistemas de información es necesario tener algunos conceptos importantes. Ya que serán de gran apoyo para entender el desarrollo del sistema, y aprender a conocer el desarrollo de un sistema estructurado y fundamentado en la documentación del mismo.

Muchos de los sistemas no se desarrollan como debe de ser. En la actualidad primero se desarrolla y después se documenta. Aquí veremos algunas metodologías para el desarrollo de un buen sistema de información y la aplicación de herramientas para un buen uso y operación.

El objetivo de un Sistema de Información es el de examinar un problema y generar una solución.

1.1. Introducción a los sistemas de información.

El enfoque que le demos a un sistema de información (SI), va a depender de lo que se quiera solucionar. Como sabemos los sistemas de información es un conjunto de datos almacenados de tal manera que nos proporciona un fin específico, es decir que cuando hablemos de los sistemas de información para la seguridad hablaremos de toda la gama de información que puede ser empleada para este fin, es decir bases de datos son: criminales, fotografías, medios electrónicos y electromagnéticos que en su conjunto nos van a proporcionar información para llegar a un fin específico. Otro ejemplo puede ser el sistema de un banco el cual contiene la información de los usuarios su dinero disponible y sus movimientos. Toda la información que conlleva a una tarea específica se le denomina Sistema de Información.

Pero ¿Qué es un sistema?. Lo definiremos como un conjunto de elementos que tienen entre sí un logro determinado. Para nuestro propósito es necesario relacionarlo con las Tecnologías de Información basada en sistemas Informáticos.

El sistema de información es el conjunto de elementos, datos y procedimientos coordinados e interrelacionados que tiene como propósito para el manejo de la información y para la obtención de un fin específico. Este resultado permitirá tomar decisiones adecuadas a nuestro propósito que se tenga en mente.

Generalmente nos preguntamos el por qué conocer estos términos. Esto es con el fin de desarrollar una aplicación que esté bien sustentada dada nuestra necesidad, es decir que desarrollaremos un sistema con el fin de que se lleve a cabo de la mejor manera y que sea confiable, tanto para el usuario como para el administrador del sistema.

1.2. El enfoque de las Tecnologías de Información.

¿Qué son las Tecnologías de Información? y ¿en dónde las podemos aplicar?. Las tecnologías de información permiten de una manera clara y objetiva reconocer patrones de información cuyo objetivo es el de dar a conocer de una manera clara y concisa lo que queremos expresar o especificar. Como sabemos en el mundo actual la información nos ayuda a conocer información de todo el mundo. Con el internet y las tecnologías de información podemos encontrar todo tipo de información. Aquí se debe de tener un poco de cuidado. Ya que mucha de la información que está en internet puede no ser cierta o es muy ambigua.

1.3. La administración de los sistemas de información.

Como he mencionado anteriormente en el punto 1, la finalidad de los sistemas de información es satisfacer una necesidad. El objetivo es solucionar un problema propuesto, ya que de los objetivos del sistema se extraen partes importantes para el análisis y desarrollo de los sistemas de información.

Debemos de considerar puntos importantes para llevar a cabo un buen sistema, es decir debemos de plantear el objetivo y sus precedentes. Debemos de considerar algunos puntos importantes en el desarrollo del sistema. Imaginemos por un instante que vamos a vender nuestro producto y que necesitamos dar un costo aproximado del sistema. Para eso necesitamos apoyarnos en la administración de los sistemas de información ya que ellos serán nuestra herramienta para poder dar un valor de nuestro trabajo. Necesitamos ver los requerimientos es decir, objetivos a corto y a largo plazo esto depende de nuestro sistema. Debemos de considerar el hardware con qué se cuenta y cuánto están dispuestos a solventar económicamente para mejorar sus sistemas, en dónde se va a instalar, el software qué se va a utilizar, qué personal vamos a necesitar para el desarrollo del sistema por un lado y por otro lado la gente capacitada que va a estar operando el sistema durante el tiempo de vida del mismo.

Cabe mencionar las cuatro fases fundamentales que se deben de considerar en un sistema de información.

- **Planeación.** Aquí se llevarán a cabo entrevistas entre los involucrados para ver los alcances del sistema entre otras actividades, cabe mencionar que este punto es importante para el óptimo desempeño del sistema.
- **Organización.** Ver el equipo de trabajo y la designación de las tareas de cada uno de los integrantes del equipo de trabajo.
- **Dirección.** En esta etapa se verá la designación de tareas y actividades y el flujo de la información con respecto al sistema. Aquí el líder del proyecto verá los avances de cada uno de los integrantes del equipo de trabajo.
- **Control.** Ver el desarrollo del proyecto para ver tiempos y movimientos que se estén cumpliendo en base a metas programadas por el líder del proyecto.



Fig. 1.1 Fases generales de un Sistema de Información.

1.4. Requerimientos del Sistema de Información.

Debemos de considerar lo siguiente en nuestro sistema:

- 1.4.1. **Hardware.** Aquí veremos la parte física de nuestro cliente, es decir ver con qué equipos cuenta en la actualidad y ver las posibilidades de crecimiento si es posible mejorar las condiciones del equipo o si el sistema va a correr en un servidor. Uno de los principales problemas que ocurren cuando uno va a analizar el hardware es que las compañías en México generalmente no quieren invertir en nuevas tecnologías por el costo que esto implica.
- 1.4.2. **Software.** En esta parte exploraremos los sistemas lógicos con que cuenta nuestro cliente. Como es el Sistema Operativo, el o los programas que utilizará para el manejo de las herramientas que se le van a implementar y la posibilidad de crecimiento en caso de ser necesario. Como sabemos podemos enfocarnos en varias partes del software que puede ser:
 - 1.4.2.1. **Software del sistema.** Son aquellos que dirigen al equipo, como pueden ser los sistemas operativos, compiladores, lenguajes de programación, etc.
 - 1.4.2.2. **Software de aplicación.** Son aquellos diseñados para el usuario, estos se interrelacionan con el software del sistema y pueden ser de uso general o específico.
- 1.4.3. **Recursos Humanos.** Son las personas involucradas con el sistema este personal tendrán que interactuar entre sí para lograr el objetivo del sistema de información como son los analistas, programadores, diseñadores, líder de proyecto, etc. Cabe señalar que si no tenemos los recursos humanos generalmente tenemos que llevar a cabo.
- 1.4.4. **Datos.** Es la parte mínima de la información que será procesada en base al manejo de lo que queremos definir. Los datos son ese conjunto de información que nos van a permitir manipular las bases de datos de nuestro sistema para poder operar funcionalmente.
- 1.4.5. **Recursos financieros.** Aunque muchos no lo consideran un punto importante a tratar el recurso financiero es imprescindible para el desarrollo del sistema, ya que dependiendo de los requerimientos que se tengan podremos dar un costo del sistema. En el sector público hay sistemas que no tienen un costo físico real, sin embargo si realizamos el análisis ya va implícito un costo solo por el hecho de considerar la elaboración del mismo y la implementación con los recursos con los que se cuentan. Debemos de considerar el sitio donde se va a alojar el sistema, costos de alojamiento, costo de cada persona involucrada y software que se va a emplear para llevar a cabo la programación. Una de las ventajas de java es gratuito y MYSQL (Base de datos). Esto permite economizar si es que no se tienen suficientes recursos financieros. Debemos de considerar todos los gastos que involucra el desarrollo del sistema por mínimo que esté sea. Y aunque se oiga mal la siguiente expresión se debe de cobrar todo y obviamente debemos de tener una ganancia, producto de el desarrollo del sistema que se va a implementar.

1.5. Ciclo de vida de un sistema de información.

Es aquí en esta parte en donde un sistema de información lleva un proceso determinado por las necesidades de nuestro proyecto.

Los pasos a seguir para nuestro proyecto serán:

- 1.5.1. **Investigación preliminar o planificación.** Es el estudio de las necesidades del sistema a trabajar. En el siguiente capítulo 2 se detallara esta parte.
- 1.5.2. **Análisis o determinación de requerimientos.** Aquí el analista detallará las necesidades, requerimientos del sistema.
- 1.5.3. **Diseño.** En el diseño veremos dos partes importantes que son:
 - **Diseño del prototipo.** Aquí se elaborará una primera etapa del proyecto para ver el costo, duración de cada una de las fases del proyecto, el establecer con el cliente la documentación necesaria y requerida para el control del sistema. Y en esta etapa se definirá la fecha de entrega del proyecto.
 - **Diseño del sistema.** Aquí se elaboran los módulos que componen el sistema, las partes que lo integran como son pantallas, procesos necesarios y la lógica detallada de todo el sistema.
- 1.5.4. **Implementación.** En esta parte del proyecto se instalará el proyecto en un medio simulado para ver el funcionamiento del sistema y se evaluará para localizar posibles errores de sistema.
- 1.5.5. **Pruebas o Puesta a punto.** En esta parte el sistema ya se encuentra aprobado por el cliente. Una vez en operación se procede a verificar cualquier tipo de error.
- 1.5.6. **Mantenimiento.** En esta parte el sistema se le dará mantenimiento, es decir aquí no se implementará otra parte, si no que se localizarán problemas del mismo sistema. Si se requiere otra implementación para mejorar el sistema se procederá a iniciar nuevamente el ciclo del desarrollo de un sistema de información.

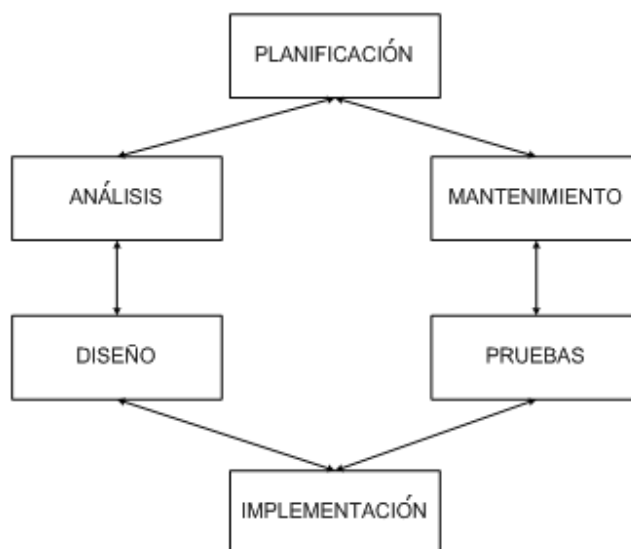


Fig. 1.2 Ciclo de vida de un Sistema de Información (SI).

1.6. Metodología de análisis y diseño de sistemas.

Como sabemos las metodologías del análisis de los sistemas se han mejorado en el desarrollo de los sistemas. Anteriormente las metodologías eran básicas y elementales. Algunos modelos eran el Modelo de datos y entidad relación. En la actualidad se manejan distintas técnicas para el análisis y diseño de sistemas. Veremos a continuación algunos modelos importantes que se utilizan en la actualidad.

1.6.1. Modelo Vista Controlador.

Este modelo es utilizado por lenguajes de programación como es Java y .NET. ¿Qué es el Modelo Vista Controlador?, no es más que un patrón utilizado en la arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control. Este patrón es muy utilizado en aplicaciones vía Web. El modelo Vista controlador (MVC) permite en sus tres etapas por un lado se muestra el HTML esta representa la vista, es decir donde se muestra la presentación de nuestras pantallas. Por otro lado tenemos el código de datos que mantiene dinámicamente a nuestra aplicación. (Ver figura 1.3).

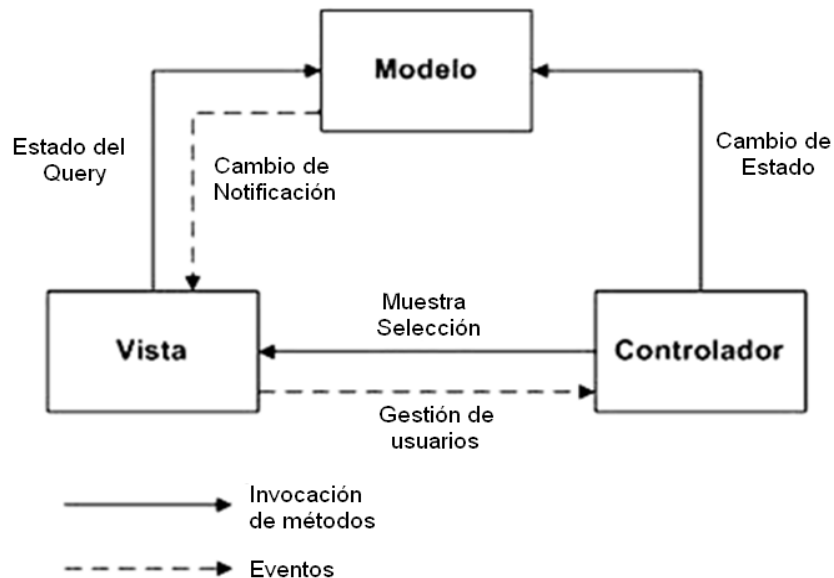


Fig. 1.3 Modelo Vista Controlador.

El modelo es nuestra Base de Datos y el controlador representa la lógica de cómo se encuentra operando nuestra aplicación. Como sabemos el modelo vista controlador es usado para desarrollar aplicaciones web usando servlets y JSPs. Este patrón es uno de los más socorridos en el modelo de J2EE (Java Enterprise Edition). Separa la presentación, la manipulación de los datos y la introducción de la información.

Modelo. Se encarga de la lógica de la aplicación y la manipulación de los datos.

Vista. Se encarga del despliegue y la actualización de los elementos que la componen.

Controlador. Se encarga de la introducción de datos a la aplicación, por lo general por parte del usuario.

1.6.2. Modelo UML.

El UML que significa (Lenguaje Unificado de Lenguaje) es una de las herramientas más utilizadas para el desarrollo de sistemas. El UML se compone de diversos diagramas con la finalidad de dar a conocer y entender una problemática específica. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que UML es un lenguaje, cuenta con reglas para combinar tales elementos. En lugar de indicarle cuales son los elementos y las reglas, vamos directamente a los diagramas ya que son más fáciles de entender y nos ayudan para hacer el análisis del sistema. La finalidad de los diagramas es el de presentar diferentes perspectivas de un sistema, a las cuales se le conoce como modelo. En el UML.

- **Diagramas de clases.** Aquí se representan cosas que tienen atributos y acciones. Por ejemplo los autos pueden ser de diferentes tipos pero tienen características similares. Y todos los autos tienen acciones, es decir todos los autos avanzan, frenan, etc. Un ejemplo de un elemento del diagrama de clase es figura 1.4:

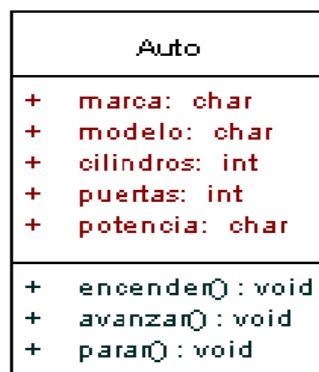


Fig. 1.4 Diagramas de Clase de un auto.

- **Diagramas de caso de uso.** Los diagramas de casos de uso es una descripción de las acciones del sistema desde el punto de vista del usuario. Para los desarrolladores del sistema esta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Este tipo de diagramas permite conocer el funcionamiento del sistema si es que el usuario no es un experto en esta área. Los diagramas de caso de uso mostrarán la lógica entre el usuario y/o usuarios y el sistema que se encuentra en operación.



Fig. 1.5 Diagramas de Caso de uso de reporte de un servicio.

▪ **Diagrama de estados.**

En este tipo de diagramas un objeto determinado se encuentra en un estado en particular. Es decir iniciamos nuestro diagrama con un punto inicial y vamos a desplazarnos por el flujo de nuestros objetos. Esto representa una situación de nuestro objeto. Y el comportamiento que va a tener nuestro objeto. Para ver a detalle el diagrama verificar anexo 2.



Fig. 1.6 Diagrama de estado para reportar un servicio

▪ **Diagrama de secuencia.** Los diagramas de clases y los diagramas de objetos representan información estática. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos. Diagramas de Secuencia

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario. Para mostrar la interacción con el usuario o con otro sistema se introducen en los diagramas de secuencia las *boundary classes*. En las primeras fases de diseño el propósito de introducir estas clases es capturar y documentar los requisitos de interfaz, pero no el mostrar cómo se implementa dicha interfaz.

Los diagramas de secuencia, formalmente diagramas de traza de eventos o de interacción de objetos, se utilizan con frecuencia para validar los casos de uso. Documentan el diseño desde el punto de vista de los casos de uso. Observando qué mensajes se envían a los objetos, componentes o casos de uso y viendo a grosso modo cuanto tiempo consume el método invocado, los diagramas de secuencia nos ayudan a comprender los cuellos de botella potenciales, para así poder eliminarlos. A la hora de documentar un diagrama de secuencia resulta importante mantener los enlaces de los mensajes a los métodos apropiados del diagrama de clases. Ver anexo 2.

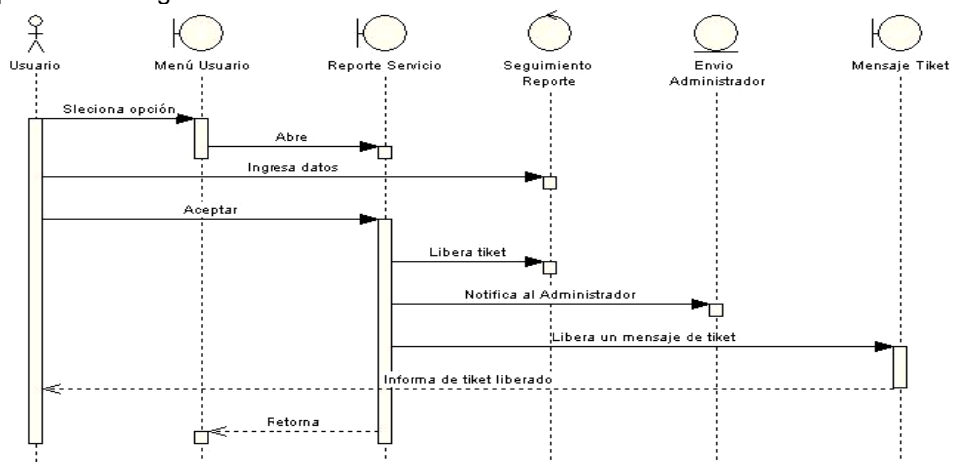


Fig. 1.7 Diagrama de secuencia para registrar un servicio.

- Diagrama de actividades.** En un diagrama de actividades se representan los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

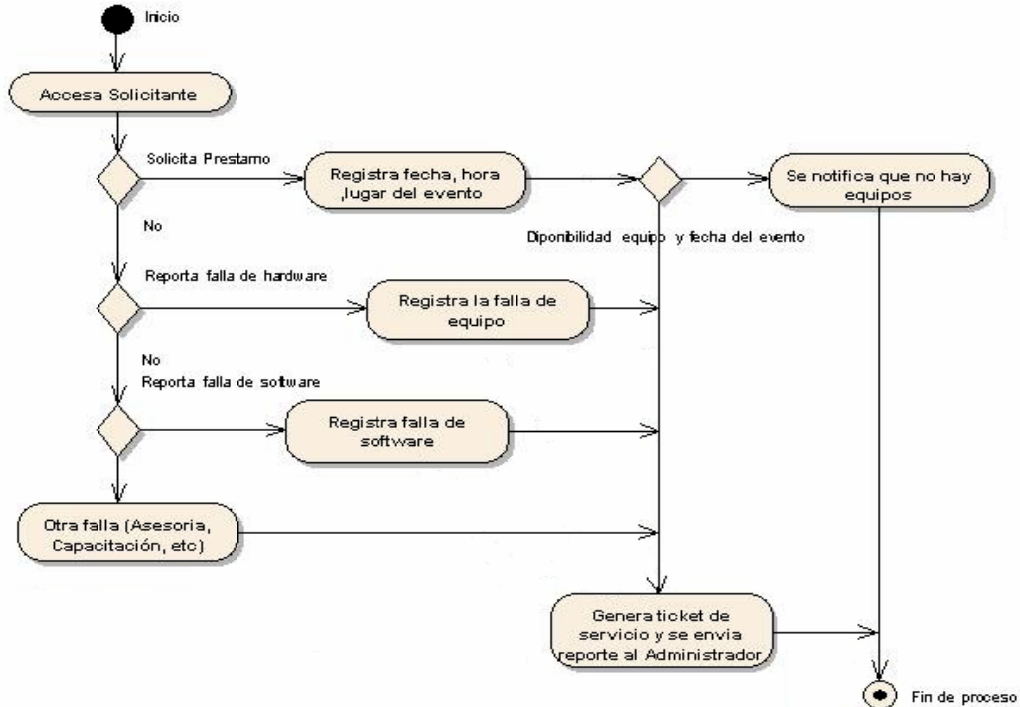


Fig. 1.8 Diagrama de actividades reporte de servicio.

- Diagrama de colaboración.** Es una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas en torno a los objetos y los enlaces entre ellos.

Los diagramas de colaboración proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros. Este tipo de diagramas se utilizan más frecuentemente en la fase de diseño, es decir, cuando estamos diseñando la implementación de las relaciones.

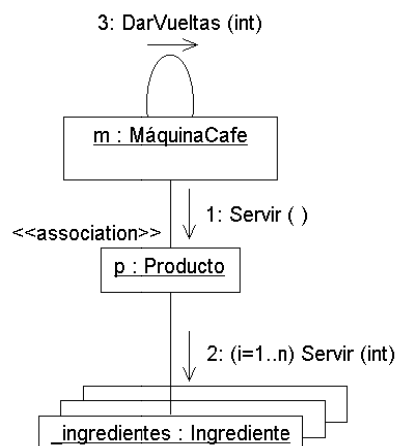


Fig. 1.9 Diagrama de colaboración de una máquina de café.

- **Diagrama de componentes.** Este tipo de diagramas muestra las organizaciones y dependencias lógicas entre componentes *software*, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue. Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida

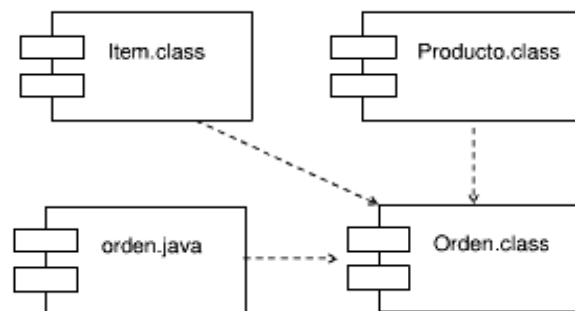


Fig. 1.10 Diagrama de componentes en java.

- **Diagrama de distribución.** Muestra la arquitectura física de un sistema de información. Puede representar los equipos y dispositivos, mostrar sus interconexiones y el software que se encontrará en cada computadora. Cada computadora está representada por un cubo y las interacciones entre las computadoras están representadas por líneas.

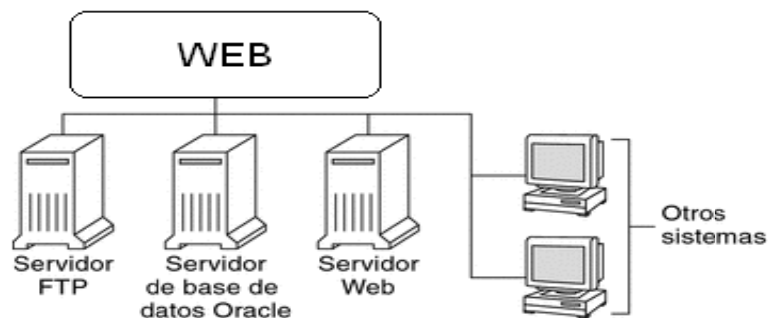


Fig. 1.11 Diagrama de distribución.

1.6.3. Modelo RUP.

El Rational Unified Process™ (RUP) es un proceso de ingeniería de software que mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo. Los contenidos específicos para e-business del RUP™ proporciona una guía específica en áreas tales como la de Modelado de Negocios, Arquitecturas Web, Pruebas y Calidad. También proporcionan lineamientos para desarrollar en plataformas IBM™ Websphere y Microsoft™ Solution con el fin de acelerar nuestros proyectos de desarrollo Web. RUP™ está fuertemente integrado con las diferentes herramientas Rational™, permitiéndole a los equipos de desarrollo de ne Digital alcanzar todos los beneficios de las características de los productos Rational™, el Unified Modeling Language (UML), y otras mejores prácticas de la industria.

1.6.4. Modelo de Madurez de Capacidad para el Software (CMM).

Este modelo proporciona a las organizaciones de software una orientación sobre cómo obtener el control de sus procesos de desarrollo y mantenimiento de software, y cómo evolucionar hacia una cultura de ingeniería del software y de gestión por excelencia.

Para obtener éxito en nuevos mercados internacionales, ne Digital ha emprendido nuevos retos tecnológicos tales como la conformación de un equipo de trabajo encargado de cumplir con los cinco niveles exigidos por la norma (inicial; repetición de experiencias exitosas; definición de procesos estándar; gestión efectiva; y optimización).

El principal objetivo será implementar el Modelo CMM para crear un enfoque hacia los procesos, basados en estándares que permitan mejorar continuamente nuestra calidad y la satisfacción de nuestros clientes ante el Software Engineering Institute™ (SEI), organización encargada de desarrollar y revisar el modelo CMM.

Beneficios de la aplicación del Modelo CMM en ne Digital

- Establecimiento de las bases para el mejoramiento continuo, al definir un plan de trabajo enfocado hacia las áreas clave de los procesos.
- Mejora de la productividad en el desarrollo de software al reducir la cantidad de trabajo, detectando errores en las etapas iniciales del desarrollo y mejorando el presupuesto de los proyectos.
- Estandarización de las prácticas establecidas en cuanto a desarrollo y mantenimiento del software.
- Mejoramiento de la calidad y satisfacción del cliente, mediante una mejor administración de los requerimientos e identificación temprana de errores.
- Mejor cumplimiento de los tiempos de entrega estimados a través de una rigurosa planificación.
- Mejores condiciones de trabajo al incrementar el espíritu de trabajo en equipo, el compromiso, la disminución del estrés y las horas extras.

1.7. Reglas de integridad.

Una vez definida la estructura de datos del modelo relacional, pasamos a estudiar las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos. Al definir cada atributo sobre un dominio se impone una restricción sobre el conjunto de valores permitidos para cada atributo. A este tipo de restricciones se les denomina restricciones de dominios. Hay además dos reglas de integridad muy importantes que son restricciones que se deben cumplir en todas las bases de datos relacionales y en todos sus estados o instancias (las reglas se deben cumplir todo el tiempo). Estas reglas son la regla de integridad de entidades y la regla de integridad referencial.

1.7.1. Regla de integridad de entidades

La primera regla de integridad se aplica a las claves primarias de las relaciones base: ninguno de los atributos que componen la clave primaria puede ser nulo.

Por definición, una clave primaria es un identificador irreducible que se utiliza para identificar de modo único las tuplas. Que es irreducible significa que ningún subconjunto de la clave primaria sirve para identificar las tuplas de modo único. Si se permite que parte de la clave primaria sea nula, se está diciendo que no todos sus atributos son necesarios para distinguir las tuplas. Nótese que esta regla sólo se aplica a las relaciones base y a las claves primarias, no a las claves alternativas.

1.7.2. Regla de integridad referencial

La segunda regla de integridad se aplica a las claves ajenas: si en una relación hay alguna clave ajena, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser completamente nulos.

La regla de integridad referencial se enmarca en términos de estados de la base de datos: indica lo que es un estado ilegal, pero no dice cómo puede evitarse. La cuestión es ¿qué hacer si estando en un estado legal, llega una petición para realizar una operación que conduce a un estado ilegal? Existen dos opciones: rechazar la operación, o bien aceptar la operación y realizar operaciones adicionales compensatorias que conduzcan a un estado legal.

Por lo tanto, para cada clave ajena de la base de datos habrá que contestar las siguientes preguntas:

- Regla de los nulos: ¿Tiene sentido que la clave ajena acepte nulos?
- Regla de borrado: ¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?
- Restringir: no se permite borrar la tupla referenciada.
- Propagar: se borra la tupla referenciada y se propaga el borrado a las tuplas que la referencian mediante la clave ajena.
- Anular: se borra la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).
- Regla de modificación: ¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?
- Restringir: no se permite modificar el valor de la clave primaria de la tupla referenciada.
- Propagar: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante la clave ajena.
- Anular: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo la clave ajena (sólo si acepta nulos).

1.8. Hacia dónde vamos con JAVA.

Al analizar las posibilidades de los lenguajes de desarrollo dentro del mercado, las tendencias actuales marcan una inclinación hacia Java. Éste tiene las ventajas de permitir una programación orientada a objetos a la vez que aporta dos valores más: el concepto de máquina virtual, que nos independiza de los sistemas servidores y su plataforma de desarrollo, que aporta partes susceptibles de ser aprovechadas (entrada/salida, red, gráficos, seguridad, etc.). Otra ventaja es que java y MYSQL son herramientas de uso libre por consiguiente no tienen un costo a comparación de los lenguajes de programación de Microsoft.

1.9. Arquitectura de las aplicaciones

La necesidad de lograr una arquitectura robusta y distribuida para dar servicios en Internet de forma flexible es una evidencia. Actualmente, para soportar este tipo de arquitecturas, el mercado nos ofrece los servidores de aplicaciones. En la actualidad la mayor parte de los servidores de aplicaciones están convergiendo hacia al modelo de Java 2 Enterprise Edition (J2EE) que soporta servlets y Enterprise Java-Beans (EJB). El modelo de componentes de EJBs, propuesto por Sun, y con el apoyo de la mayoría de vendedores de software, es ya el modelo estándar de componentes para Java, y probablemente conseguirá simplificar en gran medida la construcción de aplicaciones transaccionales empresariales.

Análisis de sistema.

2.1. Definición y planteamiento del problema.

ENUNCIADO DE TRABAJO

Objetivo General: Elaborar un Sistema Integral de Servicios para la Comisión Federal de Mejora Regulatoria (COFEMER).

Objetivos específicos:

- Elaborar un sistema de control de inventarios el cual debe de contener a todos los empleados y artículos de la COFEMER.
- Se tendrá un control de Servicios de Mantenimiento, Préstamo de equipo y cuentas de correo.
- Mejorar el sistema de Control de Inventarios para poder mover los artículos de la COFEMER entre los empleados y que los empleados puedan ascender de Puesto y por consiguiente se pueda mover el equipo.
- Emitirá estadísticas para la adquisición de equipo en futuras licitaciones.
- Comparará con el SICORE la información que se tiene actual contra el sistema actual.
- Validará los datos del SICORE en formato XML con la información del sistema.

Alcance. Este sistema pretende optimizar los tiempos de operación de una solicitud y validar en automático la información de inventarios tanto de COFEMER como del SIICORE. Además de apoyar al área administrativa a tener un buen control del área de Inventarios y de personal de Recursos Humanos.

El control de inventarios debe de contener:

Acceso restringido a usuarios con niveles de acceso

1) Administrador.

- El Administrador se encargará de controlar las cuentas de correo (alta, baja y modificación).
- Dará seguimiento al préstamo de equipo de computo, cuando un usuario (solicitante) solicita un servicio el sistema le enviará una notificación al administrador del servicio que se está solicitando. Y una vez concluido el servicio se cierra el ticket del servicio y se almacenará en el histórico para las estadísticas mensuales y anuales de los servicios realizados.
- Dará seguimiento a los servicios de mantenimiento que los usuarios soliciten, aquí se dará atención a los usuarios en general, pero si llega a ocurrir un incidente mayor será necesario enviar un reporte al sistema CAU (Centro de Atención de Usuarios de la Secretaría de Economía) para su atención. En esta parte el Administrador atenderá servicios como son:
 - Fallas de hardware.
 - Fallas de software.
- Asesoría y capacitación. En esta parte se le dará atención al usuario para solucionar su problema en cuestiones elementales de operación del equipo.
- El Administrador podrá dar de alta, baja o modificación de los usuarios de la COFEMER.
- El Administrador podrá llevar un control del inventario de la COFEMER. Alta de artículos, baja o modificación. Y si son dados de baja mencionará a que unidad administrativa fue trasladada, dónde se detallará el traslado o la baja en base a los formatos de baja o traslado.

- 2) Auxiliar.
 - El auxiliar será capaz de dar de alta usuarios, baja o modificación de los usuarios de la COFEMER.
- 3) Solicitante.
 - El solicitante será capaz de ingresar una solicitud de servicio de una falla de software, de hardware o préstamo de equipo.

El nivel de usuario permitirá tener acceso a determinadas partes del sistema.

MENUS DEL SISTEMA

- a. Menú **CAPTURA DE USUARIOS**. En esta parte se darán de alta los usuarios. Pero solo lo podrán realizar quienes tengan los permisos correspondientes.
 - a) Alta de usuarios.
 - b) Baja de usuarios.
 - c) Modificación de usuarios.
 - d) Histórico de usuarios.
- b. Menú **CAPTURA DE BIENES**. Aquí se registran los artículos, para que se lleve un control de los mismos. Los formularios deben de ser validados para minimizar la captura errónea de información
 - a) Alta de artículos.
 - b) Baja de artículos.
 - c) Modificación de artículos.
 - d) Bienes dados de baja.
- c. Menú **REPORTES (IMPRESO Y EN PANTALLA)**. Aquí se darán a conocer las cantidades de artículos y de los usuarios que están activos y/o inactivos así como los bienes que cuenta cada uno de los usuarios de la COFEMER. Permitirá imprimir los vales de Resguardo de cada usuario. Y el histórico tanto de los artículos como de los usuarios.
 - a) Artículos.
 - b) Usuarios.
 - c) Histórico de artículos y de usuarios.
 - d) Vale de resguardo.
- d. Menú **MOVIMIENTOS**. En esta parte se llevan a cabo movimientos de usuarios por grupo, es decir un usuario puede tener varios artículos y pueden pasar a otro usuarios. Al igual los Usuarios pueden subir de puesto y tener los objetos del usuario anterior. También se cuenta con el histórico de los usuarios y artículos.
 - a) Artículos a usuarios.
 - b) Usuarios (Nivel de puesto).
 - c) Antecedente Histórico de Movimientos (Usuarios y artículos).
- e. Menú **ESTADÍSTICAS**. De artículos en buen estado, obsoletos y otros conceptos para poder medir que se va a adquirir de bienes materiales.
 - a) General.
 - b) Personalizado.
- f. Menú **COMPRACION – SICORE**. Permitirá comparar los artículos de nuestro sistema con el SICORE de la Secretaria de Economía.
 - a) XML COMPARE.
- g. Menú de **AYUDA**. Explicará de manera clara cada una de las partes del sistema.

Ventajas del sistema.

- Con la implementación del sistema se optimizará el tiempo de respuesta de los reportes. Se verificará si se encuentra actualizado el inventario de la COFEMER para verificarlo con otro sistema ya implementado llamado SICORE para verificar la información entre los dos sistemas.
- Con ayuda de las estadísticas sería posible verificar el status para nuevas adquisiciones y para dar de baja equipo.
- Se tendrá un control de los reportes emitidos y los pendientes.
- Se tendrán al día los inventarios de artículos de la COFEMER y los usuarios.

Lo más importante antes de comenzar cualquier análisis y/o desarrollo, es establecer los requerimientos de nuestro sistema.

Análisis y Diseño.

La mayoría de proyectos de software son complejos, y la estrategia primaria para superar la complejidad, es el subdividir nuestro sistema. Un enfoque tradicional para realizar esto fue el análisis y diseño estructurados, donde se trata de descomponer el problema en funciones o procesos. Este método origina una división jerárquica de procesos constituidos por sub-procesos.

Otra forma de realizar la descomposición, es usando un esquema de análisis y diseño orientado a objetos. En este esquema, se busca descomponer el problema en objetos, y no en funciones. Algunas de las tareas a realizarse en la etapa de análisis son las siguientes:

- Definir los requerimientos.
- Definir los casos esenciales de uso.
- Crear y perfeccionar los diagramas de casos de uso.
- Crear y perfeccionar el modelo conceptual.
- Crear y perfeccionar el glosario.
- Definir los diagramas de secuencia de los sistemas.
- Definir los contratos de operaciones.

Algunas de las tareas a realizarse en la etapa de diseño son las siguientes:

- Definir los casos reales de uso.
- Definir los reportes, la interfaz de usuario y la secuencia de las pantallas.
- Perfeccionar la arquitectura del sistema.
- Definir los diagramas de interacción.
- Definir los diagramas de diseño de clases.
- Definir el esquema de la base de datos.

Suposiciones y Restricciones

- Se realizará el Sistema en cualquier lenguaje para WEB.
- La base podrá funcionar en SQL Server.
- Se implantará en un Servidor Dell 4400
- Las pantallas serán en base al decreto de la Secretaría de Gobernación.
- Deberá de tener los escudos de la SE y COFEMER.
- Las computadoras deberán de contar con una conexión a INTERNET.

Los usuarios del sistema deben de tener los requerimientos mínimos de uso y operación de una computadora y el manejo de Internet.

Debemos de considerar lo siguiente en nuestro análisis.

Suposiciones y Restricciones del producto		
Tipo	Restricción	Descripción
Tecnológica	Distribución	El sistema debe trabajar en red interna www.COFEMER.gob.mx/intranet
Económica	Costos adicionales	Los costos adicionales de la compra del software como son: capacitación, contratación de consultores, cambios en los procesos internos.
Tecnológica	Seguridad	El software adquirido debe cumplir con las políticas de seguridad de la organización.

Tabla 1. Consideraciones del análisis.

2.2. Definición de recursos. En esta etapa analizaremos los requerimientos tanto de personal como de equipo necesario para poder implementar el sistema SIICOFE.

2.2.1. Recursos humanos.

En este sistema será necesario el siguiente personal:

- Líder de proyecto. El se encargará de dirigir y dar seguimiento al proyecto además de establecer una comunicación continua con nuestro cliente.
- Analista. Se encargará de evaluar el diseño del sistema y se encargará de analizar el diseño del sistema.
- Programador. Se requerirá por lo menos de un programador este se encargará de la programación de nuestro sistema en base al análisis del sistema solicitado.
- Diseñador. Se encargará de darle presentación a nuestro sistema. Este diseñador se enfocará en la apariencia del sistema.

El siguiente personal se encarga de la operación del sistema. Se requiere de un Administrador, se requieren por lo menos 2 auxiliares y “n” usuarios. En este caso el mercado de trabajo son los usuarios de la COFEMER.

2.2.2. Recursos de hardware. Dos servidores. Uno dónde se aloje la Base de datos y el otro en dónde se aloje nuestra página Web.

2.2.3. Recursos de software. Se requiere de la licencia de SQL Server, del Oracle JDveloper y Photoshop CSS.

2.2.4. Diagrama de actividades (GANT). Aquí se mostrará los tiempos estimados para la elaboración del proyecto.

Como se menciona en suposiciones y restricciones había mencionado las especificaciones de los recursos de hardware y de software a ser utilizado por el sistema. Es importante considerar el hosting. Es decir el lugar dónde se va a alojar la página a desarrollar.

2.3. Los requerimientos de un sistema:

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

Los requerimientos necesitan ser:

- Los adecuados al sistema.
- Necesitan estar más enfocados al usuario.
- Son los primeros en someterse a un proceso de reingeniería.

2.3.1. Requerimientos funcionales.

- El Administrador se encargará de controlar las cuentas de correo.
- Dará seguimiento al préstamo de equipo de cómputo.
- Dará seguimiento a los servicios de mantenimiento.
 - Fallas de hardware.
 - Fallas de software.
- Asesoría y capacitación.
- El Administrador y el Auxiliar podrán registrar a los usuarios.
- El Administrador y el Auxiliar podrán llevar un control del inventario de la COFEMER.
- El Administrador y el Auxiliar podrá mostrar las cantidades de artículos y de los usuarios que se encuentran activos y/o inactivos así como los bienes con que cuenta cada uno de los usuarios de la COFEMER.
- El Administrador y el Auxiliar podrá cambiar los artículos en buen estado, ya obsoletos y otros conceptos para poder medir que se va a adquirir de bienes.
- El sistema permitirá comparar los artículos de nuestro sistema con el SICORE de la Secretaría de Economía

2.3.2. Definición de requerimientos no funcionales.

- Se realizará el Sistema en cualquier lenguaje para WEB.
- La base podrá funcionar en SQL Server.
- Se implantará en un Servidor Dell 4400.
- Las pantallas serán en base al decreto de la Secretaría de Gobernación.
- Deberá de tener los escudos de la SE y COFEMER.
- Las computadoras deberán de contar con una conexión a INTERNET.
- Los usuarios del sistema deben de tener los requerimientos mínimos de uso y operación de una computadora y el manejo de Internet.

2.4. El UML para el análisis del sistema.

Clases, objetos y algo más.

Para poder entender un poco la parte de elaboración de nuestro sistema es necesario conocer algunos conceptos importantes (Ver Apéndice 1).

En el campo de las tecnologías de la información, surgió inicialmente como un enfoque para la programación de aplicaciones, pero se ha extendido a todo el ciclo de desarrollo de sistemas como:

Programación Orientada a Objetos	Método de implementación en el que los programas se organizan como colecciones cooperativas de objetos
Diseño Orientado a Objetos	Método de diseño que abarca el proceso de descomposición OO y una notación para describir los modelos lógico, físico, estático y dinámico del sistema
Análisis Orientado a Objetos	Método de análisis que examina los requisitos desde la perspectiva de los objetos que se encuentran en el dominio del problema.

Tabla 2. Desarrollo de sistemas.

Cada una de las metodologías permite optimizar los SI (Sistemas de Información) para obtener mejores resultados en nuestras aplicaciones. Esto permitirá ir mejorando e implementado sistemas más objetivos y con un alto desempeño, que es lo que buscan los sistemas actuales. Siempre responder lo más rápidamente posible y obtener resultados exactos en corto plazo.

El UML es una metodología que permite llevar a cabo un análisis a detalle y que permitirá visualizar sin aun llegar al desarrollo de nuestra aplicación y nos permite ver las necesidades que va a requerir el sistema. Esta metodología permite organizar y manipular la información requerida en nuestro sistema a desarrollar. Y permitirá en corto plazo informar a nuestro cliente de los requerimientos y necesidades que se van presentando en el sistema.

2.5. Diagramas de casos de uso del sistema SIICOFE.

En nuestros diagramas se mostrará cada uno de los elementos que involucran a nuestro sistema. Por un lado tenemos de nuestro caso de uso general y por otro lado el Modelo estático, conocidos como diagramas estructurales. En dónde se verán los diagramas de clases, diagramas de componentes y de despliegue. Y por otro lado tenemos el modelo Dinámico este diagrama muestra el comportamiento de nuestro sistema con los respectivos diagramas de comportamiento. Aquí se verán los diagramas de Secuencia, Colaboración, de estados y de Actividad.

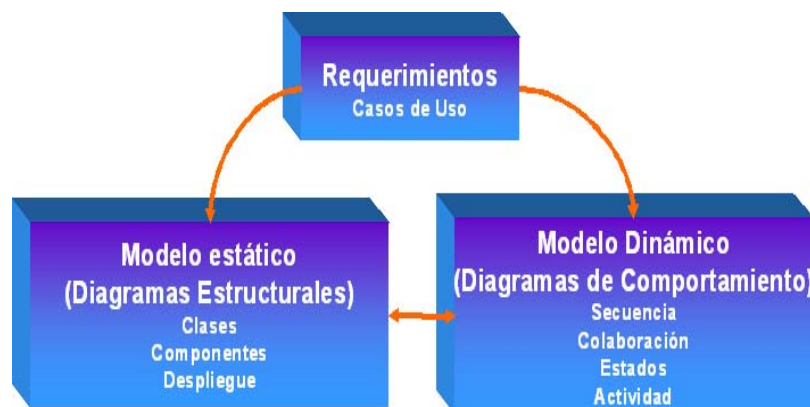


Figura 2.1 Diagrama de cómo UML define los modelos.

Para llevar a cabo un análisis que nos permita desarrollar nuestra aplicación lo enfocaremos a la técnica de UML. El UML permite llevar a cabo un análisis basado en clases y objetos en dónde se permitirá ver a detalle cada una de las técnicas desarrolladas en el proceso de elaboración de los Sistemas de Información. Lo que permitirá documentar de una manera clara y objetiva para el proceso de aprobación por parte del cliente y ver que requerimientos son los solicitados y cuál es el alcance del proyecto mismo. Sin perder de vista cuales van a ser los costos que esto implica así como a futuro el mantenimiento y soporte que el sistema va a requerir.

Los diagramas deben de ser claros para el cliente y deberán de estar enfocados a los objetivos específicos para el desarrollo.

2.5.1. Casos de Uso.

Aquí se muestra la lógica del caso de uso de nuestro sistema.

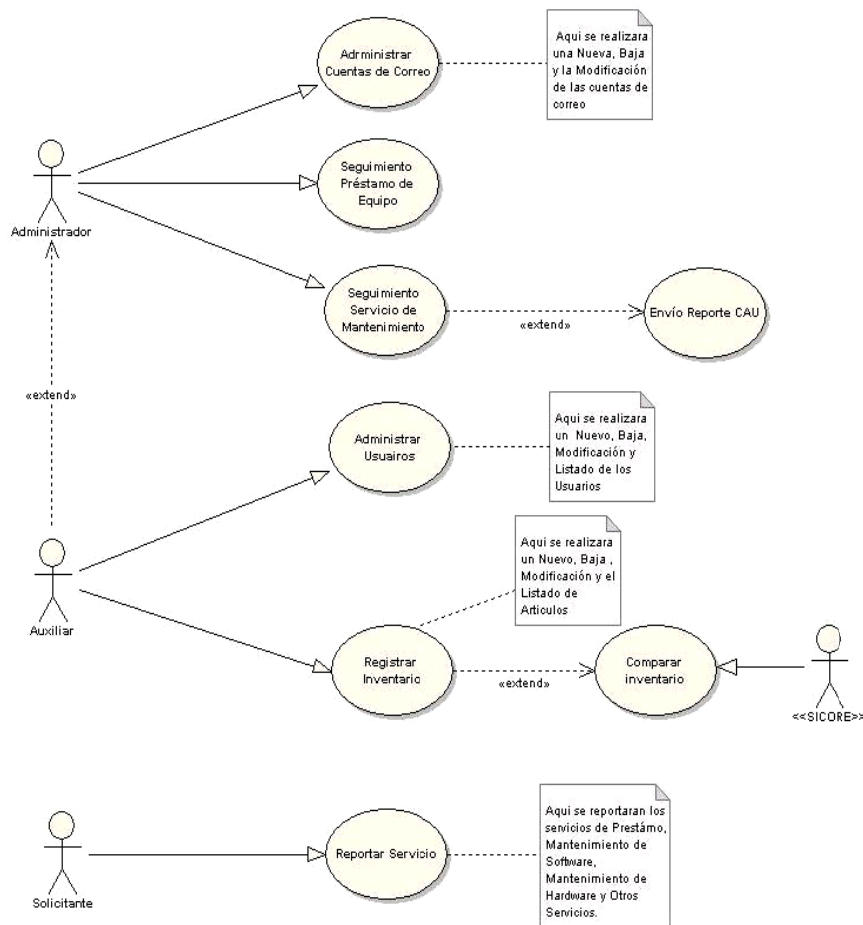


Figura 2.2 Diagrama de caso de uso sistema SICOFE.

Para comprender un poco nuestro sistema nos enfocaremos a las funciones que desempeña cada uno de nuestros actores como son el Solicitante, el Auxiliar y el Administrador en nuestro diagrama de caso de usos (UML). Ver anexo 2.

El solicitante para nuestro sistema solo va a requerir reportar sus servicios a el área de soporte técnico para poder llevar a cabo alguna reparación ya sea falla de hardware o de software o simplemente una asesoría sin embargo se debe de registrar cada uno de los movimientos que se lleven a cabo en nuestro sistema asignándole un ticket de servicio para ser atendido. Como sabemos los servicios solicitados en el área deben de ser atendidos en un plazo no mayor a 24 hrs. de lo contrario se activan puntos en contra por los servicios no concluidos. Esto nos servirá de referencia en el sistema para llevar a cabo un control de servicios atendidos contra servicios no atendidos para futuras evaluaciones.

El auxiliar está representado en la COFEMER (Comisión Federal de Mejora Regulatoria) por el área Administrativa. Esta área se va a encargar de capturar al personal de esta institución para el registro del personal el cual debe de contar con los datos del empleado a dar de alta. Una de las ventajas del sistema es que el sistema va a permitir mostrar el estatus del personal y verificará al momento de su captura que los empleados tengan todos los papeles en regla, así como verificará el estatus de servicios solicitados (credencial de trabajador, correo electrónico, equipo de cómputo asignado, mobiliario asignado, etc.).

El área Administrativa tendrá el control de inventarios en general y de usuarios de la COFEMER. Actualmente no se tiene un control detallado del control tanto de usuarios como de mobiliario y equipo. Si requieren un informe no se tiene a la mano. Por lo que es necesario contar con una herramienta que nos proporcione esta información en poco tiempo. Cabe mencionar que el sistema permitirá comparar con otro sistema llamado SIICORE. Este sistema es de la Secretaria de Economía para el control de inventario de activo fijo sin embargo actualmente de tiene equipo en arrendamiento y este equipo no forma parte del activo fijo de la COFEMER. Este sistema permitirá tener un control de todo el inventario con que se cuenta en la COFEMER. El auxiliar también podrá modificar el inventario en caso de reasignación del mismo o la baja tanto de usuarios como de inventario en general.

El Administrador podrá tener un control de todo lo anterior, pero la principal función será la de dar de alta, baja y modificación de los servicios de asignación de equipos, activación de cuentas de correo, de elaboración de credenciales y registro (alta, baja o modificación de datos) de la huella digital para el control de Asistencia. Y también tendrá el control del seguimiento en los servicios que requieran los usuarios de la COFEMER. El Administrador tiene la capacidad de poder emitir reportes de los servicios solicitados.

Para entender un poco como se detalla cada una de las partes que nos servirán para tener una idea de las funciones que nuestro sistema tiene explicare una de estas partes llamada Narrativas.

Las Narrativas permitirán ver como cada uno de los actores y como interactúa en cada una de las acciones que lleva a cabo.

NARRATIVAS

CASOS DE USO DE ALTO NIVEL SIICOFE

CASO DE USO	1.- Administrar Cuenta de Correo
ACTOR	Administrador
TIPO	PRIMARIO, ESCENCIAL
DESCRIPCION	El Administrador es el único que puede Crear la cuenta darla de Baja o Modificarla.

Tabla 3. Caso de cuenta de correo.

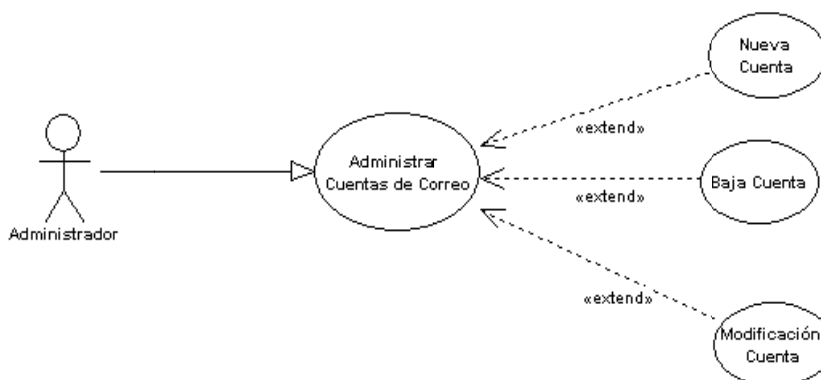


Figura 2.3 Narrativa de una cuenta de correo sistema SICOFE.

Precondiciones.

- El Administrador debe de ingresar al SIICOFE.
- Seleccionar Administrar Cuenta de Correo.

Como podemos observar en la figura 2.3 se da un panorama general una de las actividades que realiza el Administrador. Cuyo objetivo es el de dar de Alta, Baja o Modificación de una cuenta de correo.

Caso de uso. Crear una Nueva Cuenta de Correo.



Figura 2.4 Diagrama del caso de uso de una cuenta de correo.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Selecciona la opción de Nueva Cuenta	2	El Sistema coloca a los usuarios existentes	
3	Selecciona un usuario	4	Muestra los datos del usuario que fue seleccionado	E1
5	Ingresa password, Justificación ,beneficios y observaciones y da un clic en Solicitud	6	Valida el password y envía mensaje de "Solicitud Exitosa"	E2

Tabla 4. Flujo del diagrama de cuenta de correo.

Excepciones

Id	Nombre	Acción
E1	El usuario no existe	Envía información de que el usuario no existe.
E2	Políticas del password	Que el password no contenga información de nombre o fechas de edad.

Tabla 5. Excepciones de del diagrama de cuenta de correo.

Pos condiciones:

El sistema ha registrado un nuevo usuario y se la ha asignado su cuenta de correo.

Caso de uso. Dar de Baja una Nueva Cuenta de Correo.



Figura 2.5. Diagrama de caso de uso de Baja de una cuenta.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Selecciona la opción de Baja de Cuenta	2	El Sistema coloca a los usuarios existentes	
3	Selecciona el usuario a dar de Baja	4	Muestra los datos del usuario que fue seleccionado	E1
5	Ingresa password, Justificación, Beneficios, Observaciones y Fecha de Baja y da un clic en Baja.	6	Valida el password y envía mensaje de "Baja Exitosa"	E2

Tabla 6. Flujo de la baja de una cuenta de correo.

Excepciones

Id	Nombre	Acción
E1	El usuario no existe	Envía información de que el usuario no existe.
E2	Verifica el password	Que el password no coincida con el que va a ser dado de baja.

Tabla 7. Excepciones de baja de cuenta de correo.

Pos condiciones:

El sistema ha dado de baja la cuenta de un usuario y lo manda a histórico.

Caso de uso. Realizar una Modificación de una Cuenta de Correo.

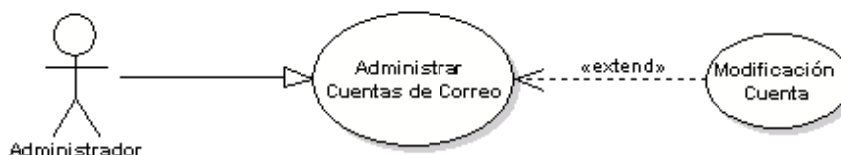


Figura 2.6. Caso de uso de modificación de cuenta de correo.

Flujo:

ACTOR		SISTEMA		
Paso	Acción	Paso	Acción	Excepción
1	Selecciona la opción de Modificar una Cuenta	2	El Sistema coloca a los usuarios existentes	
3	Selecciona el usuario a realizar la Modificación.	4	Muestra los datos del usuario que fue seleccionado	E1
5	Ingresa password, Justificación, Beneficios y/o Observaciones a ser modificado.	6	Envía mensaje de "Modificación Exitosa"	

Tabla 8. Flujo de modificación de cuenta de correo.

Excepciones

Id	Nombre	Acción
E1	El usuario no existe	Envía información de que el usuario no existe.

Tabla 9. Excepciones de modificación de cuenta de correo.

Pos condiciones:

El sistema ha Modificado la cuenta de un usuario.

Como se observa en la secuencia entre Actor y Sistema se ve como se irá comportando nuestro sistema en base a cada paso y cada acción. En la etapa de las excepciones se verá las posibles opciones que no se cumplan o tenga una acción diferente a lo requerido.

2.5.2. Diagrama de Navegación para Solicitar un Reporte.

Los diagramas de Navegación permitirán observar el flujo de nuestro sistema, explicare de manera breve una de las etapas del sistema como se lleva a cabo el acceso. En la COFEMER se cuenta con una página principal de acceso que es [www.COFEMER.gob.mx /intranet](http://www.COFEMER.gob.mx/intranet) de esta manera tenemos acceso a la intranet de COFEMER. Una vez que se ingresa a esta con el usuario y contraseña se selecciona una de las opciones la cual puede ser Reportar Servicios, como se muestra en la figura 2.7.



Figura 2.7 Diagrama de Navegación de una solicitud Sistema SIICOFE.

Una vez seleccionado el servicio se manda el Reporte al Administrador del sistema para que este atienda el reporte en un lapso no mayor a 24 hrs. El usuario recibe el tiquet de atención de cliente y regresa a la página principal. Como podemos observar aquí se da una idea general de cada uno de los módulos que tendrá nuestro sistema.

En esta etapa podremos observar cada modulo a detalle y permitirá ver la secuencia que el cliente requiere para poder vender nuestro sistema y evaluar el costo y el tiempo de ejecución de cada uno de los módulos. Sin necesidad de llegar aun al desarrollo del mismo.

2.5.3. Diagrama de secuencia para registrar un servicio.

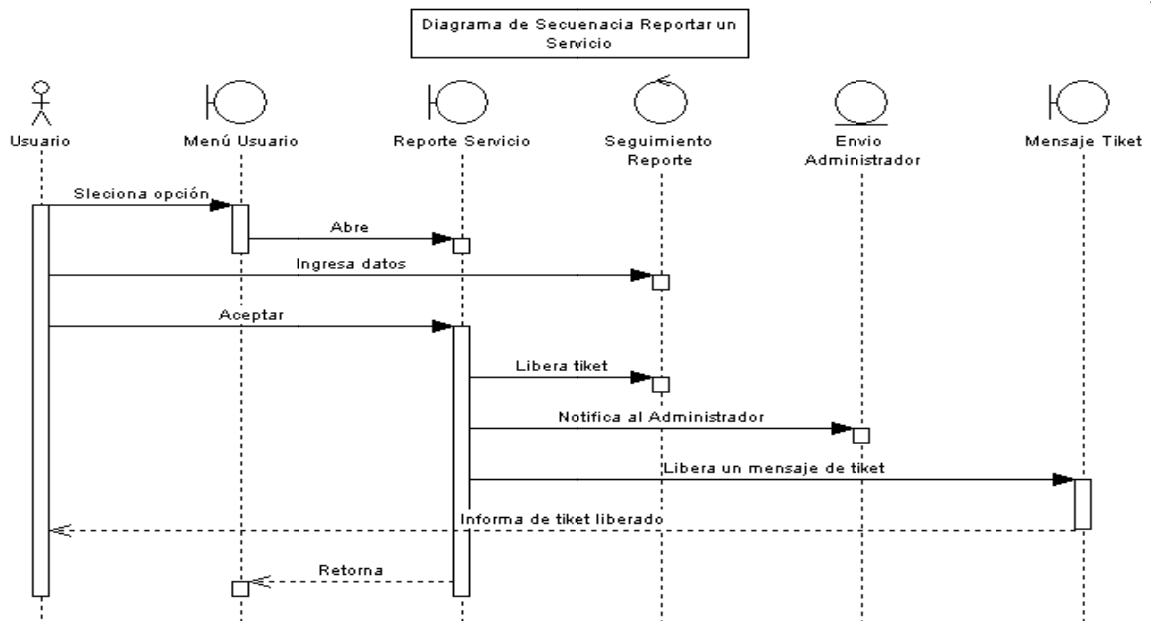





Figura 2.8 Diagrama de secuencia de reporte de servicio sistema SIICOFE.

En los diagramas de secuencias UML muestra la mecánica de la interacción con base en tiempos y movimientos del sistema, Como podemos observar en la Figura 2.8 el usuario selecciona una opción (el usuario es uno de nuestros actores representados en el diagrama de Casos de uso). Este usuario selecciona una opción que se llevara a cabo para poder levantar un servicio. El usuario ingresa sus datos y al darle la opción de aceptar libera un ticket el cual se le entregará un aviso al usuario en sí y al administrador que es quien lleve a cabo el proceso de liberación de tickets. Como podemos observar nuestro diagrama de secuencia permite ver a detalle cada uno de los pasos que se deben llevar a cabo en el sistema antes de su elaboración. Aquí se ven representados tanto actores como acciones que se llevarán a cabo por los mismos.

Aquí el símbolo  nos denota la interfaz gráfica de nuestro sistema y representa las pantallas del mismo. El siguiente símbolo representa la conexión a la base de datos  en esta parte se va a ver la conexión o conexiones que se estén utilizando. Y por último los procesos que se llevan a cabo representado por el siguiente símbolo . Por medio de esta simbología e interacción con los sentidos de las flechas y la conexión de las mismas se puede notar como el sistema se encuentra operando lógicamente.

2.5.4. Diagrama de Actividades.

El diagrama de Actividades mostrará a la tarea que va a llevar a cabo un usuario o el sistema.

Como podemos observar en nuestro diagrama se ve la lógica este es muy similar a un diagrama de flujo con la diferencia de que en este se muestra cada actividad separada por una barra. En un diagrama conceptual, una actividad es cierta tarea que debe ser llevada a cabo, ya sea por un ser humano o por una computadora. En un diagrama de perspectiva de especificación o de perspectiva de implementación, una actividad es un método sobre una clase. Ver anexo 2.

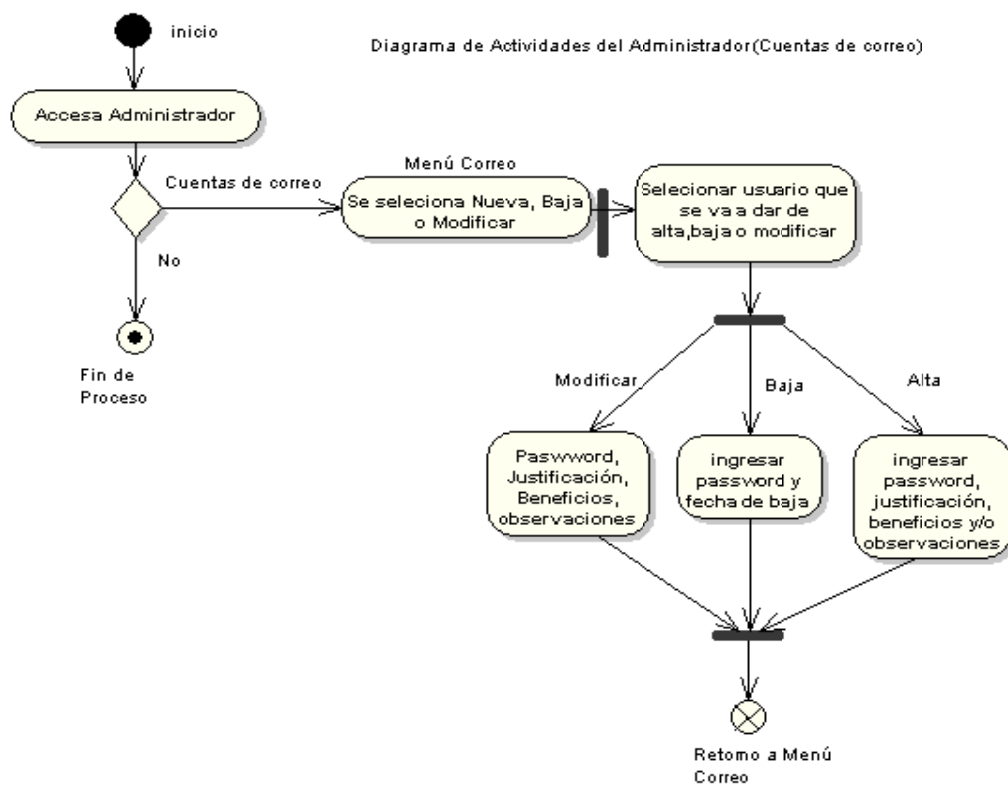


Figura 2.9 Diagrama de Actividades del administrador de una cuenta de correo sistema SIICOFE.

Como se observa en la figura 2.9 cada actividad está delimitada por una línea, en este caso el Administrador realiza la selección de lo que va a realizar ya sea un alta, baja o modificación de una cuenta de correo y por el otro lado debe de seleccionar al usuario. Cabe mencionar que el Administrador no va a registrar al usuario. Para nuestro uso específico es el Auxiliar el que va a registrar al usuario y que el Administrador se va a encargar de dar de alta, baja o modificación de su cuenta de correo.

2.5.5. Modelo conceptual.

El Modelo conceptual es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, e incluso otros elementos de agrupación pueden incluirse en un paquete. Al contrario de los componentes (que existen en tiempo de ejecución), un paquete es puramente conceptual (sólo existe en tiempo de desarrollo). Para nuestro objetivo en el sistema SIICOFE veremos los paquetes en general de todo el sistema. Esto nos servirá para el desarrollo en general ya que da una idea de cómo interactúa cada elemento de nuestro sistema. Cabe mencionar que cada componente lleva implícito un desarrollo detallado de lo que va a realizar.

Esta es una representación de un modelo en partes manejables mediante la agrupación de clases u otros paquetes. Cada carpeta representa los elementos incluidos y la flecha indica la dependencia es la que especifica que un elemento puede afectar a otro.

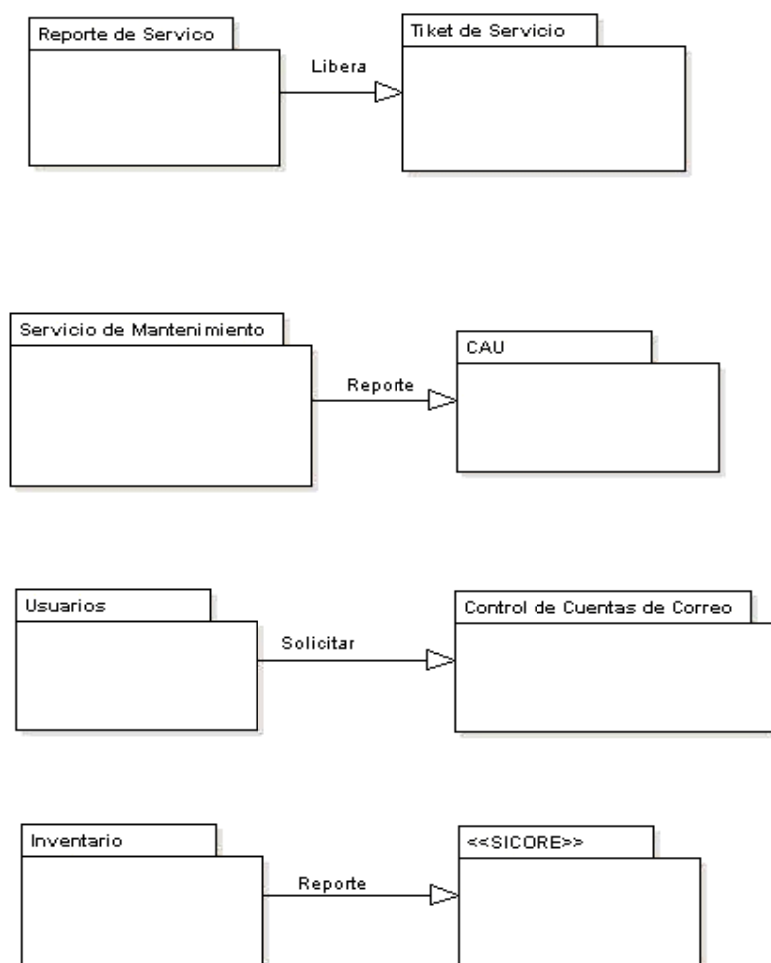


Figura 2.10 Modelo conceptual sistema SIICOFE.

2.5.6. Diagrama General de clases.

Una vez teniendo nuestro modelo conceptual bien definido podemos detallar nuestro diagrama general de clases. Este diagrama nos muestra de manera general como están interactuando nuestras clases. Cuantas y como se relacionan unas con otras. Además de tener un flujo y una relación. Esto permite ver la correspondencia entre cada una de nuestras clases. Recordemos que estas son herramientas que nos permiten entender más nuestro sistema.

Las relaciones pueden ser 1 a 1 , 1 a * (muchos), o * a * (muchos a muchos).

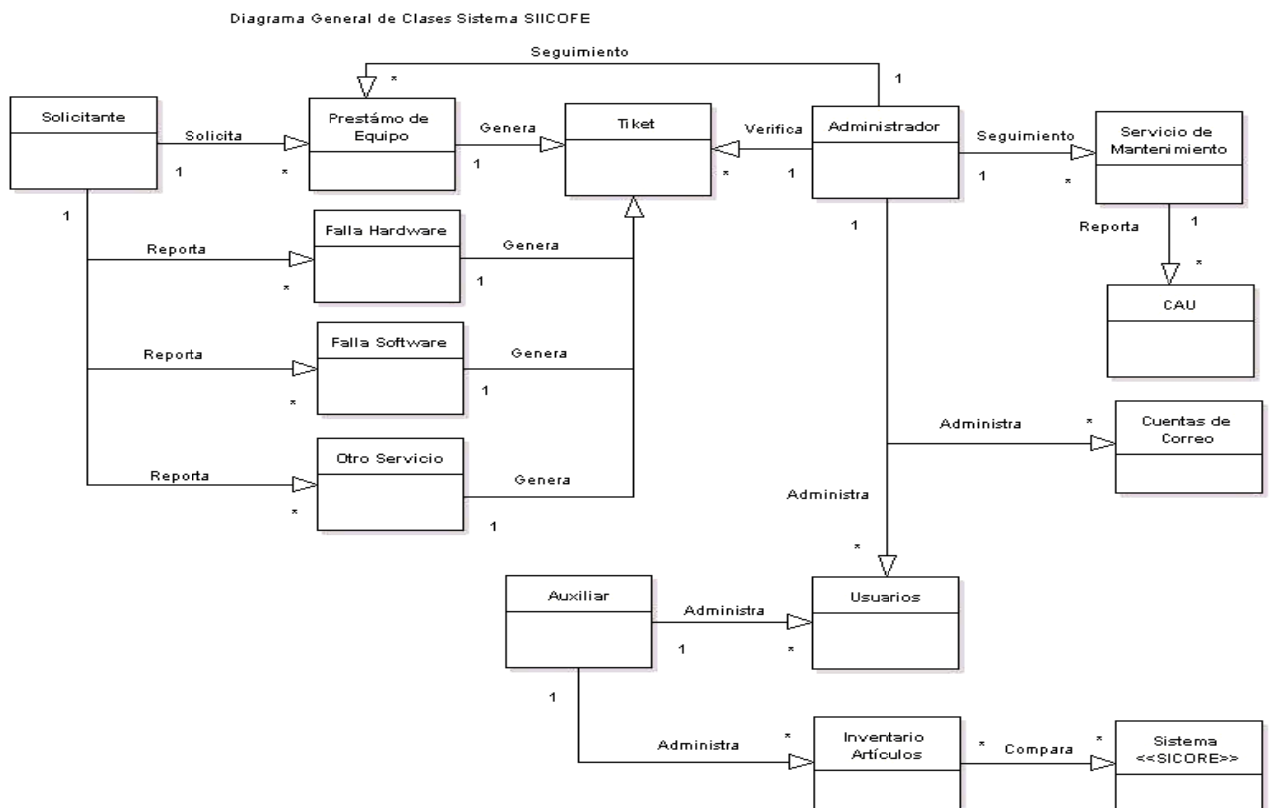


Figura 2.11 Diagrama de clases sistema SIICOFE.

Como es perceptible en este diagrama es muy general y nos ayuda a ver de manera gráfica como está nuestro sistema, pero como veremos en el siguiente punto veremos a detalle cada uno de los elementos que se ven involucrados en el desarrollo del sistema, sin embargo podemos apreciar la manera en que se va desarrollando cada una de las partes de nuestro sistema.

En este diagrama se ve la secuencia que lleva, sin embargo en el diagrama de clases a detalle veremos las variables que van a intervenir en el sistema y la manera en que se van a asociar cada una de ellas. Veremos terminología que se usa como son las clases, atributos y operaciones del sistema.

Para el análisis de nuestro sistema es recomendable evaluar cada uno de los factores que van a intervenir en nuestro sistema y por consiguiente detallar el comportamiento del mismo.

2.5.7. Diagrama de clases a detalle.

Esta es una de las partes medulares de nuestro sistema ya que aquí se encuentran documentadas todas las clases, atributos y operaciones. La relación entre cada una de ellas denominada (asociación) y las agregaciones.

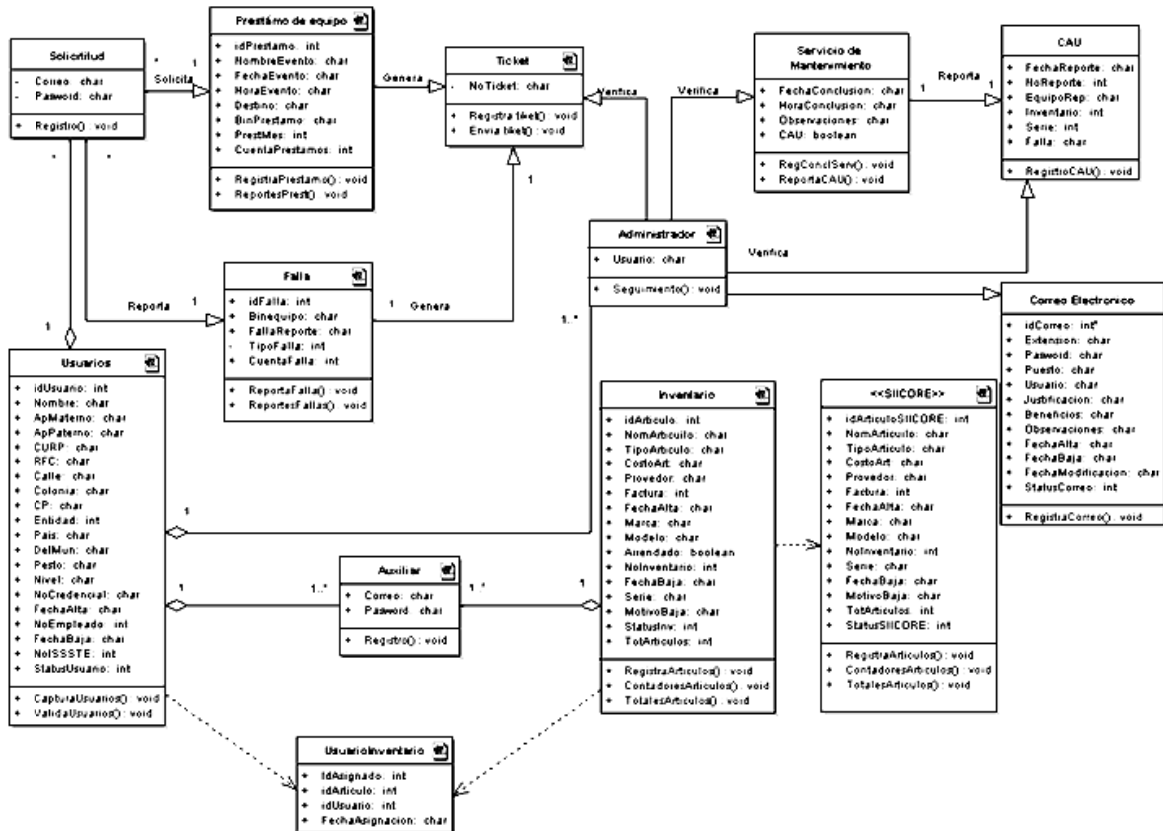





Figura 2.12 Diagrama de clases a detalle del sistema SIICOFE.

Es importante destacar que es cada uno de estos términos (clase, atributo, operación, asociación, relación, dependencia e interface ver apéndice). Para entender un poco el comportamiento de este diagrama de clases. Aquí muestro de manera general el comportamiento de todo el sistema y la interrelación que van a tener cada una de las clases así como sus atributos y la(s) operación(es) que se llevan a cabo en nuestro sistema. Cabe mencionar que la manera en que está relacionada cada una de ellas permitirá facilitar las tareas en la etapa de la programación. Como sabemos en Java se manejan clases y objetos que nos permitirán programarlo de una manera más objetiva y clara.

2.5.8. Diagrama de Clases de Análisis

En este tipo de diagramas se detallan las pantallas propuestas del sistema así como su interacción entre los usuarios y el sistema. Aquí el símbolo  nos denota la interfaz gráfica, la cual está representada por nuestra pantalla así como los campos tanto de nuestra base de datos como de nuestra pantalla de referencia. El siguiente símbolo representa la conexión a la base de datos  en esta parte se va a ver la conexión o conexiones que se estén utilizando y las tareas que se obtengan de el proceso que se tenga que llevar a cabo. Y por último los procesos que se llevan a cabo representado por el siguiente símbolo . En esta parte se consideran los procesos que se llevarán a cabo una vez que el usuario da un clic en solicitar o Cancelar. Este tipo de análisis permite interactuar entre el cliente del sistema y el líder de proyecto para ver tiempos y movimientos del sistema. Y por consecuencia costos tanto del sistema como del lado del cliente.

Reporte Préstamo

Menu SIICOFE

Reporte	Usuario: XXXXXXXX	Extensión: XXXX
Correo	Coordinación: XXXX	Correo: XXXXXXX@XXXX
Préstamo	<input checked="" type="radio"/> Préstamo <input type="radio"/> Falla hardware <input type="radio"/> Falla software <input type="radio"/> Otro	
Mantenimiento	Nombre evento: _____	
Usuario	Fecha evento: _____ Hora evento: _____	
Inventario	Destino: _____	
	<input type="radio"/> Laptop <input type="radio"/> Proyector <input type="radio"/> Internet <input type="radio"/> Conexión (N,R)	
	Solicita	Cancela
	Ticket: XXXXX-120707-11:30	




		
Reporte	Buscar datos de Usuario	Información de Usuario
<ul style="list-style-type: none"> o Nombre del evento o Fecha del evento o Hora del evento o Destino o Ticket <ul style="list-style-type: none"> o Laptop o Proyector o Internet o Conexión 	<ul style="list-style-type: none"> o Conexión 	<ul style="list-style-type: none"> o Usuario o Extensión o Coordinación o Correo
<ul style="list-style-type: none"> ● Solicita ● Cancela 	<ul style="list-style-type: none"> ● Localizar datos del empleado 	<ul style="list-style-type: none"> ● Regresar información

Figura 2.12 Diagrama de clases de análisis de un reporte de servicio del sistema SIICOFE.

El análisis nos permite tener un enfoque general de lo que queremos desarrollar. Y mientras más detallado este menor grado de error podemos tener. Sin embargo como sabemos en los sistemas siempre deben de estar en mantenimiento constante debido a que son perfectibles es decir siempre están en mejora continua.

Diseño del sistema SIICOFE.

Una de las partes medulares en el desarrollo de un SI (Sistema de información) es la etapa del diseño.

Una vez que tenemos el análisis del sistema procedemos a elaborar el mismo una vez aprobado por nuestro cliente. En base a las necesidades que el cliente tiene veremos si en el mercado

3.1. Definición de Audiencia (Niveles de Usuario)

El sitio se orientara a todos los servidores públicos de la COFEMER que tengan a su cargo un equipo de cómputo asignado. Los usuarios podrán dar de alta un reporte y de esa manera se tendrá un control de los servicios del área de Soporte Técnico.

El Auxiliar podrá dar de alta usuarios y inventario.

El Administrador podrá además generar administrar las cuentas de correo y asistir las solicitudes de los usuarios tanto de fallas de software, hardware y de préstamo de equipo.

Como bien sabemos los sistemas se dividen para su estudio en tres partes que son:
Capa de Vista, Capa de Datos y Capa de Negocio.

3.2. Capa de Vista.

En esta etapa del desarrollo se le dará forma a nuestro sistema nos mostrara las pantallas y el diseño del sistema en general. Esta etapa la elaboran los diseñadores principalmente. El desarrollo de nuestro sistema implementado se vera como cada una de las pantallas se correlacionan y permiten interactuar en cada una de ellas. Veremos la navegación del sistema para ello requerimos ver el Mapa del sitio.

3.2.1. Mapas del sitio

El mapa del sitio permitirá ver de una manera gráfica como estará distribuido nuestro sistema y como se va a ir interpretando cada una de las pantallas. Si realizamos una modificación del sistema lo veremos reflejado en el proceso del mismo.

Una de las herramientas es el Árbol Organizacional en donde se ve de manera grafica en donde va a estar situada nuestra aplicación. En el árbol organizacional podemos ver que el Sistema Integral COFEMER (SIICOFE), se encuentra situado dentro de la intranet de la COFEMER (www.cofemer.gob.mx/intranet). Conviviendo con otros sistemas ya implementados de la intranet como se muestra en la figura 3.1.

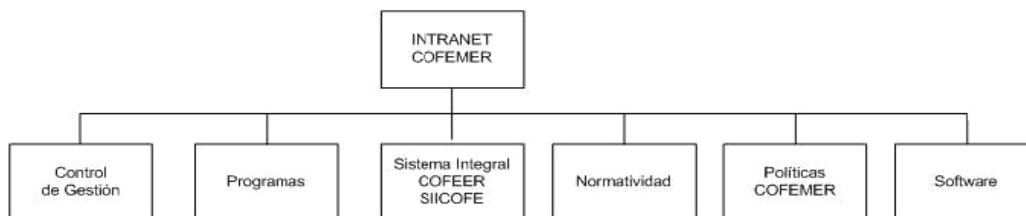


Figura 3.1 Mapa del sitio donde estará alojado el sistema SIICOFE.

Una vez definido en donde va a quedar nuestro sistema analizaremos como va a estar estructurado utilizando la herramienta llamada árbol funcional.

3.2.2. Árbol Funcional.

El árbol funcional permitirá ver la manera como están organizadas cada una de las pantallas y el flujo de cómo se va a estructurar cada una de las partes de la que va a estar detalladas las pantallas y los módulos que va a constar nuestro sistema como se muestra en la figura 3.2.

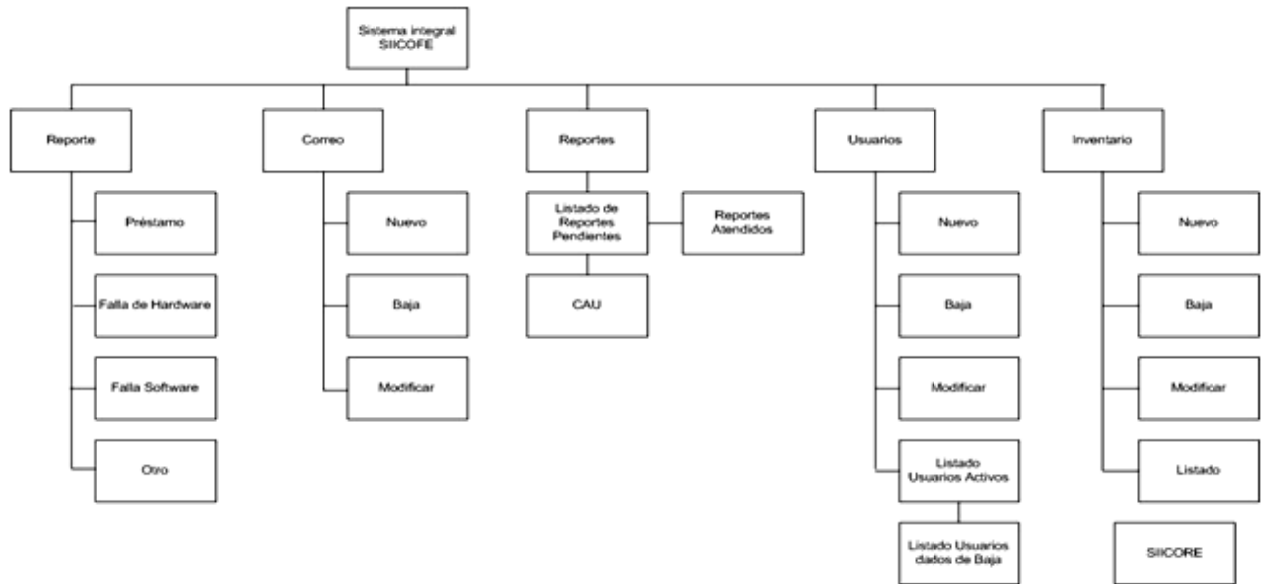


Figura 3.2 Árbol funcional del sistema SIICOFE.

A continuación mostrare solo alguna de las primeras páginas de nuestro sistema.

3.2.3. Interfaces Web

Interfaces de contenido de Información

Interfaz principal. En esta parte tenemos la zona de navegación y la zona de información, para nuestro sistema serán de gran ayuda estas especificaciones.



Figura 3.3 diagrama de Interfaz de contenido sistema SIICOFE.

Así es como se muestra nuestro menú principal en donde se estará navegando constantemente. Estas páginas son elaboradas en código HTML y los fondos son invocados por hojas de estilo que nos permiten una interfaz que puede ser modificada fácilmente en el código.

Mostrare una parte del código de HTML del menú principal en donde invocamos una hoja de estilo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {background-image: url(IMAGENES/MarcoIzQ2.jpg);
background-repeat: no-repeat;}
-->
</style>
<script src="SpryAssets/SpryMenuBar.js" type="text/javascript"></script>
<link href="SpryAssets/SpryMenuBarVertical.css" rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
.style3 {color: #FFFF00}
-->
</style>
</head>
<body>
<table width="200" border="0">
<tr> <td bordercolor="#009999"><ul id="Menu" class="MenuBarVertical">
<li><a class="MenuBarItemSubmenu" href="#">Reporte</a>
<ul> <li><a href="RepPrestamo.html" >Prestamo</a></li>
<li><a href="RepFallaHardware.html">Falla Hardware</a></li>
<li><a href="RepFallaSoftware.html">Falla Software</a></li>
<li><a href="RepOtro.html">Otro</a></li>
</ul> </li>
<li><a href="#" class="MenuBarItemSubmenu">Correo</a>
<ul> <li><a href="CorreoNuevo.html">Nuevo</a></li>
<li><a href="CorreoBaja.html">Baja</a></li>
<li><a href="CorreoModificar.html">Modificar</a></li>
</ul> </li>
<li><a href="Pendientes.html">Pendientes</a> </li>
<li><a href="#" class="MenuBarItemSubmenu">Usuarios</a>
<ul> <li><a href="UsuarioNuevo.html">Nuevo</a></li>
<li><a href="UsuarioBaja.html">Baja</a></li>
<li><a href="UsuarioModifica.html">Modificar</a></li>
<li><a href="ListaUsuarios.html">Listado</a></li>
</ul> </li>
<li><a href="#" class="MenuBarItemSubmenu">Inventario</a>
<ul> <li><a href="InventarioNuevo.html">Nuevo</a></li>
<li><a href="InventarioBaja.html">Baja</a></li>
<li><a href="InventarioModificar.html">Modifica</a></li>
<li><a href="Construccion.html">SIICOFÉ</a></li>
</ul> </li> </ul> </td> </tr> <tr>
<td bordercolor="#009999"><p class="style3">Dr &copy COFEMER</p>
<p class="style3">Federico S&aacute;nchez Contreras</p>
</td> </tr></table>
<p>&nbsp;</p><script type="text/javascript">
<!--
var MenuBar1 = new Spry.Widget.MenuBar("Menu", {imgRight:"SpryAssets/SpryMenuBarRightHover.gif"});
//-->
</script>
</body>
</html>
```

Figura 3.4 Código de página de interfaz de sistema.

Como se muestra en el código aquí solamente se maneja el diseño del sistema. En la siguiente etapa del diseño se integraran los códigos en Java para manipular información.



Figura 3.5 Interfaz aprobada del sistema SIICOFE.

Usted podrá dar un clic en la opción a seleccionar y dar un clic. Una vez seleccionada la opción se activara la ventana de Préstamo como se muestra en la figura 3.5.

Una vez implementadas todas nuestras ventanas de navegación podremos continuar con el proceso de la Capa de datos en donde se definirá la base de datos y por último las reglas del sistema.

Es importante considerar los estándares internacionales para la mejor operación de nuestra página obviamente sin perder de vista lo que requerimos y el objetivo que vamos a tener de la misma. Considere todos los pro y contra de agregar o quitar elementos y verifique los requerimientos del sistema para su aprobación.

Aquí muestro la pantalla de Préstamo en la etapa de vista.

The image shows a form titled 'Reporte - Préstamo' on a teal background. It contains several input fields: 'Usuario:' and 'Coordinación:' with placeholder text 'XXXXXXXXXXXXXXXXXX'; 'Extensión:' with 'XXXXXX'; 'Correo:' with 'XXXXX@XXXXXXXX.XXX.XX'; 'Nombre Evento:' and 'Fecha Evento:' with empty text boxes; 'Destino:' and 'Hora Evento:' with empty text boxes. Below these are four checkboxes: 'Laptop', 'Internet', 'Proyector', and 'Conexión (N,R)'. At the bottom left are three buttons: 'Borrar', 'Solicita', and 'Cancela'. At the bottom right, there is a text label 'Tiket: XXXXXX-120707-11:30'.

Figura 3.6 Pantalla propuesta de reporte de préstamo de equipo.

En la capa de vista debemos de considerar varios puntos que son:

- Los sitios Web deben de tener un peso máximo permitido por página en (kb) para no saturarse cuando se estén descargado por primera vez.
- Las Normas Internacionales mencionan que más de:
- 5 segundos para que aparezca algo visible.
- 10 segundos para que aparezca algo legible.
- 30 segundos para hacer un clic hacia otra parte del sitio.
- Diagramación de las páginas web. (utilizando hojas de estilo **CSS**)
- Construcción de estructuras de tablas para presentación de contenidos.

- Si desea hacer una presentación en **Macromedia Flash** se recomienda no hacerlo directamente en la portada.
- Utilizar marcos o <<frames>> ya que su finalidad es agrupar archivos para que se desplieguen de manera simultánea permitiendo a los usuarios ver varios contenidos al mismo tiempo.
- De preferencia no utilizar background ya que agrega un paso excesivo a los sitios afectando el tiempo de descarga y acceso a la información.

Normas para incorporar Elementos gráficos y multimedia.

- Optimizar el peso de las imágenes; se debe de bajar máximo posible sin perder calidad y se debe de reducir el número de colores(72 dpi es la norma).
- Elegir el formato adecuado de las imágenes, como sabemos existen diferentes formatos unos más pesados que otros yo recomiendo el jpg ya que optimiza el espacio y la calidad de la imagen.
- Ubicación de las imágenes. Es recomendable utilizar un solo directorio para la colocación de las imágenes. Para la seguridad se recomienda impedir que un programa visualizador pueda ver el contenido de dicho directorio.
- Utilizar el atributo ALT en imágenes en el código HTML para que despliegue texto antes de las imágenes y facilite de esta manera la comprensión del contenido de los usuarios.
- Ofrecer plug-ins Esto es importante cuando se utilizan archivos multimedia esto es válido para sitios que ofrecen presentaciones principalmente de tecnología Flash.
- Indicar el peso de los archivos. Esto es con la finalidad de que se tenga información de el tamaño del archivo a bajar y más que nada el tiempo que pueda tardar en bajar este archivo.

Como sabemos nuestro sitio debemos de manejar ciertos Estándares para mejorar el rendimiento de nuestra página alguno de los estándares deben de ser:

- **Validación de HTML.** Existe un página en el sitio (<http://www.w3c.org/>) que se encarga de validar una página HTML y XML en la construcción de un sitio Web. Con esta validación se asegura que ningún visualizador que cumpla con los estándares internacionales tenga problema alguno para su visualización y por consiguiente para su certificación.
- Es necesario **validar las hojas de estilo (CSS)**, estas hojas serán validadas en la página en el sitio W3C (<http://jigsaw.w3.org/css-validator>) mediante el cual será validada la hoja de estilo en cascada (Cascade Style Sheet o CSS), mediante la cual se describe la forma de presentar contenidos de una página Web.

Debemos de realizar muchas pruebas antes de subir nuestra página, algunas de las pruebas son:

- Verificar la accesibilidad con herramientas automáticas y revisión humana.
- Utilizar los métodos de validación desde las primeras etapas del desarrollo.
- Utilizar una herramienta automatizada de validación de la accesibilidad y la navegación.
- Validar la sintaxis de la programación de las páginas.(ver el W3c).
- Validar las hojas de estilo. (CSS).
- Utilizar varios navegadores antiguos y actuales. (verificar como se muestra en cada uno).
- Revisar la ortografía y la gramática para que la página se muestre clara.
- Revisar el documento en cuanto a claridad y simplicidad.

Con esto podemos asegurar el buen funcionamiento de nuestra pagina en cuanto a navegación y verificar que no tenga ligas rotas, por otro lado es necesario verificar que cada institución cuenta con una Normatividad. Y aquí en el Gobierno Federal se manejan estándares para los colores de la dependencia con quien se trabaja. Y formas que deben de ser aprobadas por el OIC (Órgano Interno de Control de la Secretaria de Economía en conjunto con el Gobierno Federal).

3.3. Capa de Datos

Como sabemos la información de nuestro sistema debe de estar almacenada en un repositorio de datos ya sea de una aplicación Web o de escritorio (Stand Alone) ya sea para obtener información o para almacenarla. Para almacenar la información lo podemos llevar a cabo por medio de un archivo de texto, archivos delimitados por comas, o un sistema administrador de base de datos (SMBD).

También es importante contar con un lenguaje de consulta de datos y conocimientos de procedimientos almacenados. Esto con la finalidad de realizar la manipulación de la Base de Datos.

Para entender esta parte de la capa de datos debemos de tener un conocimiento de manejo de base de datos y como se crean y manipula la información de la Base de datos.

Debemos de tener conocimiento de los siguientes conceptos como son: dato, campo, registro, tabla y definir que es una base de datos (ver apéndice). Una vez que tenemos bien definidos estos términos implementamos la base de datos en base al análisis que se había realizado en la unidad 2 en la figura 2.12. Como es de observarse el diagrama detalla las tablas que van a ser creadas y como se van a relacionar unas con otras.

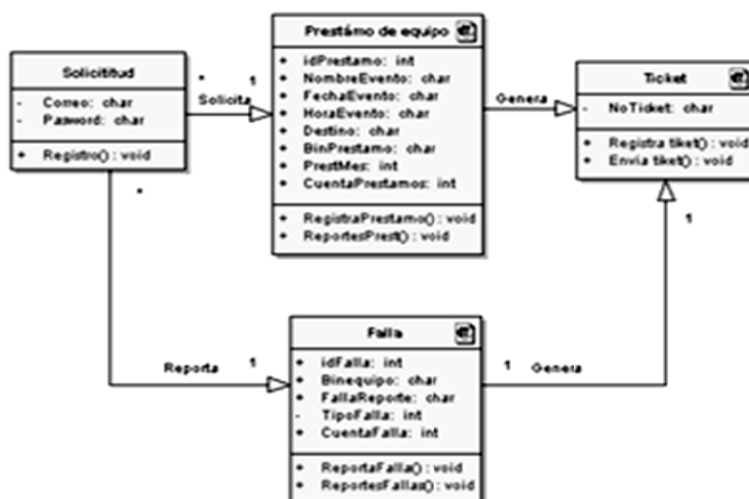


Figura 3.7 Diagrama de clases.

Con este diagrama de clases se elaboraron cada una de las tablas con sus respectivos campos.

PrestamoEquipo							
idPrestamo	NombreEvento	FechaEvanto	HoraEvento	Destino	BinPrestamo	PrestMes	CuentaPrestamos

Tabla 10 Campos de tabla PrestamoEquipo.

Una vez creada cada una de las tablas procedemos a relacionarlas cada una de ellas con su correspondiente tipo de relación. Es importante mencionar que manejaremos la información en MySQL. Database Server es una base de datos de código abierto más usada en el mundo. Por funcionalidad y versatilidad es una herramienta de gran apoyo ya que permite manejar tablas de manera rápida y confiable. La extensiva reutilización del código dentro del software permite tener características y un funcionamiento muy aceptable ya que la velocidad, compactación, estabilidad y fácil despliegue de transacciones. La exclusiva separación del Core Server del manejador de

tablas, permite funcionar a MySQL bajo control estricto de transacciones. Teniendo en cuenta que MySQL es usado en gran número de sistemas web se han implementado un caché de consultas para acelerar las consultas que son ejecutadas comúnmente.

El SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

El lenguaje puede ser descompuesto en dos sub-lenguajes, cada uno de ellos orientados a un fin específico como son:

DDL (Data Definition Language). Lenguaje de definición de datos. Que permite crear y definir nuevos objetos de bases de datos (Base de datos, tablas, índices, vistas, etc.).

Existen tres operaciones básicas como son el CREATE, ALTER y DROP.

DML (Data Manipulation Lenguaje). Lenguaje de manipulación de datos. Que permite recuperar, ordenar y filtrar los datos de la Base de Datos.

Existen los comandos INSERT, UPDATE, DELETE, SELECT. Que nos servirán para manipular la información de la Base de datos. El SELECT nos servirá para realizar consultas detalladas de cada uno de los registros previamente almacenado.

Procedimientos Almacenados.

Un procedimiento almacenado es un conjunto de comandos de SQL que pueden almacenarse en el servidor. Una vez que se tiene el procedimiento almacenado no necesitan realizar los comandos individuales pero se refieren al procedimiento para que este se ejecute desde el servidor y con una aplicación se invoque a este procedimiento.

El procedimiento almacenado servirá cuando múltiples aplicaciones cliente escriben en distintos lenguajes o funcionan en distintas plataformas, pero necesitan realizar la misma operación en la base de datos.

Cuando la seguridad es muy importante. Esto proporciona un entorno seguro y consistente para asegurar cada operación.

Para nuestro sistema utilizaremos procedimiento almacenado ya que se va a estar interactuando con diferentes consultas de información con las tablas de Usuarios y Artículos. Este procedimiento almacenado permite mejorar el rendimiento ya que necesita enviar menor información entre el servidor y el cliente. El intercambio que hay es en que aumenta la carga del servidor de la base de datos ya que la mayoría del trabajo se realiza en la parte del servidor y no en el cliente.

Los procedimientos almacenados permiten tener bibliotecas o funciones en el servidor de base de datos. Esta característica es compartida por lenguajes de programación modernos que permiten este diseño interno con la utilización de clases para su operación y manejo de información.

Utilizando estas características del lenguaje de programación cliente es de gran utilidad para el programador incluso fuera del entorno de la base de datos.

En esta etapa del diseño solo se verá como se integra la información y como se modifica la misma en la captura de la información. En la figura 2.12 del capítulo 2 mostré el diagrama de clases para una mejor comprensión de cómo va a estar estructurada nuestra base de datos.

3.4. Capa de Negocio.

En esta fase del diseño se van a llevar a cabo todas las transacciones que van a realizar las acciones del usuario e invoca cambios en el modelo y probablemente en la vista. En la aplicación web el controlador recibe la petición del usuario, interactúa con el modelo para procesar los datos y hace disponible esos datos a la vista. El controlador tiene la capacidad de procesar datos y mostrarlos, es autosuficiente.

El usuario interactúa con la interfaz del usuario de alguna forma (por ejemplo en el registro de un servicio).

El controlador recibe la notificación de una acción solicitada por el usuario. El controlador gestiona el evento que llega en ese momento. El controlador accede al modelo, actualizándolo o modificándolo de forma adecuada a la acción solicitada por el usuario.

El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierto direccionamiento entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio a la vista aunque puede dar la orden a la vista de que se actualice. Ver figura 1.3.

Para entender un poco más esta parte veremos algunos conceptos elementales.

Servlet. Un servlet es un módulo que corre dentro del servidor y enriquece los servicios orientados a una petición y a una respuesta en nuestro sistema cuando iniciemos con la carga de un servicio solicitado vemos que cuando se envía esta solicitud el controlador responderá con un ticket de servicio para responder a el servicio y este a su vez mandara una petición dependiendo del tipo de servicio solicitado al área de soporte para que atienda el servicio.

En una implementación a futuro se me ocurre que como es un área de servicio además de almacenar el servicio en el registro correspondiente envíe un mensaje de texto a un teléfono celular para ser atendido (en este caso solo para informar al personal de soporte que hay un nuevo servicio en petición).

Los servicios son bloques de código que implementa un protocolo a nivel de aplicación tal como FTP, DHCP o HTTP.

Los servlets están escritos en lenguaje Java y permiten el acceso a métodos remotos, manejo de grandes volúmenes de información, colaboración entre múltiples usuarios, sincronización de datos y autenticar algún proceso o evento que se esté manejando.

Para comprender un poco más esta parte tendremos que entender los servlets y como se crean y como se destruyen ya que en el sistema estaremos invocando los servlets para el uso y manejo de nuestra aplicación. Cabe mencionar que no requerimos mucha seguridad lo que si vamos a ver es la integridad de nuestro sistema.

El ciclo de vida de un servlet consta de cuatro pasos.

1. Carga e instancia del servlet por un servicio.
2. Invoca con el método `int`
 - Se llama este método cuando el método es cargado.
 - Se llama solamente una vez.
 - No puede ser recargado, mientras no halla sido destruido por el método `destroy`.
 - No es factible utilizar la concurrencia cuando es llamado, ya que este es invocado solamente una vez.
3. Invocación de `Service`.
 - Los servicios son los métodos encargados de procesar las peticiones de los clientes, mediante los métodos `doGet`, `doPost`, los cuales reciben como parámetros los objetos `request` and `response`.
 - Aquí se permite el manejo de concurrencia.
4. Invocación de `Destroy`.
 - Este método es invocado cuando un servicio descarga un Servlet de memoria este método deshace cualquier trabajo inicializado, el estado de sincronización es persistente con el servlet que se encuentra en memoria.

Método `doGet`.

Este método es llamado por cada petición que se realice al servlet, aquí cabe mencionar que `doGet` es solamente uno de varios métodos HTTP soportados. El método `doGet` es utilizado en nuestro desarrollo ya que el cliente genera una petición tipo GET, `doGet` acepta los objetos `HttpRequest`, `HttpResponse`.

Método `doPost`.

El método `doPost` es llamado por cada petición que se realice al Servlet y es necesario mencionar que `doPost` es solamente uno de varios métodos HTTP soportado. El método `doPost` es usado ya que el cliente genera una petición tipo POST, `doPost` acepta los objetos `HttpRequest`, `HttpResponse`.

Manejo de Formularios.

Para poder poner un ejemplo del funcionamiento del sistema vernos la parte de el registro de un servicio con un usuario y la fecha.

```
package registro.usuario.view;
import registro.dao.Usuario;
import registro.usuario.dao.UsuarioDAO;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;

public class Gestor extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        UsuarioDAO dao = new UsuarioDAO();
```

```
String sMensaje = "";
int id = Integer.parseInt((String) request.getParameter("txtId"));
String nombre = (String)request.getParameter("txtNombre");
String fecha = (String)request.getParameter("txtFecha");
String ACTION = (String)request.getParameter("hdnActi
Usuario usuario = new Usuario(id,nombre,fecha);
if(ACTION.equals("ALTA")){
    dao.insertAlumno(Usuario);
    sMensaje ="Datos Insertados!";
    request.getSession().setAttribute("txtNombre",nombre);
    request.getSession().setAttribute("txtFecha",fecha);
    response.sendRedirect("vUsuario4.jsp");
}else if(ACTION.equals("EDITAR")){
    dao.updateUsuario(usuario);
    request.getSession().setAttribute("txtNombre",nombre);
    request.getSession().setAttribute("txtFecha",fecha);
    response.sendRedirect("vUsuario3.jsp");
    sMensaje ="Datos Actualizados!";

}else if(ACTION.equals("ELIMINAR")){

    Usuario usuario = dao.getUsuario(id);
    dao.deleteUsuario(usuario);
    sMensaje ="Datos Eliminados!";
    request.getSession().setAttribute("txtNombre",usuario.getNombre());
    request.getSession().setAttribute("txtFecha",usuario.getFecha());
    response.sendRedirect("usuario2.jsp");

}else if(ACTION.equals("CONSULTAR")){

    Usuario usuario = dao.getUsuario(id);

    request.getSession().setAttribute("txtNombre",usuario.getNombre());
    request.getSession().setAttribute("txtFecha",usuario.getFecha());
    response.sendRedirect("vusuario1.jsp");

}else if(ACTION.equals("CONSULTAR_EDITAR")){

    Usuario usuario = dao.getUsuario(id);
    request.getSession().setAttribute("txtId", usuario.getId() + "");
    request.getSession().setAttribute("txtNombre",usuario.getNombre());
    request.getSession().setAttribute("txtFecha",usuario.getFecha());
    response.sendRedirect("datosusuario.jsp");
}
out.println("<html>");
out.println("<head><title>servletUsuario</title></head>");
out.println("<body>");
out.println("<p>" + sMensaje + "</p>");
out.println("</body></html>");
out.close();
}
}
```

Figura 3.8 Código página registro de un servicio.

El método doGet es el encargado de procesar las peticiones que realizan los formularios mediante la petición GET, la característica del envío de datos mediante este método es que son visibles y manipulables por el usuario.

El método doPost procesa peticiones realizadas por el cliente mediante la petición POST, la transferencia de datos no es visible por el usuario y además permite una transferencia mayor de información.

El API JDBC provee un conjunto de clases con interfaz genérica, que son implementadas por un controlador de base de datos específico. El API provee accesos a cualquier base de datos relacional que tenga soporte JDBC.

El uso de la base de datos consta de 5 pasos, los cuales son:

1. Carga del driver
2. Establecer la conexión
3. Realizar la consulta a la base de datos.
4. Extraer los resultados
5. Cerrar las conexiones.

Como podemos ver estos pequeños tips nos ayudaran a implementar un sistema Web. Aunque se pueden detallar más en cursos completos de Java, SQL, HTML aplicado con CSS. Y uno de los cursos que es pieza fundamental para el análisis y desarrollo lo que es UML aplicado a el desarrollo de un sistema.

Creo que este trabajo servirá de pauta para iniciar con la generación de una página Web.

Implementación del Sistema.

Una vez terminadas las fases de nuestro sistema procedemos a la implementación del mismo es decir tenemos que ponerlo en operación. Una vez que ya tenemos el sistema en si procedemos a llevar a cabo las pruebas del sistema y evaluación del mismo para verificar que la información capturada y los procesos elaborados se estén realizando correctamente.

Ya teniendo el servidor instalado y en operación se procede a colocar el sistema en el servidor de aplicaciones donde se encuentra instalada la intranet de COFEMER (www.cofemer.gob.mx/intranet). Ingresamos a la intranet que se encuentra en el servidor 207.248.177.25. Este servidor aloja a la página de COFEMER. Aquí tenemos instalado el SQL Server en esta versión se genero la base de datos la cual operara para la manipulación de la información nuestra base se llama SIICOFE y en ella se encuentran todas nuestras tablas de datos. Se instalo el sistema en el servidor y se ejecuto para pruebas antes de la puesta en marcha.

4.1. ¿Que probar antes de Implantar el sistema?

4.2. Pruebas de Interfaces y Contenidos.

Consiste en hacer revisiones de la forma en que se despliegan las páginas del sitio ver si cumplen con los términos de referencia, verificar si cumplen con los estándares mínimos que se hayan definido como meta. Las pruebas a realizar en esta etapa son:

4.2.1.Verificación de contenidos.

Es una prueba para revisar si todos los contenidos que se han especificado en términos de referencia de datos corresponden a lo sugerido dentro del desarrollo. Se puede hacer de manera manual o automática.

Sistema manual. Para ello se recomienda primero construir un índice de contenidos y luego verificar la existencia de cada uno de los ítems que contiene a través de hacer un recorrido a detalle del sitio. Lo que debemos de verificar son:

1. Verificación de ortografía y redacción.
2. Verificación de los enlaces principales.
3. Verificación de imágenes en página.
4. Verificación de lista de chequeo de accesibilidad.

4.2.2.Sistema Automático.

Se encarga de verificar enlaces rotos lo cual se verifica con sistemas basados en Internet o software especializado.

- a. Los sistemas basados en internet utilizan el servicio W3C verificando el link (<http://validator.w3.org/checklink>).

El ejemplo muestra la validación de la pagina para verificar posibles fallas en la página solicitada para su corrección, si los errores son leves aparecerá como se muestra en la figura 4.1 mencionando la referencia del tag de inicio, posible solución es que no se cerró otro tag anteriormente.

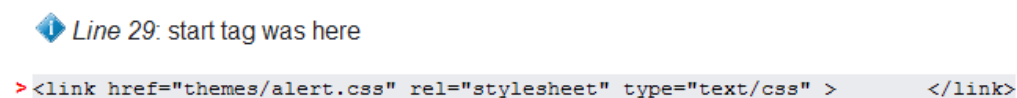


Figura 4.1 Muestra de fallas de un link.

Por otro lado si nuestro error es grave aparecerá un mensaje como en la figura 4.2 en donde se notificara la causa del error.

✖ **Line 69, Column 149: document type does not allow element "span" here**

```
..., title: "<span class='CeldaGridDet'>...<strong>TNT</strong></span>", showEffect
```

Figura 4.2 Error grave en código.

En este sitio podemos validar un HTML por otro lado podemos verificar un CSS en la pagina (<http://jigsaw.w3.org/css-validator>)

4.2.3. Verificar las interfaces. En esta prueba se revisan todos los aspectos gráficos del sitio para determinar si su despliegue en las páginas es el correcto. Dentro de esta verificación se debe de tomar en cuenta:

- **Los plug-ins.** Cuando se requieren elementos audiovisuales o interactivos se debe de ofrecer un enlace para los usuarios que no lo tengan instalado.
- **Consistencia de la diagramación.** Las páginas del sitio deben contener elementos consistentes es decir áreas fijas que permitan identificar la página para saber la referencia de integración de nuestro sistema.
- **Ejecución en diferentes browsers.** Con esto se verificara el sistema en diferentes browsers los más utilizados en nuestra área son el Internet Explorer 8 y Firefox. En estos navegadores ejecutamos la página Web y vemos su presentación y que no tenga rupturas de página o que las imágenes utilizadas no se vean bien.
- **Imágenes escaladas.** Esto se refiere a la presentación cuando estemos manejando imágenes. Es decir para una imagen pequeña es recomendable no sobrepasar el límite de 5 kb en iconos, para los de tamaño mediano no sobrepasar de 25 kb. Debemos de recordar que el peso máximo para una página es de 100 kb. Incluyendo todos sus elementos.¹

Como sabemos estas no son reglas pero nos sirven de referencia para tener una página con estándares generales para una mejor operación y para un buen funcionamiento en diferentes ámbitos de trabajo y para optimizar al mínimo posibles errores que cuando elaboramos nuestra página Web. Esto solo es pauta para poder elaborar un sistema óptimo y eficiente en cuanto a metodología. En la práctica utilizamos varias estrategias para optimizar tiempos y mejorar la calidad de las imágenes y llevamos a cabo trucos para aplicar una mayor seguridad y mejor eficiencia.

4.3. Pruebas de Funcionalidad y operación.

Como sabemos debemos de realizar otro tipo de pruebas denominadas de funcionalidad y Operación. En esta parte del proceso nos encargaremos de verificar la carga de datos así como la validación de cada uno de los datos que se ingresen a la base de datos.

4.3.1. Validación de formularios. Aquí vamos a verificar cada una de las páginas de nuestro sitio Web y validaremos que la información ingresada al sitio sea la del formulario. Aquí puedo hacer notar que muchas veces al ingresar las fechas tenemos problemas con los formatos horarios entre los equipos de

¹ Guía Wb 1.0 pagina 75 Imágenes Escaladas.

los usuarios. Sin embargo debemos de validar esta información para una buena operación de nuestro sitio. En la validación de información debemos de incluir campos obligatorios para que el usuario (se consideran con un asterisco “*”) en general los tenga en cuenta. Ver figura 4.3

Dirección del Registro Federal de Trámites y Servicios
Subdirección de Sistemas
Departamento de Soporte Técnico

Control de Soporte Técnico **NO REPORTE:** COFE-PRES-0001-150309-11:30

DATOS SOLICITANTE **Fecha de solicitud:** 15 03 09

Nombre: FEDERICO SANCHEZ CONTRERAS **Extensión:** 22610

Titular CGMIR CGMRI CGMRS CGMRSAJ CGPE CEM DA TOOLKIT **Correo:** fscontre @cofemer.gob.mx

A) Préstamo de equipo

Nombre del evento: CAPACITACION

Fecha: 19/03/2009 **Hora:** 12:00 **LAPTOP/PC:** **CAÑÓN / PROYECTOR:** **RED (INTERNET):**

Destino: PISO 12 SALA A

B) Servicios 1) Reparación 2) Configuración 3) Instalación 4) Asesoría

HARDWARE
 LAPTOP/PC IMPRESORA SCANNER UNIDAD CINTA SERVIDOR Otro

SOFTWARE
 WINDOWS OFFICE ANTI/VIRUS GROUPWISE VIRUS Otro

Falla:

SALIR **REGISTRAR**

Figura 4.3 Pagina de registro de un reporte.

Validaciones Locales. Para reducir la carga de validaciones en el Servidor se recomienda incorporar validaciones del lado del cliente utilizando en forma estándar el lenguaje Java script. En este sistema se válida para el empleado datos como el RFC, CURP y el No de credencial que deben de ajustarse a los requerimientos de validación de información de estos. El RFC y CURP tienen un algoritmo para la validación de la información del empleado. Por otro lado cuando se ingresa el correo electrónico debemos de verificar su sintaxis para evitar que no sea un correo electrónico valido en el sistema. Esto con la finalidad de emitir reportes y actualizar nuestra lista de correos electrónicos. Los números de inventarios de equipo perteneciente al empleado se validan por la cantidad de datos que ingresa ya que los números de inventarios pertenecen a la Secretaría de Economía. Aquí he ingresado inventarios iguales y marca un error por tener datos iguales en lo que son inventarios y series. Si capturamos letras en el inventario se envía un mensaje de error captados por un catch de errores.

El sistema cuenta con diferente validación en todas las páginas y verifica cada cuadro de dialogo que me permite mostrarle al usuario el error que cometió en la captura de la información. El sistema va a ir almacenando los errores con fechas y hora del sistema (log-error) con la finalidad de ir mostrando posibles fallas de los usuarios para soportar estadísticas del sistema.

Una vez que se ingresan los datos debemos de verificar la información en la Base de Datos que se esté grabando correctamente y en los campos establecidos en nuestro mapa de datos. Una vez verificada esta información y corroborada por nuestro jefe, solicitamos al área su aprobación y puesta en operación. No sin antes iniciar la carga de información inicial.

4.3.2.Carga inicial.

Una vez que verificamos que la información se almacena de manera eficiente se procede a la carga de información. En este sistema se cuenta con información de contenido en Excel y de ahí se filtrara a la base de datos por medio de un proceso. Una vez que cargamos a los empleados de la COFEMER y procedemos con los catálogos de artículos, los correos ya cargados, de Estados de la República Mexicana y el inventario de computo asignado a los usuarios. Esta carga se realizara de manera inicial.

Ahora ya podemos operar el sistema y veremos cómo se comporta el mismo en una fase de pruebas que se estarán llevando a cabo.

4.3.3.Pruebas de Carga.

En esta etapa se refiere a la capacidad máxima que tiene un servidor Web (hardware y software), para atender a un conjunto de usuarios de manera simultánea. Las actividades de esta etapa tienen relación con comprobar de manera anticipada el funcionamiento que tendrá el servidor del sitio Web este en plena operación. Las pruebas en este caso consisten en simular una carga masiva de información y verificar cómo se comporta el servidor para verificar si el servidor soporta cargas fuertes de trabajo. Lo que se debe de verificar es:

1. Tiempo de acceso de los usuarios a los datos.
2. Volumen de datos y ancho de banda utilizado.
3. Archivos solicitados y tiempos usados en transferencia de datos.
4. Tiempo de espera de los usuarios tras hacer un clic.
5. Tiempo de respuesta a clics de varios usuarios.
6. Niveles de error tras clics de los usuarios.

4.3.4.Seguridad.

Nuestro sitio no requiere de mucha seguridad ya que se encuentra alojado en RTN (red Tecnológica Nacional), ahí se encuentra alojado el hosting el cual cuenta con niveles de seguridad altos. Aun así nuestro servidor se encuentra protegido contra diferentes ataques de virus o de intrusos. Como sabemos hoy en día la seguridad tiene una gran complejidad debido a los constantes ataques de que son objeto las páginas del Gobierno Federal. Sin embargo tenemos que estarnos capacitando constantemente para prevenir esos ataques y mejorar los niveles de seguridad de nuestro sitio.

Una vez realizado lo anterior ponemos nuestro sistema en marcha para su operación. Con este sistema se mejora la coordinación entre el área Administrativa y el área de Informática de la COFEMER. Además de contar con herramientas para el control de los servicios del área de Soporte Técnico de la COFEMER.

CONCLUSIONES.

El conocer técnicas de UML para el desarrollo del sistema permite detallar de manera confiable las necesidades de los usuarios teniendo en cuenta que es lo que se debe de llevar a cabo y en cuanto es el tiempo que se debe de establecer para la entrega del proyecto.

El proyecto SIICOFE me permitió mejorar las técnicas de análisis y desarrollo para su implementación. Me dejo ver y analizar el como el desarrollo en java tiende a mejorar la perspectiva de otros lenguajes y como al ser de uso libre va a permitir competir de una manera muy fuerte con Microsoft.

La utilización de bases de datos gratuitas permitirá a clientes con bajos recursos poder implementar sistemas que se desarrollen a la medida con un bajo costo. Pero como sabemos la implementación incluye un costo que los empresarios y el gobierno en si deberán de contemplar para la implementación de sistemas.

El Sistema SIICOFE permite ahorrar los tiempos de trámites y procesos y ahora se hacen en poco tiempo. Y un valor agregado del mismo es que nos permite el ahorro de papel y la utilización de menos elementos humanos ya que el sistema reduce de 8 personas que realizaban esta labor a solo 3 personas. Esto permite reducir recursos en tiempos de Austeridad en base al programa de Austeridad del Gobierno Federal.

Con la elaboración de este sistema logramos reducir tiempos de operación de elaboración de reportes en un 75% así como un ahorro de papel del 40%. Esta implementación permite que los usuarios en automático impriman su información y se tenga la información actualizada al día del sistema.

Así las Tecnologías de Información permiten el aprovechar los recursos y utilizar el rendimiento de los equipos de cómputo y así mejorar los servicios que se proporcionan con los usuarios finales.

El Diplomado de creación de páginas Web me permitió establecer lineamientos en el desarrollo de esta aplicación e implementar el sistema de manera optima, clara y concisa para su desarrollo. En el Diplomado pude darme cuenta de los recursos necesarios para la implementación de una página Web desde su análisis hasta la fase de implementación.

Actualmente se tiene proyectada una 2ª fase de desarrollo para mejorar aún más los servicios de la COFEMER.

APÉNDICE 1

OBJETO	<p>Es la unidad atómica que encapsula un estado y comportamiento, este puede caracterizar una entidad física (un coche) o abstracta (ecuación matemática). Un OBJETO es algo que posee un:</p> <ul style="list-style-type: none">• Estado: El estado de un objeto lo constituyen todos los datos que encapsula en un momento determinado.• Comportamiento: la manera en que actúa y reacciona un objeto, esto en función de sus cambios de estado y el paso de mensajes.• Identidad: La idea es que los objetos no se definen por los valores de sus atributos en un momento determinado, un objeto tiene una existencia continua.
CLASE	<p>Es un conjunto de objetos que comparten una estructura y un comportamiento común, así mismos describe un conjunto de objetos que tienen un rol o roles equivalentes en un sistema. Una CLASE es un grupo de objetos con características comunes (atributos), comportamiento común (operaciones) y relaciones comunes con otros objetos (asociaciones y agregaciones).</p>
ABSTRACCIÓN	<p>Es la propiedad que nos permite representar las características esenciales de un objeto, permitiéndonos omitir las propiedades y acciones de un objeto que no son de nuestro interés y dejar sólo aquellas que nos interesan para una situación en particular.</p>
JERARQUÍA	<p>Es el concepto, que en conjunto con la propiedad de ABSTRACCIÓN, nos permite clasificar y ordenar a los objetos de dos maneras: De tipos (Generalización/Especialización). Jerarquía establecida de lo particular a lo general, conceptualmente se utiliza La frase "ES UN" para establecer la generalización. De Partes (Agregación). Jerarquía establecida de las partes hacia el "todo", conceptualmente se utiliza La frase "ES PARTE DE" para establecer la agregación</p>
HERENCIA	<p>Dentro del Paradigma Orientado a Objetos, es uno de los más significativos y de más trascendencia. Consiste en trasladar al contexto de objetos el más puro significado de la palabra "herencia", es decir el hecho de que una clase "padre" herede características, estructuras y comportamiento a todas las clases "hijas". En sentido estricto se rehúsan las clases para definir nuevas. La clase general o "padre" es llamada superclase y a la clase especializada o "hija" es llamada subclase. La superclase define un comportamiento (protocolo) que todas las subclases heredan Asimismo, las subclases pueden agregar nuevos atributos y operaciones Las subclases pueden sobre-escribir operaciones (conservando la firma pero proporcionando distintas implementaciones).</p>
POLIFORMISMO	<p>Es la propiedad por la cual dos o más clases responden al mismo mensaje cada uno de manera diferente, lo que permite realizar la misma operación de forma diferente en cada objeto. El polimorfismo es una característica de una clase que puede tomar varias formas: El comportamiento de una clase puede variar de acuerdo a las circunstancias. En ocasiones una operación tiene el mismo nombre en distintas clases. El polimorfismo en UML es la posibilidad de desencadenar operaciones distintas en respuesta a un mismo mensaje. Así mismo cada subclase hereda las operaciones pero tiene la posibilidad de modificar el comportamiento de estas operaciones, pero únicamente de manera interna a la clase. El resultado del código (o acción) que se va a ejecutar es el resultado de un enlace dinámico, el polimorfismo nos permite obtener un comportamiento y diseño coherente de nuestro sistema.</p>
ENCAPSULAMIENTO	<p>Esta propiedad en general, hace referencia a cualquier tipo de ocultamiento de una clase. Permite asegurar que el contenido de la información (atributos y métodos) de un objeto está oculta al mundo exterior, protegiéndola de esa forma y ocultando su implementación a los usuarios. La única forma de acceder a la información de un objeto y manipularla es a través de las operaciones. Debemos notar las ventajas de la encapsulación: Da seguridad a nuestros datos al protegerlos de accesos no autorizados. Disminuye el acoplamiento entre las clases lo que le da flexibilidad a nuestro sistema. Nos permite hacer módulos de nuestro sistema lo que facilita su mantenimiento</p>
MODULARIDAD	<p>La Modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas llamada módulos. Cada uno de estos módulos puede ser tan independiente como sea posible de la aplicación y de las otras partes.</p>
PERSISTENCIA	<p>Esta característica consiste en la conservación de la información en algún medio con la posibilidad de recuperarla en cualquier momento. Por lo que permite que los objetos existan fuera del programa en ejecución</p>

APÉNDICE 2.



Figura 1.6 Diagrama de estados de reporte de un servicio.

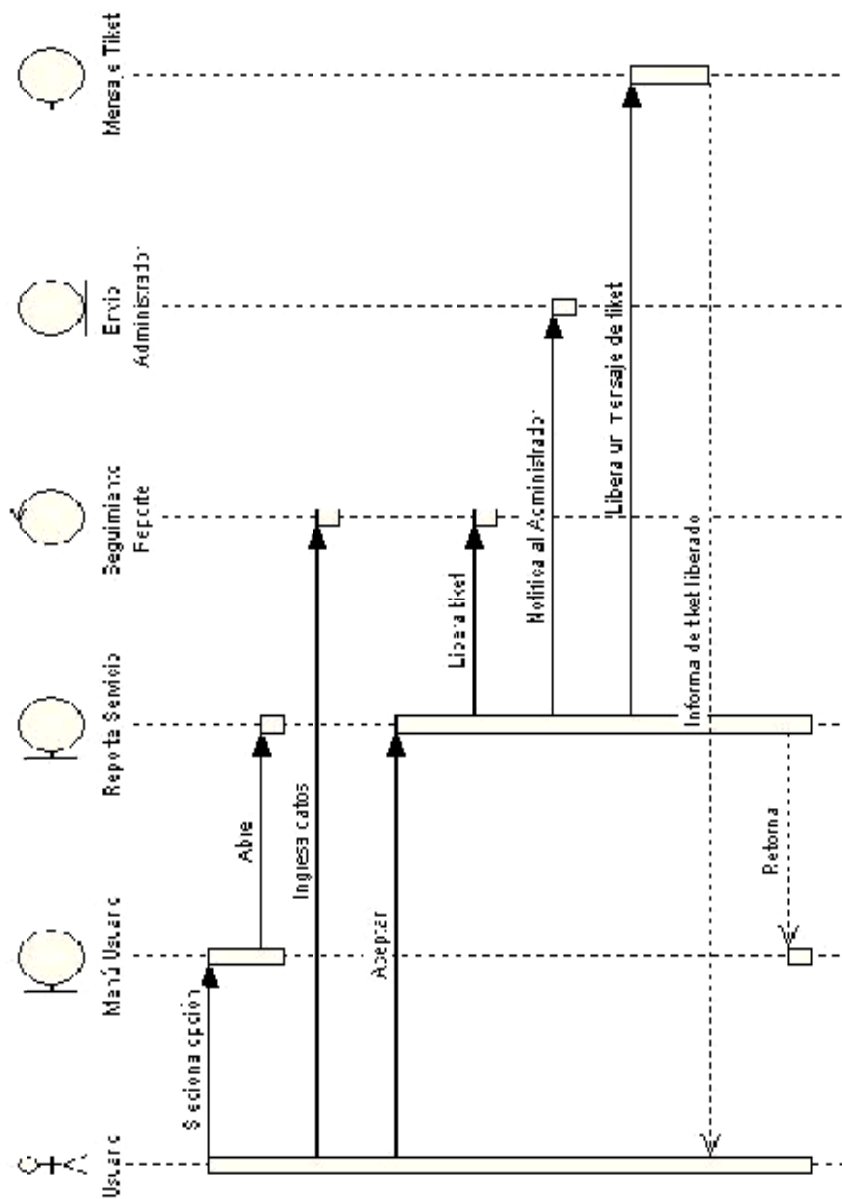


Fig. 1.7 Diagrama de secuencia para registrar un servicio.

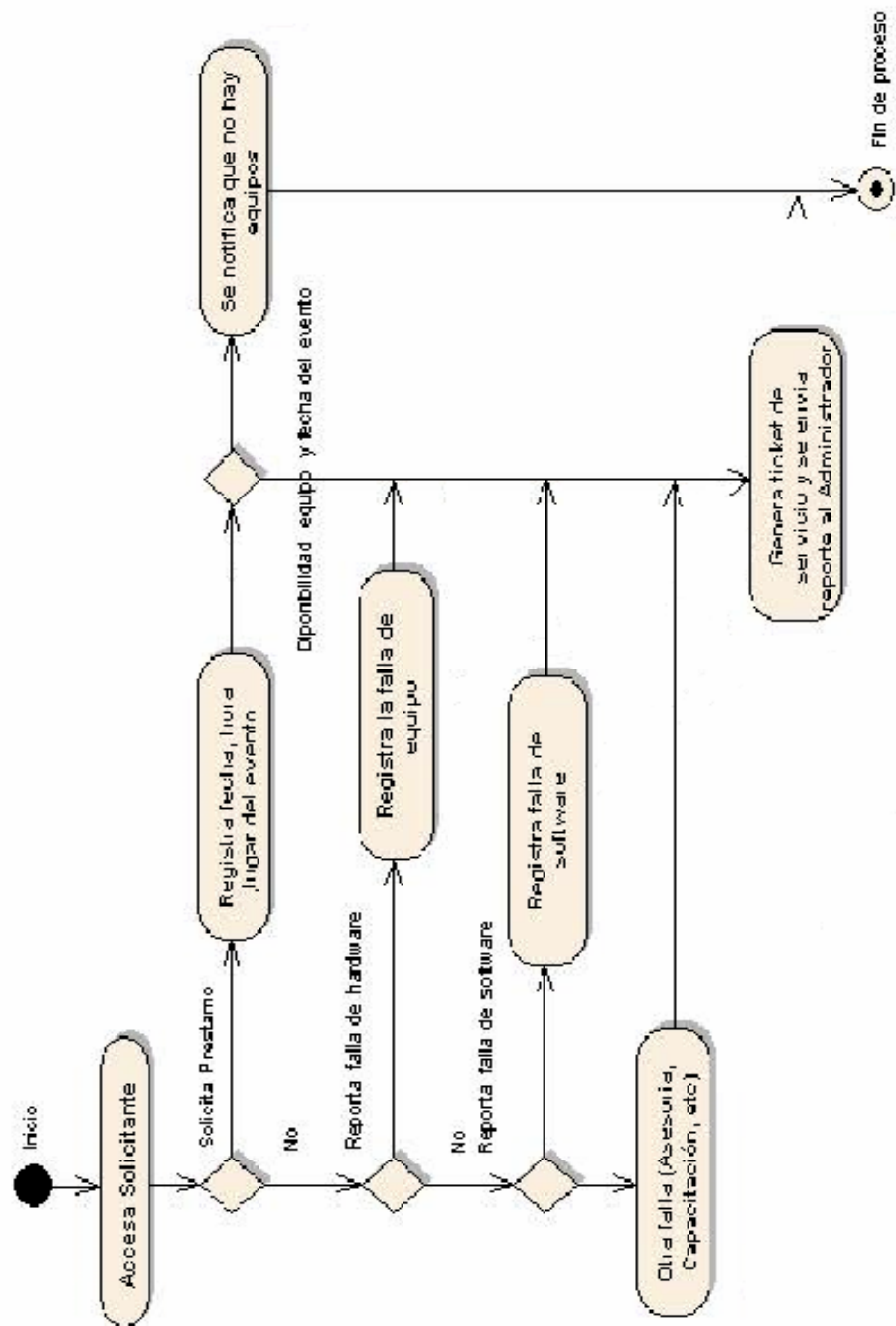


Fig. 1.8 Diagrama de actividades reporte de servicio.

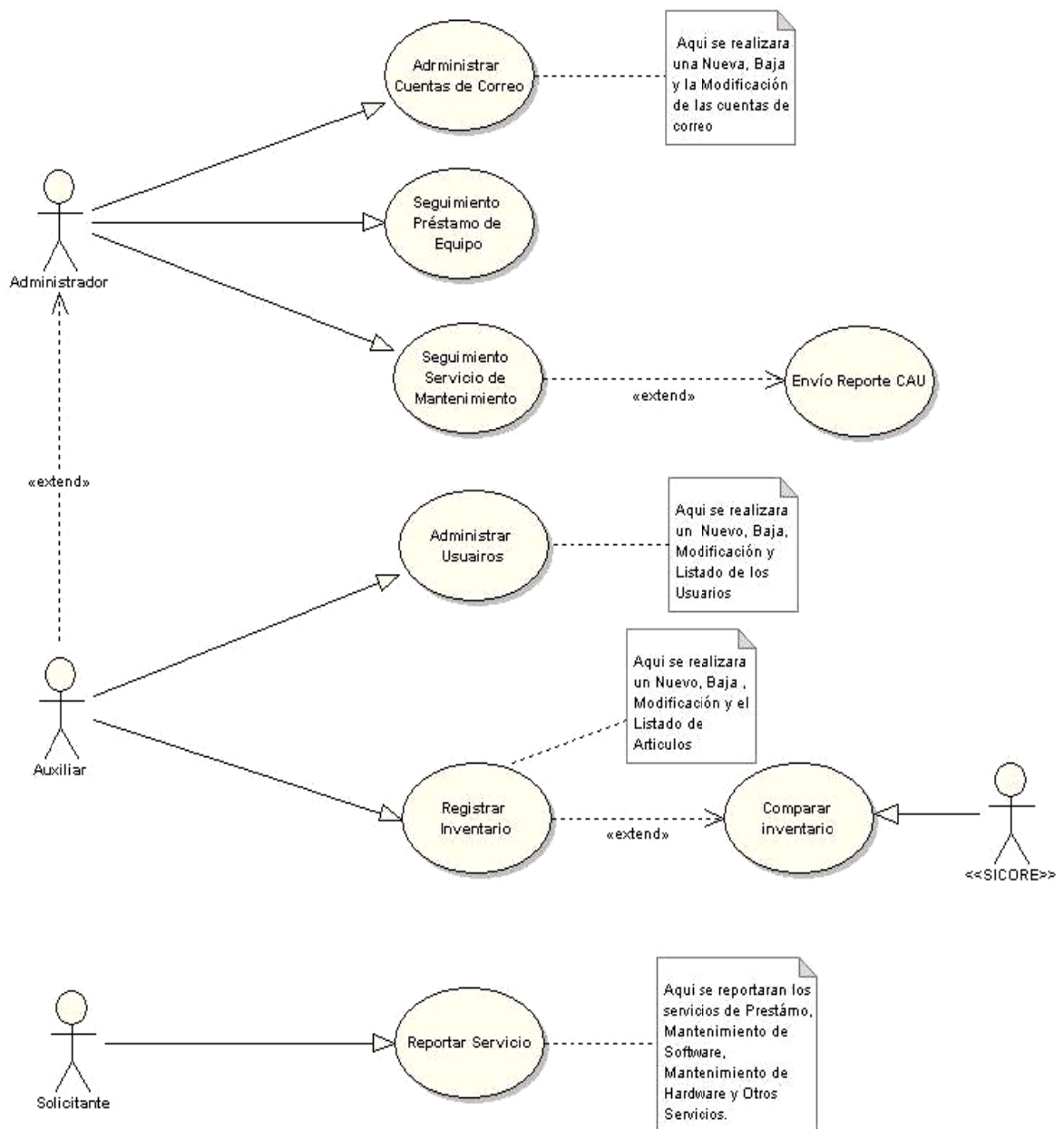
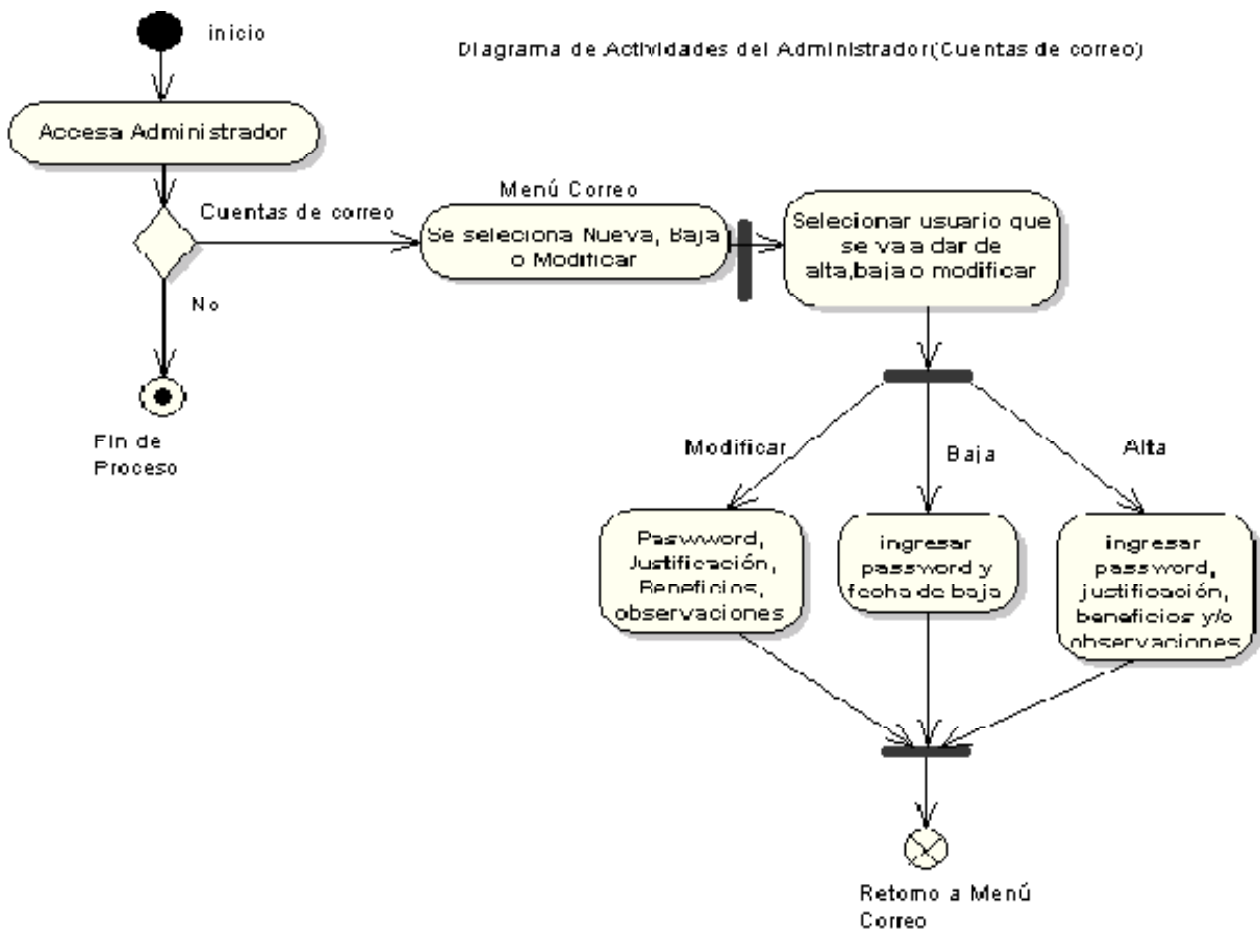


Figura 2.2 Diagrama de caso de uso sistema SICOFE.

Fig. 2.9 Diagrama de actividades de una cuenta de correo del sistema SIICOFFE.



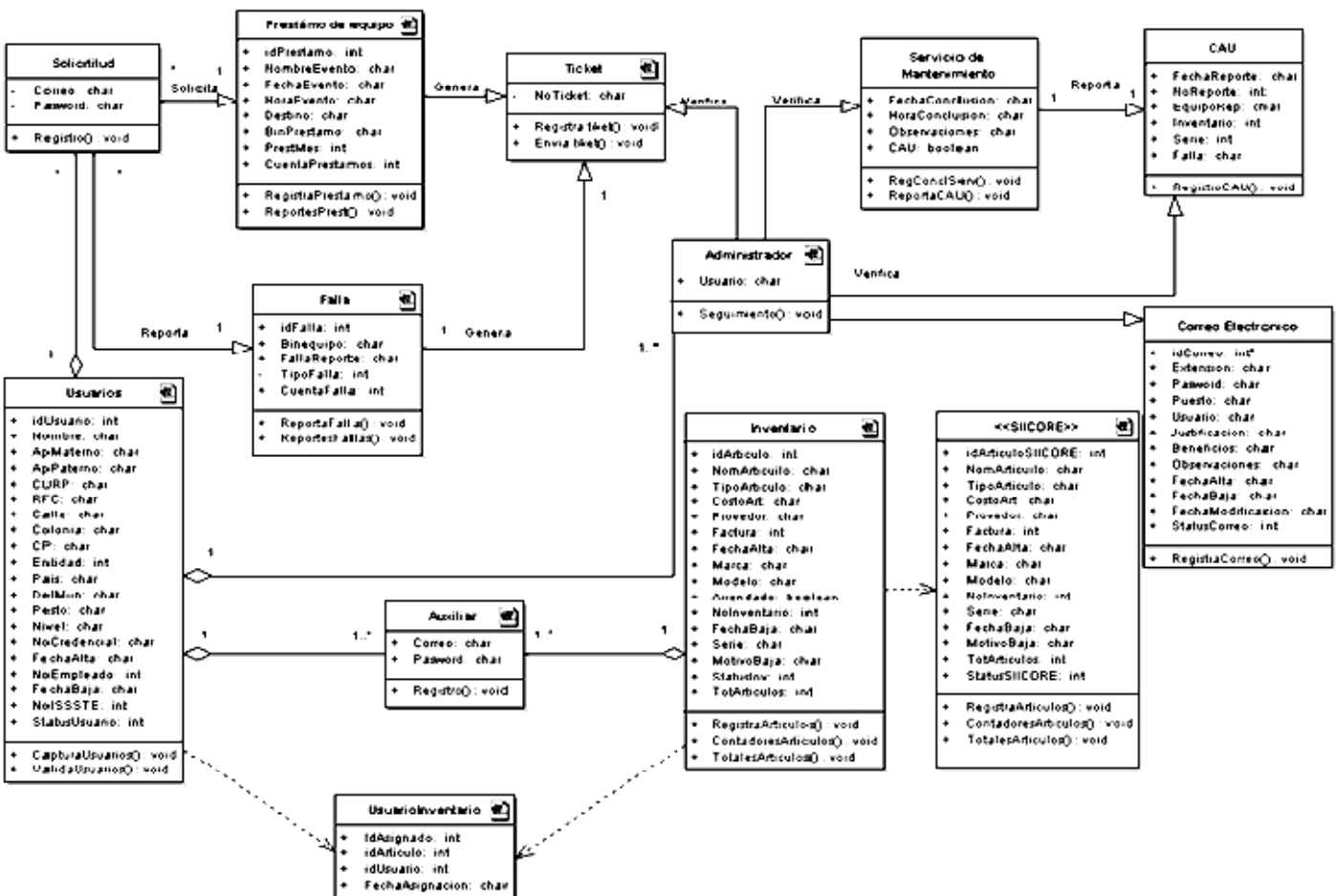


Figura 2.12 Diagrama de clases a detalle del sistema SIICOFE.

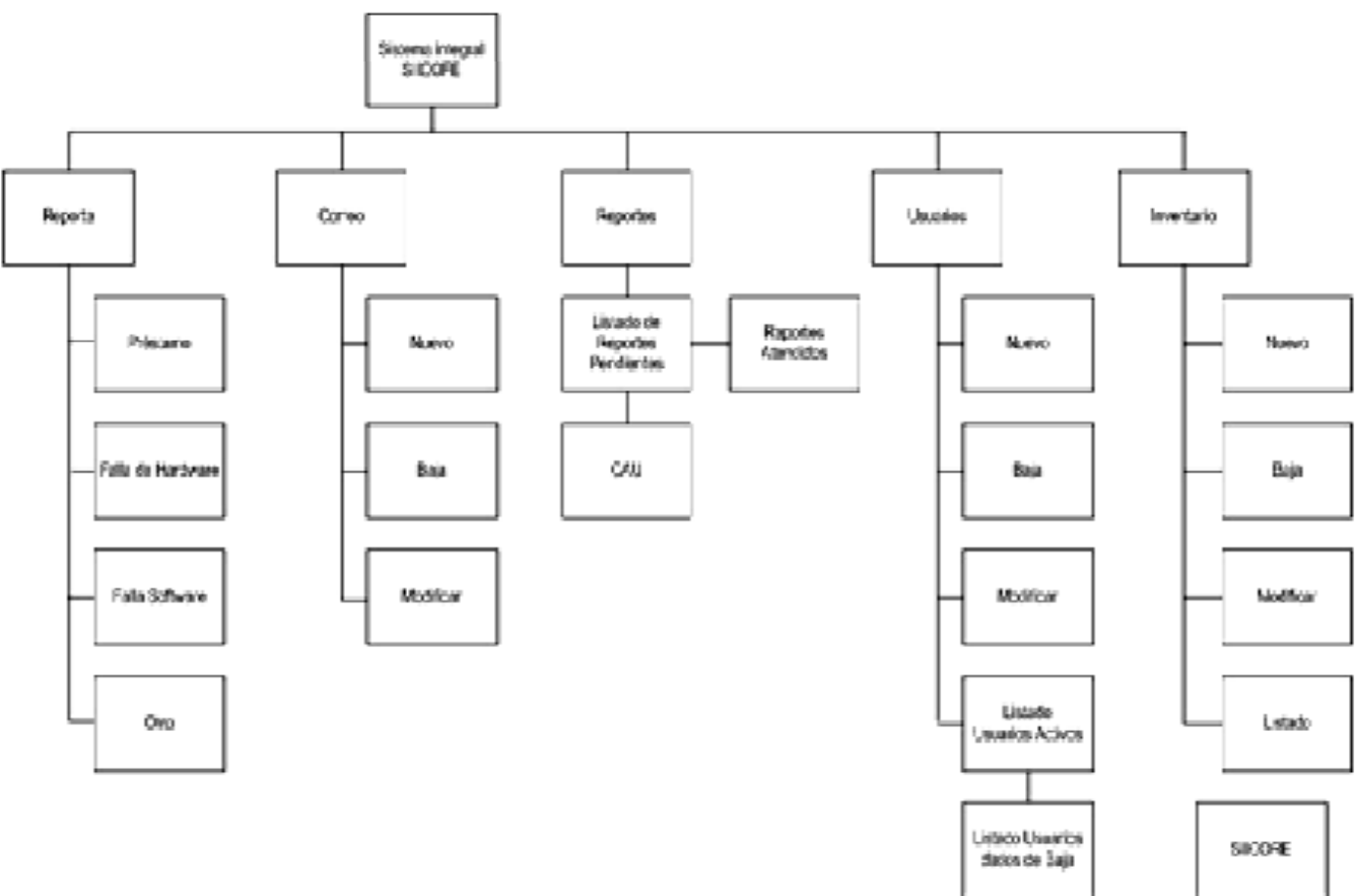


Figura 3.2 Árbol funcional del sistema SIICOFE.

BIBLIOGRAFÍA

Fowler, Martín
UML, gota a gota
Addison Wesley Longman de México, SA de CV
México 1.999
Paginas 224

Por Paola Miani
Consultora de Xperience Consulting
Junio2003

Guia para desarrollo de sitios Web 2004
Gobierno de Chile
Ministerio Secretaria General de Gobierno

Objetos, componentes y Estructuras con UML, The Catalysis Aproach, por Desmond F. D'Souza y Alan C. Wills, Addison Wesley Longman, 1998.

CSS 2.1 Specification — Cascading Style Sheets, level 2 revision 1, W3C Working Draft, 6 de noviembre de 2006.

HTML 4.01 Specification — W3C Recommendation, 24 de diciembre de 1999.
WCAG 1 — *Web Content Accessibility Guidelines* 1.0, W3C Recommendation, 5 de mayo de 1999.

Mobile *Web* Best Practices 1.0 — Basic Guidelines, W3C Proposed Recommendation, 2 de noviembre de 2006.

Scope of Mobile *Web* Best Practices — W3C Working Group Note, 20 de diciembre de 2005.

HTML 4.0 Guidelines for Mobile Access — W3C Note, 15 de marzo de 1999.

Proyecto Creative Commons México.

Diseño del sistema de información

<http://www.csi.map.es/csi/metrica3/dsiproc.pdf>

El ciclo de vida de un sistema de información

<http://elvex.ugr.es/idbis/db/docs/design/1-process.pdf>

Modelo Conceptual

<http://tvdi.det.uvigo.es/~avilas/UML/node12.html>

Entendiendo UML: La guía del desarrollador, con una aplicación java basada en web, por Paul Harmon y Mark Watson; Morgan Kauffman Publishers, Inc., 1998
(www.mkp.com/books_catalog/1-55860-465-0.asp).

- ¿Qué le falta a UML? un artículo por Scott Ambler, 'Objet Magacine', Octubre de 1997, SIGS Publications (www.sigs.com/omo/articles/ambler.html)

- Especificación UML 1.1 (Centro de Recursos de UML en www.popkin.com)

java

http://www.programacion.com/java/tutorial/servlets_jsp/