

# **CAPITULO I**

## **ASPECTOS GENERALES**

### **1.1 PRESENTACION**

El presente proyecto presenta el Análisis y Desarrollo del Sistema de Control Técnico y Administrativo para el Área de Informática del Ministerio del Ambiente (SISTEC).

El análisis y desarrollo del sistema se ampara en el paradigma espiral orientado a la Web.

La metodología utilizada es la OOHDM dividida en cinco fases consecutivas: Obtención de Requerimientos, Modelo Conceptual, Diseño Navegacional, Diseño de Interfaz Abstracta e Implementación.

La arquitectura que utilizada es la de tres capas, en la que el navegador (cliente) hace una petición al servidor Web, y este a su vez consulta en la base de datos para satisfacer la petición.

EL Back End utilizado en el desarrollo del sistema es MySQL, el Front End de desarrollo es PHP.

### **1.2 AMBITO**

La Unidad de Gestión de Recursos Tecnológicos (GRT) del Ministerio del Ambiente esta encargada de administrar, coordinar, y gestionar los recursos e infraestructura tecnológica, garantizando su adecuado funcionamiento y su alta disponibilidad para facilitar y apoyar el trabajo de los funcionarios.

Actualmente, en la unidad se lleva un control manual de la información en lo que se refiere al control de equipos de computación así como también el control de actividades realizadas por los técnicos del ministerio (Soporte Técnico).

### **1.2.1 SOPORTE TÉCNICO**

Cuando un funcionario tiene alguna dificultad con su equipo, realiza una llamada al departamento de Informática; en donde se asigna un técnico para que atienda ese requerimiento; quien acude al puesto de trabajo para la revisión del daño reportado por el funcionario.

Estas actividades de soporte, están divididas en tres grupos bien definidos:

- Asistencia Técnica
- Mantenimiento Interno
- Mantenimiento Externo

#### **1.2.1.1 Asistencia Técnica**

Es cuando el técnico acude al puesto de trabajo, pero existe un daño leve y la solución del problema la realiza en forma inmediata. En este caso no se genera un documento en el que se registre dicha actividad.

#### **1.2.1.2 Mantenimiento Interno**

Esta actividad es similar a la anterior, pero con la diferencia de que el daño no puede ser resuelto en el sitio de trabajo, y es necesario trasladar el equipo al departamento de Informática, en donde se soluciona el daño.

En este caso es necesario generar un documento en donde conste la actividad de mantenimiento realizada, el cual es firmado por el funcionario y por el técnico encargado.

### **1.2.1.3 Mantenimiento Externo**

Cuando el daño ocurre en equipos que aun tienen garantía, son llevados al proveedor para que se encargue de la revisión y reparación del equipo.

También es necesario el mantenimiento externo cuando el equipo no puede ser reparado dentro del ministerio; como por ejemplo daños en scanners, impresoras, UPS, etc. También es necesario generar un documento para registrar el seguimiento del equipo.

Cada cierto tiempo la Subsecretaria de Desarrollo Organizacional del ministerio, solicita al departamento de Informática reportes en los que consten las actividades de mantenimiento, los mismos que son realizados de forma manual.

## **1.2.2 TRANSFERENCIA DE EQUIPOS Y MATERIALES**

El departamento de Informática tiene como una de sus funciones gestionar los recursos e infraestructura tecnológica, para lo cual es necesario estar constantemente actualizando y mejorando los equipos de computación de los funcionarios. Por esta razón, son frecuentes la transferencia (asignación y descarga) de equipos entre los funcionarios del ministerio; estas trasferencias se las ha dividido en:

- Transferencia de Equipos
- Transferencia de Materiales

### **1.2.2.1 Transferencia de Equipos**

Cada funcionario tiene a su cargo un grupo de equipos de computación, los mismos que pueden ser transferidos entre los diferentes funcionarios. Cuando a un funcionario se le entrega equipos, se denomina Asignación de Equipos; mientras que si se le quita se denomina Descarga de Equipos.

Estas transferencias son registradas en un documento en donde consta el detalle de los equipos transferidos, el mismo que debe ser firmado por el funcionario y por el administrador del departamento.

#### **1.2.2.2 Transferencia de Materiales**

Con el uso diario de los equipos y con el avance en la tecnología, es necesario remplazar los componentes del CPU (materiales), ya sea por daño o por actualización. Por este motivo, constantemente se realizan la Asignación y Descarga de materiales (Transferencia), las mismas que son registradas en documentos en donde consta el detalle de los materiales transferidos.

Todos los documentos generados en la transferencia de equipos y materiales son archivados, para poder presentar informes de los equipos existentes en el ministerio cuando se realicen auditorias internas o externas.

### **1.3 PROBLEMA**

Actualmente en la Unidad de Gestión de Recursos Tecnológicos del Ministerio del Ambiente se lleva un control manual de la información en lo que se refiere actividades realizadas por los técnicos del ministerio, equipos y materiales de computación (stock), transferencia de equipos y materiales; lo que ha dificultado la obtención de reportes del inventario y las actividades realizadas diariamente.

Toda esta información es difícil de manejar a la vez que causa una pérdida de tiempo debido a que se encuentra archivada en una gran cantidad de documentos en carpeta y el momento de presentar informes se torna un proceso muy complicado.

A continuación detallamos los inconvenientes que se generan en el ministerio por el actual manejo de la información:

- No es posible la generación de reportes sobre el soporte técnico realizado por los técnicos de la Unidad.

- Es difícil tener un control eficiente del stock de repuestos y equipos de computación.
- Existen dificultades cuando los equipos son reasignados a diferentes funcionarios.
- No se cuenta con una base de datos que permita controlar la información de una manera adecuada.
- El proceso para acceder a la información existente es demasiado lento y conflictivo.
- No es posible conocer los inconvenientes más comunes que tienen los funcionarios con el manejo de sus equipos, para así poder brindarles capacitación.
- Cuando se realizan auditorias externas, la recolección de datos de los equipos es un proceso conflictivo y de mucha demora, ya que se lo debe realizar en cada puesto de trabajo interrumpiendo las actividades de los funcionarios.

#### **1.4 OBJETIVO GENERAL**

Mejorar los servicios administrativos que la Unidad de Recursos Tecnológicos del Ministerio del Ambiente ofrece a los usuarios mediante el desarrollo e implementación de un sistema automatizado que permita el control de las actividades e inventario de equipos y materiales de computación.

#### **1.5 OBJETIVOS ESPECIFICOS**

- Modelar, diseñar y seleccionar una base de datos que nos permita administrar toda la información.
- Crear un módulo que permita ingresar las actividades realizadas por los técnicos.

- Desarrollar un modulo para la generación de reportes de las actividades periódicas realizadas por los Técnicos.
- Crear un modulo que facilite y agilite la transferencia (asignación y descarga) de los equipos.
- Crear un modulo que facilite y agilite la transferencia (asignación y descarga) de los repuestos o materiales.
- Diseñar un módulo que permita controlar los equipos y repuestos (stock) pertenecientes a la Unidad.
- Generar reportes que nos permita conocer los inconvenientes más comunes funcionarios con el manejo de sus equipos para proporcionarles cursos de capacitación.
- Diseñar un módulo que nos permita hacer un seguimiento a los proveedores de servicios quienes se encargarán del soporte externo de los equipos.

## **1.6 JUSTIFICACION PRÁCTICA**

Con la elaboración de este sistema para la Unidad de Gestión de Recursos Tecnológicos del ministerio de Ambiente esperamos llenar todos los requerimientos necesarios para un mejor control de las actividades de soporte técnico, así como de equipos de computación e inventario de repuestos.

Actualmente este control se lleva de manera manual, lo cual conlleva a un proceso lento y desorganizado al momento de archivar la información

Todos estos beneficios brindarán al GRT una confiabilidad de su información y seguridad, para ofrecer un mejor servicio a sus usuarios, realizando todos estos procesos de manera rápida y segura.

## 1.7 ALCANCE

El sistema permitirá consultar información referente a las unidades, funcionarios, equipos, materiales que pertenecen al Ministerio, así como también la información sobre los proveedores que trabajan con el ministerio.

Llevará el control de las actividades de soporte que son realizadas por los técnicos del departamento de informática. Además permitirá la administración de las transferencias de equipos y materiales.

Permitirá generar reportes de los funcionarios y los equipos a su cargo y reportes de todos los equipos y materiales disponibles (stock). También se podrá generar reportes estadísticos en lo que se refiere a los tres tipos de actividades de soporte.

Además permitirá hacer un respaldo de la Base de Datos, como una medida de seguridad para resguardar la integridad de la información.

El manejo del sistema será diseñado para soportar diferentes tipos de usuarios, los cuales podrán realizar actividades dependiendo de los permisos que les sean asignados.

La realización y desarrollo de este sistema brindará beneficios tales como:

- Llevar un control ordenado de la información lo cual nos permitirá acceder a la misma de una manera más rápida.
- Agilizar el proceso de ingreso de información sobre el soporte técnico brindado a los funcionarios del ministerio.
- Disminuir el tiempo requerido para la presentación de informes sobre las actividades realizadas por los técnicos.
- Conocer las falencias de los funcionarios en el manejo del software, para de esta manera gestionar cursos de capacitación.
- Generar informes sobre los equipos y materiales (stock) con los que cuenta la unidad de gestión de recursos tecnológicos.
- Tener un control adecuado sobre la transferencia de equipos y materiales, lo cual nos permitirá conocer lo que está a cargo de cada uno de los funcionarios.

## 1.8 LIMITACIONES

SISTEC no maneja información de equipos de oficina, únicamente equipos de computación.

SISTEC no maneja información del software utilizado en el ministerio.

SISTEC no generará reportes de técnico por actividad realizada.

SISTEC no generará reportes de funcionarios por unidad.

## 1.9 ANÁLISIS DE COSTOS

### Recursos Humanos

Personal	Total Horas (6 meses)	Costo/Hora	Costo
Analista	360	6.00	2160.00
Desarrollador	360	5.00	1800.00
TOTAL			3960.00

### Hardware

Descripción	Costo
Computador PIV	600.00
Impresora	60.00
Total	660.00



## Software

Descripción	Costo
Macromedia: Dreamweaver Flash Mx Fireworks	1450.00
Apache, MySQL, PHP, PHPMyadmin.	Software Libre
<b>Total</b>	<b>1450.00</b>

## Otros

Descripción	Costo
Internet	40.00
Suministros de Oficina	50.00
<b>Total</b>	<b>90.00</b>

## Costo Total

Detalle	Costo
Recursos Humanos	3960,00
Hardware	660.00
Software	1450,00
Otros	400,00
<b>Total</b>	<b>6470,00</b>

## **CAPITULO II**

### **ASPECTOS TEÓRICOS**

#### **2.1 SISTEMAS Y APLICACIONES WEB (WEBAPPS)**

Las Aplicaciones Web (WebApps) son aquellas en las que los usuarios acceden a ellas mediante un servidor Web, ya sea a través de Internet o de una Intranet.

A inicio de la década de los años 90, cuando se abre paso al concepto del uso de Internet, la estrategia del uso de la WWW con fines de comercializar sus productos se inicia con el lanzamiento de sitios Web compuestos por páginas planas (estáticas), que representaban las exigencias y disponibilidades tecnológicas en aquellos momentos.

La rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que la información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior; además esta información debe ser compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa o a aquella parte que les corresponda en cada momento.

En la actualidad, las novedosas herramientas para desarrollo de sitios Web han sido consideradas como una alternativa valiosa para el desarrollo y publicación de sitios más complejos y dinámicos; y los sitios web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción sofisticada con el usuario.

Actualmente las WebApps cuentan con una gran popularidad debido a la facilidad para actualizarlas y mantenerlas, ya que si se desea hacer una mejora en una aplicación tradicional es necesario mejorar tanto el servidor como el cliente, lo que no sucede en las WebApps; lo que significa un gran ahorro de recursos. Además con las WebApps no existe la necesidad de distribuir o instalar software en la parte del cliente, como es necesario en una aplicación cliente-servidor tradicional.

## **2.2 INGENIERÍA WEB (IWEB)**

Las WebApps se integran cada vez más en grandes y pequeñas empresas, y cada vez es más importante la necesidad de construir sistemas fiables, utilizables y adaptables. Por esta razón es primordial el empleo de un enfoque disciplinado que nos permitan el desarrollo empleo y evaluación de sistemas y aplicaciones basados en Web. De esta manera nace el concepto de Ingeniería Web que se la puede definir como:

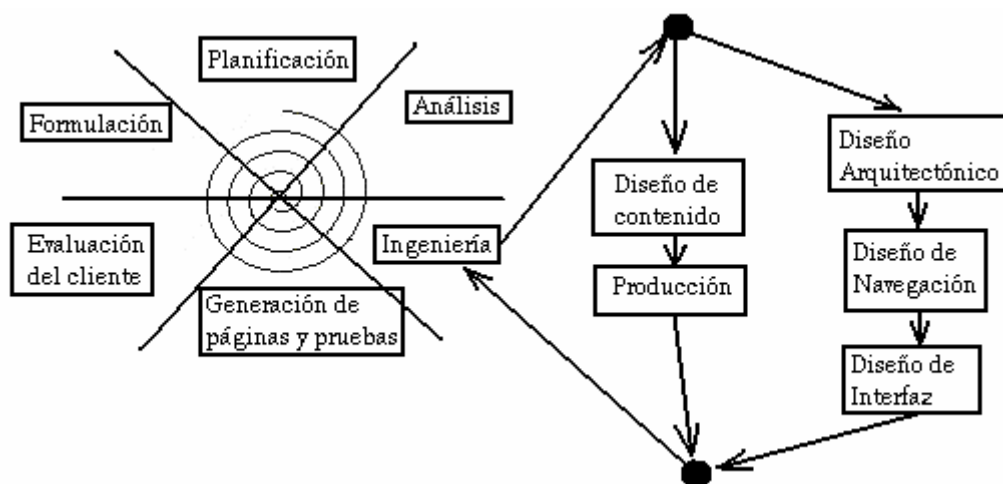
“Es el establecimiento y utilización de principios científicos, de ingeniería y de gestión, con enfoques sistemáticos y disciplinados del éxito del desarrollo, empleo y mantenimiento de sistemas y aplicaciones basados en Web de alta calidad.

O de manera más sencilla se puede decir que Ingeniería Web “Es el proceso para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.”

La ingeniería Web se diferencia de la ingeniería de software, pero toma prestados varios conceptos y principios básicos de software convencional, dando importancia a las mismas actividades técnicas y de gestión. Y aunque existen diferencias sutiles en la manera en que realizan las actividades, tienen una misma filosofía primordial, ya que se dicta un enfoque disciplinado para el desarrollo de un sistema basado en computadora.

### 2.3 PARADIGMA ESPIRAL ORIENTADO A LA WEB

A medida que la evolución de las WebApps pasa de utilizar recursos estáticos de información controlada por el contenido a utilizar entornos de aplicaciones dinámicos controlados por el usuario, cada vez es más importante la necesidad de aplicar una gestión sólida y unos principios de ingeniería. Para conseguir esto, es necesario desarrollar un marco de trabajo IWeb que acompañe a un modelo de proceso eficaz, popularizado por las actividades del marco de trabajo y por las tareas de ingeniería.



El proceso IWeb comienza con la formulación, mediante esta actividad se identifica las metas y los objetivos de la WebApp y se establece el ámbito del primer incremento. La planificación estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo para el incremento final de la WebApp.

El análisis establece los requisitos técnicos para la WebApp e identifica los elementos del contenido que se van incorporar. Además se definen los requisitos del diseño gráfico.

La actividad de ingeniería incorpora dos tareas paralelas, el diseño del contenido y la producción son llevadas a cabo por dos personas no técnicas del equipo IWeb. El objetivo de estas tareas es diseñar, producir y adquirir todo el contenido

del texto, gráfico y video que se vayan a integrar en la WebApp. Al mismo tiempo se llevan a cabo un conjunto de tareas de diseño.

La generación de páginas es una actividad de construcción que hace mucho uso de las herramientas de automatizadas para la creación de la WebApp. El contenido definido en la actividad de ingeniería se fusiona con los diseños arquitectónicos, de navegación y de la interfaz para elaborar páginas Web ejecutables en HTML, XML y otros lenguajes orientados a procesos.

Las pruebas ejercitan la navegación, intentan descubrir los errores de las applets, guiones y formularios; y ayuda a asegurar que la WebApp funcionará correctamente en diferentes entornos, como por ejemplo en diferentes navegadores.

Durante la evaluación del cliente se revisa cada incremento producido como parte del proceso IWeb. En este punto se pueden solicitar cambios, los cuales se integran en la siguiente ruta mediante el flujo incremental del proceso.

### **2.3.1 FORMULACIÓN**

Se sugiere una serie de preguntas que se deben formular y responder al comienzo de la etapa de formulación:

¿Cuál es la motivación principal para la WebApp?

¿Por qué es necesaria la WebApp?

¿Quién va a utilizar la WebApp?

Las respuestas que se han establecido anteriormente implican metas específicas para la aplicación Web. Una vez identificadas las metas se desarrolla el perfil del cliente. El perfil del usuario recoge las características relevantes de los usuarios potenciales incluyendo antecedentes, conocimiento, y preferencias.

Una vez desarrolladas las metas y los perfiles de usuario, la actividad de formulación se centra en la afirmación del ámbito para la WebApp.

## **2.3.2 ANÁLISIS**

Para crear un modelo de análisis completo para la WebApp se elabora el ámbito definido durante la actividad de formulación. Durante la IWeb se realizan cuatro tipos de actividades diferentes:

- Análisis del contenido
- Análisis de la interacción
- Análisis funcional
- Análisis de la configuración

### **2.3.2.1 Análisis del contenido**

Se trata de la identificación del espectro completo del contenido que se va a proporcionar. En el contenido se incluyen datos de texto, gráficos, imágenes, video y sonido. Para identificar y describir cada uno de los objetos de los datos que se a utilizar dentro de la WebApp.

### **2.3.2.2 Análisis de la interacción**

Se trata de la descripción detallada de la interacción del usuario y la WebApp. Para proporcionar descripciones detalladas de esta interacción se pueden desarrollar casos prácticos.

### **2.3.2.3 Análisis funcional**

Los escenarios de utilización (casos de uso) creados como parte del análisis de interacción definen las operaciones que se aplicaran en el contenido de la WebApp e implicaran otras funciones de procesamiento. Aquí se realiza una descripción detallada de todas las funciones y operaciones.

#### **2.3.2.4 Análisis de la configuración**

Se efectúa una descripción detallada del entorno y la infraestructura en donde reside la WebApp. La WebApp puede residir en Internet, en una Intranet o una extranet. Además se deberá identificar la infraestructura de la WebApp, es decir de los componentes y el grado de utilización de la base de datos para generar el contenido.

### **2.3.3 DISEÑO**

La naturaleza de las aplicaciones basadas en el Web unida a la presión de evolucionar continuamente obliga que un ingeniero establezca un diseño que resuelva el problema comercial inmediato, mientras que al mismo tiempo obliga a definir una arquitectura de aplicación que tenga la habilidad de evolucionar rápidamente con el tiempo. El problema desde luego es que resolver el problema inmediato puede dar como resultado compromisos que afecten a la habilidad que tiene la aplicación de evolucionar con el paso del tiempo.

#### **2.3.3.1 Principios y Métodos de Diseño**

Es importante destacar que los conceptos y principios del diseño se aplican a todas las WebApp. La modularidad y eficaz, la elaboración paso a paso, y cualquier otra heurística del diseño de software conducirá a sistemas y aplicaciones basadas en Web mas fáciles de adaptar, mejorar, probar, y utilizar.

Cuando se crea la aplicación de Web se puede reutilizar los métodos de diseño que se utilizan para sistemas orientados a objetos. La hipermedia define objetos que interactúan mediante un protocolo de comunicación algo similar a la mensajería. De hecho, la notación de diagramas propuesta por UML puede adaptarse y utilizarse mediante las actividades de diseño de las WebApp.

Las aplicaciones hipermedia interactivas lleva construyéndose ya hace una década, durante este tiempo los diseñadores han desarrollado un conjunto de heurísticas de diseño que se podrá volver ha aplicar durante el diseño de aplicaciones nuevas.

Configuraciones de Diseño.- Se puede aplicar no solo a elementos funcionales de una aplicación sino también a documentos gráficos y estética general de un sitio Web ya que son un enfoque genérico para resolver pequeños problemas que se pueden adaptar en una variedad más amplia de problemas específicos.

Plantillas.- Se pueden utilizar para proporcionar un marco de trabajo esquemático en cualquier configuración de diseño o documentos utilizado dentro de una WebApp.

#### **2.3.3.2 Diseño Arquitectónico**

El diseño arquitectónico para los sistemas y aplicaciones Web se centra en la definición de la estructura global hipermedia para la WebApp y en la aplicación de las configuraciones de diseño y plantillas constructivas para popularizar la estructura y lograr la reutilización. Una actividad paralela llamada diseño de contenido deriva la estructura y el formato detallados del contenido de la información que se presentara como parte de la WebApp.

#### **2.3.3.3 Diseño de Navegación**

Una vez establecida una arquitectura de WebApp e identificados los componentes de la arquitectura, el diseñador deberá definir las rutas de navegación que permitan al usuario acceder al contenido y a los servicios de la WebApp. Para que el diseñador pueda llevar a cabo debe primeramente identificar la semántica de la navegación para diferentes usuarios del sitio; para luego definir la sintaxis para lograr la navegación.



#### 2.3.3.4 Diseño de Interfaz

Los conceptos, principios y métodos de diseño de interfaz comúnmente utilizados son también aplicables para las WebApp. Sin embargo, las características especiales de los sistemas y aplicaciones Web requieren otras consideraciones adicionales.

A continuación se presentan algunas sugerencias para el rediseño de una WebApp:

Probabilidad de que los errores del servidor, incluso que los más pequeños abandonen el sitio Web.

La velocidad de lectura del monitor de una computadora es aproximadamente es un 25% más lento que leer una copia impresa, por lo que, no hay que obligar al usuario a leer cantidades voluminosas de texto.

Evite los símbolos bajo construcción que son los que levantan expectación y provocan un enlace innecesario.

Los usuarios previeren no tener que recorrer la pantalla. Dentro de las dimensiones normales de una pantalla se deberá incluir información importante.

Los menús de navegación y las barras de cabecera deberán diseñar consecuentemente y deberán estar disponibles en todas las páginas que el usuario tenga acceso. El diseño no deberá depender de las funciones del navegador para ayudar en la navegación.

La estética nunca deberá sustituir la funcionalidad. Por ejemplo; un botón sencillo podrá ser una opción de navegación mejor que una imagen o icono estéticamente agradables, pero vagos cuya intención no es muy clara.

Las opciones de navegación deberán ser obvias, incluso para el usuario casual. El usuario deberá utilizar la pantalla para terminar como enlazar con otro contenido de servicio.

Una interfaz bien diseñada mejora la percepción del contenido o de los servicios del usuario que proporciona el sitio Web.

#### **2.3.4 PRUEBAS**

Las pruebas constituyen el proceso de ejercitar el software con la intención de encontrar y corregir los errores.

Un enfoque de las pruebas de las WebApp adopta los principios básicos de todas las pruebas del software y aplica estrategias y tácticas que ya han sido recomendados para los sistemas orientados a objetos.

Los pasos a seguir son los siguientes:

- El modelo de contenido de la WebApp es revisado para descubrir errores.
- El modelo de diseño para la WebApp es revisado para descubrir errores de navegación.
- Se aplican pruebas de unidad a los componentes de proceso seleccionados y las paginas Web.
- Se construye la arquitectura, se realizan las pruebas de integración.
- La WebApp ensamblada se prueba para conseguir una funcionalidad global y un contenido.
- La WebApp se implementa se implementa en una variedad de configuraciones diferentes de entornos y comprobar así la compatibilidad con cada configuración.
- La WebApp se comprueba con una población de usuarios finales controlada y monitorizada. (Pressman Roger, 2002).

## 2.4 METODOLOGIA OOHDM

Ciencia que estudia los métodos utilizados por el ser humano para encontrar soluciones óptimas a problemas teóricos o prácticos. En relación con proyectos o informes científicos o técnicos, para describir los métodos, procedimientos e instrumentos que se han utilizado o se utilizarán para lograr los objetivos propuestos.

OOHDM (Método de Diseño Hipermedia Orientado a Objetos) es una metodología propuesta por Gustavo Rossi y Daniel Schwabe. Tiene por objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia.

Las metodologías tradicionales de ingeniería de software, o las metodologías para sistemas de desarrollo de información, no contienen una buena abstracción capaz de facilitar la tarea de especificar aplicaciones hipermedia. El tamaño, la complejidad y el número de aplicaciones crecen en forma acelerada en la actualidad, por lo cual una metodología de diseño sistemática es necesaria para disminuir la complejidad y admitir evolución y reusabilidad.

En primer lugar, la navegación posee algunos problemas. Una estructura de navegación robusta es una de las claves del éxito en las aplicaciones hipermedia. Si el usuario entiende dónde puede ir y cómo llegar al lugar deseado, es una buena señal de que la aplicación ha sido bien diseñada. (Silva – Mercerat ,2002)

OOHDM propone el desarrollo de aplicaciones hipermedia a través de un proceso compuesto por cinco fases o etapas: Obtención de Requerimientos, Modelo Conceptual, Diseño Navegacional, Diseño de Interfaz Abstracta e Implementación.



### **2.4.1 FASE 1: OBTENCIÓN DE REQUERIMIENTOS**

Como en todo proyecto informático la obtención de requerimientos es una de las etapas más importantes, la mayoría de los estudios entregan resultados claros que los errores más caros son los que se cometen en esta etapa.

En esta etapa el analista deberá introducirse cuidadosamente en el dominio del sistema, ahora su principal labor será identificar los diferentes roles que podrían cumplir cada uno de los potenciales usuarios de la aplicación.

Los usuarios juegan roles importantes en cada intercambio de información con el sistema es muy importante tener identificado el rol de cada usuario, ya que serán ellos los que entregarán su conformidad con respecto al proceso en el que participan.

Luego para cada rol el analista deberá identificar las tareas que deberá soportar la aplicación.

Se debe definir escenarios que son descripciones narrativas de cómo la aplicación será utilizada. Cada usuario deberá especificar textual o verbalmente los escenarios que describen su tarea.

Se representa la interacción entre el usuario y el sistema, agrupando las tareas representadas en los escenarios existentes. Es muy importante que el analista identifique cual es la información relevante en cada uno de los procesos.

(Universidad Católica de Valparaíso, Chile 2003)

### **2.4.2 FASE 2: DISEÑO CONCEPTUAL O ANÁLISIS DE DOMINIO**

Se construye un esquema conceptual el cual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos (desde simples cadenas de caracteres a gráficos, imágenes, texto, sonido, etc.) para representar perspectivas diferentes de las mismas entidades del mundo real.

El esquema de las clases consiste en un conjunto de clases conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones que son usadas para construir enlaces. (Silva – Mercerat ,2002)

### **2.4.3 FASE 3: DISEÑO NAVEGACIONAL**

En esta fase se define la estructura de navegación a través del hiperdocumento mediante la realización de modelos navegacionales que representen diferentes vistas del esquema conceptual de la fase anterior. Se trata, en definitiva, de reorganizar la información para adaptarla a las necesidades de los usuarios del sistema. El Diseño Navegacional se expresa, también con un enfoque orientado a objetos, a través de dos tipos de esquemas o modelos:

- El denominado esquema de clases navegacionales, con las posibles vistas del hiperdocumento a través de unos tipos predefinidos de clases, clases navegacionales tales como nodos, enlaces y estructuras de acceso (índices y visitas guiadas) inducidas del esquema conceptual. Los enlaces derivan de las relaciones y los nodos representan ventanas lógicas (views) sobre las clases conceptuales
- El esquema de contexto navegacional, que permite la estructuración del hiperespacio de navegación en subespacios para los que se indica la información que será mostrada al usuario y los enlaces que estarán disponibles cuando se acceda a un objeto (nodo) en un contexto determinado. (Universidad de Alcalá de Henares, 1998)

### **2.4.4 FASE 4 DISEÑO DE INTERFAZ ABSTRACTA**

Se realiza un modelo, también orientado a objetos, para especificar la estructura y el comportamiento de la interfaz del sistema hipermedia con el usuario Este modelo es abstracto y, por tanto, independiente de la implementación final del sistema. (Universidad de Alcalá de Henares, 1998)

Una vez que las estructuras navegacionales son definidas, se deben especificar los aspectos de interfaz. Esto significa definir la forma en la cual los objetos navegacionales pueden aparecer, cómo los objetos de interfaz activarán la navegación y el resto de la funcionalidad de la aplicación, qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia.

El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación. (Silva – Mercerat ,2002)

#### **2.4.5 FASE 5: IMPLEMENTACIÓN**

Hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase se debe tomar en cuenta el entorno particular en el cual se va a correr la aplicación.

Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada. (Silva – Mercerat, 2002)

#### **2.4.6 VENTAJAS DE OOHDM**

- OOHDM posee una notación diagramática bastante completa, que permite representar en forma precisa elementos propios de las aplicaciones hipermedia, tales como nodos, anclas, vínculos, imágenes, estructuras de acceso y contextos.

- En cada etapa de la metodología, especialmente en las de análisis y diseño, el usuario es considerado un integrante fundamental en la validación del producto obtenido. Esta interacción ayuda al desarrollador a entender y lograr en cada etapa lo que el usuario realmente necesita
- OOHDM genera una cantidad considerable de documentación a través de sus distintas etapas de desarrollo, lo que permite llevar un control del desarrollo de las etapas y tener la posibilidad real de realizar una rápida detección, corrección de errores y mantención.
- OOHDM ofrece la posibilidad de crear estructuras de reuso, tales como los frameworks, cuyo principal objetivo es simplificar las tareas de diseño y disminuir su consumo de recursos.
- OOHDM utiliza una herramienta diagramática, la cual es muy útil y sencilla de usar. Este instrumento es capaz de representar en forma precisa y con claridad los casos de uso obtenidos. (Universidad Católica de Valparaíso, 2003).

## **2.5 UML (Lenguaje de Modelado Unificado)**

Los sistemas, ya sean naturales o artificiales, son difíciles de comprender. Esto es debido a que en general son bastante complejos sin embargo, es posible representar y entender un sistema usando la técnica del modelado, esto es, construyendo modelos del sistema.

Un modelo es una simplificación de la realidad creada para comprender mejor un sistema; un modelo es una abstracción que captura la parte esencial de los sistemas a un determinado nivel de detalle.

Los modelos:

- Ayudan a visualizar cómo es o queremos que sea un sistema.
- Permiten especificar la estructura o componentes del sistema.
- Son una guía para la construcción y mantenimiento de los sistemas.
- Permiten experimentar múltiples soluciones.
- Reducen los riesgos por errores.
- Documentan las decisiones adoptadas.

UML es un lenguaje gráfico que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

UML utiliza modelos orientados a objetos. Un modelo orientado a objetos es una representación de un sistema a partir de los objetos u entidades que lo constituyen, con unos atributos y operaciones asociados, que interactúan con otros objetos para conseguir conjuntamente satisfacer los objetivos del sistema.

Se debe recalcar que UML no es una metodología, aunque proporciona técnicas que pueden ser usadas en conjunto o parcialmente en metodologías, fundamentalmente aquellas destinadas al desarrollo orientados a objetos.

Los diagramas se utilizan para visualizar un sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema. Representan una vista resumida de los elementos que constituyen un sistema. UML tiene nueve diagramas fundamentales, agrupados en dos grandes grupos, uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico. (Universidad de Alcalá, 2002).

Los diagramas estáticos; modelan la estructura y son: el de clases, de objetos, de componentes y de despliegue.

### **2.5.1 DIAGRAMA DE CLASES**

Los diagramas de clases son una potente herramienta de diseño, ayuda a los desarrolladores a planificar y establecer la arquitectura y estructura del sistema y subsistemas antes de escribir ningún código. Esto permite asegurar que el sistema está bien diseñado desde el principio. Los diagramas de clases son usados prácticamente en la totalidad de sistemas en que se utiliza UML para su modelado.

Un Diagrama de Clases está compuesto por los siguientes elementos: Clases y Relaciones



### 2.5.1.1 Clase

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase).

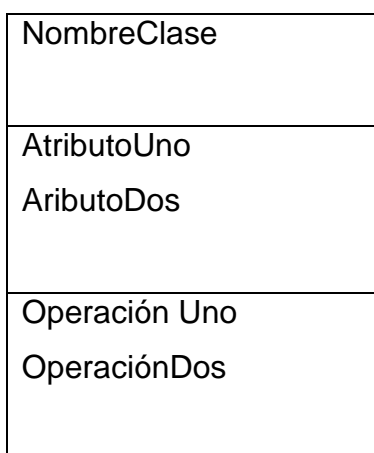
### 2.5.1.2 Representación Gráfica

Una clase es representada mediante un rectángulo que tiene tres divisiones internas llamadas compartimientos en los cuales se indica:

Comportamiento Superior: Contiene el nombre de la Clase.

Comportamiento Intermedio: Contiene los atributos que caracterizan a la Clase.

Comportamiento Inferior: Contiene las operaciones, las cuales son la forma como interactúa un objeto de la clase con su entorno.



Adicionalmente, podemos colocar otros compartimientos en los cuales se pueden describir, en texto libre, otras características de las clases como pueden ser sus responsabilidades.

Tanto la sección de los atributos como la de las operaciones pueden no mostrarse, en cuyo caso tampoco se mostrará la línea que las separa. Tampoco es necesario visualizar todos y cada uno de los atributos y operaciones, lo cual

puede ser útil para remarcar qué atributos u operaciones son interesantes en una determinada vista.

### 2.5.1.3 Relaciones

Las clases casi nunca se encuentran aisladas. Por lo general la mayoría de ellas colaboran con otras de varias maneras. Por tanto, al modelar un sistema también hay que modelar la forma en que las clases se interrelacionan para cumplir un objetivo común.

Existen tres tipos de relaciones entre las clases, las cuales son:

- Relación de Dependencia
- Relación de Generalización
- Relación de Asociación

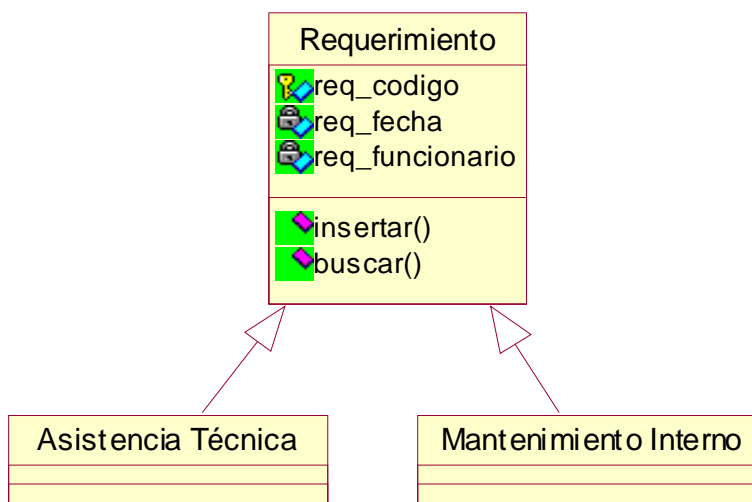
Relación de Dependencia.- Es un tipo de relación unidireccional, ya que el elemento dependiente debe conocer al independiente, pero el independiente desconoce la existencia del elemento dependiente. Gráficamente se muestra como una línea discontinua acabada en flecha que va del elemento dependiente al independiente (esto es, el elemento dependiente señala al elemento del que depende). Por ejemplo:



En este caso, la clase Funcionario (dependiente) dependerá de la clase Unidad (independiente), y un cambio en la clase Unidad puede ocasionar que la clase Funcionario necesite ser adaptada. Un cambio en la clase Funcionario no tiene ningún efecto sobre la clase Unidad, ya que esta última es independiente de la primera (no la necesita).

Relación de Generalización (Herencia).- Es una relación entre dos clases en donde una de ellas, llamada subclase o clase hija, hereda los atributos y el comportamiento de otra, llamada superclase o clase padre.

La generalización se utiliza para modelar la herencia en los lenguajes orientados a objetos. Una de las características de la herencia es que permite simplificar la construcción de clases relacionadas, ya que gracias a ella es posible agrupar las características comunes de un conjunto de clases en una clase padre (superclase) y hacer que todas ellas hereden de la superclase.



Relación de Asociación.- Es una relación estructural que describe un conjunto de enlaces o conexiones entre dos o más clases, permitiendo asociar objetos de las clases que colaboran entre si para llevar a cabo un comportamiento deseado.

Una asociación se representa gráficamente como una línea que conecta las clases relacionadas:

Esta asociación significa que la clase A puede acceder a los atributos y operaciones públicas de la clase B y que, de forma similar, la clase B puede acceder a los atributos y operaciones públicas de la clase A.

También se puede restringir la navegabilidad de la asociación, añadiendo una flecha que indique el sentido de dicha asociación:

En este caso, Funcionario puede acceder a los atributos y operaciones públicas de la clase Unidad, pero Unidad no puede acceder a la clase Funcionario.

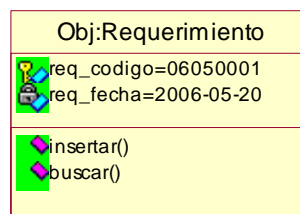


## 2.5.2 DIAGRAMA DE OBJETOS

Los diagramas de objetos son análogos a los de clases, con la particularidad de que en lugar de encontrar clases, encontramos instancias de éstas. Son útiles para explicar partes pequeñas del modelo en las que hay relaciones complejas. (Universidad Jaime I, Castellón ,2004)

### 2.5.2.1 Objetos

Un objeto se representa de la misma forma que una clase. En el compartimiento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre\_del\_objeto: nombre\_de\_la\_clase. Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase.



Los diagramas de objetos se utilizan, generalmente, para visualizar, especificar y documentar modelos estructurales, esto es, estructuras de objetos cuya complejidad aconseje dibujar un diagrama que sirva para una mejor comprensión del modelo.

En general, existen dos posibilidades:

- Partir del diagrama de clases y crear una instancia (diagrama de objetos) que permita una mejor comprensión del mismo.
- Partir de un diagrama de objetos significativo y modelar un diagrama de clases que posibilite, al menos, dicha instancia. Esto sucede, por ejemplo, cuando se conocen ejemplos de la estructura real y se quiere derivar un diagrama de clases al que se ajuste.

Para construir un diagrama de objetos hay que:

- Identificar la estructura o mecanismo que se quiera modelar.
- Identificar las clases, interfaces y otros elementos que intervengan en la relación.
- Considerar un escenario en el que intervenga ese mecanismo y congelar ese escenario, representando cada objeto que participe en dicho mecanismo.
- Mostrar los valores y atributos necesarios para comprender el escenario.
- Mostrar los enlaces entre esos objetos, que representarán instancias de asociaciones entre ellos.

Como en los diagramas de clases también se pueden mostrar objetos, un diagrama de clases que sólo contenga instancias de clases, esto es, objetos, se le puede denominar diagrama de objetos. Este caso sucede, por ejemplo, en Rational Rose: no hay un diagrama de objetos como tal, por lo que se debe utilizar un diagrama de clases para representar un diagrama de objetos. (Universidad de Alcalá, 2003)

Los diagramas de comportamiento; modelan el comportamiento y son: el de Casos de Uso, de secuencia, de colaboración, de estados y de actividades.

### **2.5.3 DIAGRAMA DE CASOS DE USO**

Los diagramas de Casos de Uso describen lo que hace un sistema desde el punto de vista de un observador externo.

Ayudan a capturar los requerimientos del sistema que será una cuestión básica para todo el desarrollo posterior. En los diagramas de casos de uso se muestran las diferentes formas posibles de utilización de un sistema. Los diagramas de casos de uso permiten visualizar el comportamiento de un sistema, de forma que los usuarios puedan comprender cómo se utiliza ese elemento y de forma que los desarrolladores puedan implementarlo. En un diagrama de casos de uso importa qué hace el sistema (qué proporciona), no cómo lo hace.

Los elementos principales que aparecen en los diagramas de casos de uso son:

- Actores.
- Casos de uso.
- Relaciones.

#### **2.5.3.1 Actores**

Es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema, son por lo tanto agentes externos al sistema que interactúan con él.

#### **2.5.3.2 Casos de uso**

Un caso de uso representa una funcionalidad que el sistema proporciona o, de forma equivalente, un caso de uso muestra una forma en la que alguien podría utilizar el sistema. Un caso de uso representa una secuencia de transacciones en un sistema cuyo objetivo es proporcionar un resultado medible de valor para el actor del sistema.

#### **2.5.3.3 Relaciones**

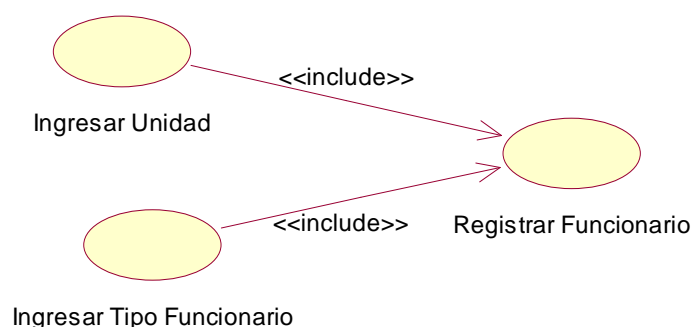
Hay cuatro tipos de relaciones que pueden aparecer en un diagrama de casos de uso.

Relación de comunicación.- Es el tipo más normal de relaciones y tiene lugar entre un caso de uso y un actor. Gráficamente es una línea, a la que se puede añadir una punta de flecha para indicar quién inicia la comunicación. Ejemplo:



En este caso, el actor "Administrador" se comunica con el caso de uso "Crear usuarios", siendo el actor quien inicia la comunicación.

Relación de inclusión.- Una relación de inclusión entre dos casos de uso significa que un caso de uso base incorpora explícitamente el comportamiento de otro caso de uso en el lugar especificado en el caso base. Esta relación se usa para evitar describir el mismo flujo de eventos repetidas veces, poniendo el comportamiento común en un caso de uso aparte (que será incluido por un caso base). Una relación de inclusión se representa como una dependencia, usando la palabra include. La relación de inclusión se muestra gráficamente como una relación de dependencia con el estereotipo «include». Ejemplo:

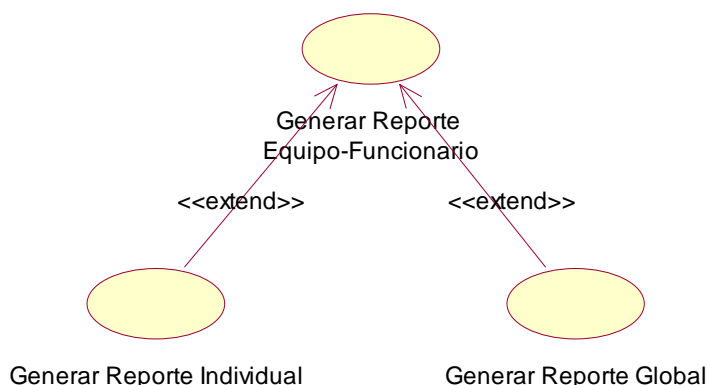


En este fragmento se puede ver que tanto el caso de uso "Ingresar Unidad" como "Ingresar Tipo Funcionario" incluyen el caso de uso "Registrar Funcionario".

Relación de extensión.- Una relación de extensión entre casos de uso significa que un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por el caso de uso que extiende al base. Un caso de uso puede extenderse solamente en ciertos puntos, llamados puntos de extensión. La extensión se puede ver como que el caso de uso que extiende, incorpora su comportamiento en el caso de uso base. Se representa como una dependencia con la palabra extend.

Una relación de extensión se usa para modelar la parte de un caso de uso que el usuario puede ver como comportamiento opcional del sistema. De esta forma se separa el comportamiento opcional del obligatorio. Es decir, un caso de uso base puede, bajo ciertas condiciones, incorporar el comportamiento de otro caso de uso en el lugar especificado.

La relación de extensión se muestra gráficamente como una relación de dependencia con el estereotipo «extend». Ejemplo:



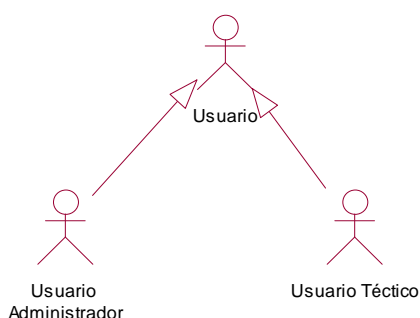
En este ejemplo, la funcionalidad “Generar Reporte Individual” y “Generar Reporte Global” extienden la funcionalidad “Generar Reporte Equipo-Funcionario”.

Relación de generalización.- Trata de representar la relación entre dos objetos del mismo tipo en el cual uno de ellos se comporta igual que otro pero que además contiene características adicionales que lo diferencian. La generalización es una relación de herencia y puede ser entre actores o entre casos de uso.



Se representa mediante una línea sólida con cabeza de flecha hueca, apuntando desde el caso de uso hijo hacia el caso de uso padre, o desde el actor hijo hacia el actor padre.

Entre actores: La relación de generalización entre actores se utiliza cuando varios actores tienen características comunes. Gráficamente se muestra como una relación de generalización que va del actor especializado al genérico. Para los usuarios del teléfono móvil podríamos representar, por ejemplo:



Se deben de usar las relaciones de generalización entre actores cuando algunos actores tengan en común parte de sus comportamientos o cuando interese mostrar explícitamente que cualquier especialización de un tipo de actor actúa de igual forma.

Entre casos de uso: La relación de generalización entre casos de uso se utiliza para indicar que un caso de uso es un refinamiento de otro.

Una relación de generalización desde un caso de uso A hacia un caso de uso B, indica que A hereda la funcionalidad de B, pero además contiene otras funcionalidades que lo diferencian. El caso de uso A puede reemplazar a B.

Este tipo de relación se utiliza poco y no se debe abusar de ella.

En los diagramas de interacción se muestra un patrón de interacción entre objetos, la vista de interacción describe secuencias de intercambios de mensajes entre los roles que implementan el comportamiento de un sistema. Un rol es la descripción de un objeto, que desempeña un determinado papel dentro de una interacción, distinto de los otros objetos de la misma clase. Esta visión

proporciona una vista integral del comportamiento del sistema, es decir, muestra el flujo de control a través de muchos objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

#### 2.5.4 DIAGRAMA DE SECUENCIA

El diagrama de interacción, representa la forma en como un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Objeto/Actor



Dicho diagrama puede ser obtenido de dos partes, desde el Diagrama Estático de Clases o el de Casos de Uso.

Muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

Consta de dos ejes. Generalmente, el eje vertical es el eje del tiempo, transcurriendo éste de arriba a abajo. En el otro eje se muestran los objetos que participan en la interacción, siendo el primero de ellos el actor que inicia la ejecución de la secuencia modelada.

De cada objeto parte una línea discontinua, llamada línea de la vida, que representa la vida del objeto durante la interacción. Si el objeto existe durante toda la interacción, éste aparecerá en el eje horizontal y su línea llegará hasta el final del diagrama de secuencia.

Los mensajes parten de la línea de vida del objeto que lo envía hasta la línea de vida del objeto al que va. Cada mensaje lleva un número de secuencia creciente

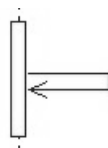
con el tiempo y el nombre de la operación requerida, así como posibles argumentos que pueden utilizarse como valores de entrada y/o salida.

#### Mensajes



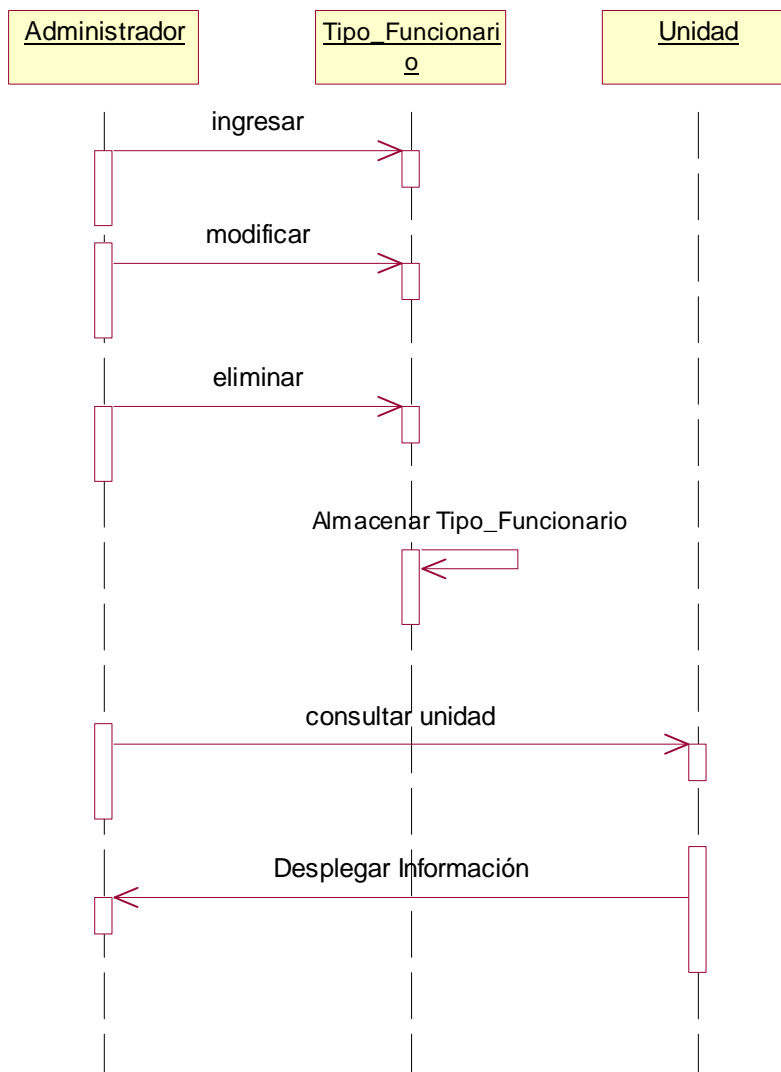
Opcionalmente, se puede mostrar en los diagramas de secuencia el foco de control, que es un pequeño rectángulo puesto sobre la línea de la vida que indica qué objeto tiene el control en cada momento (esto es, el tiempo durante el cual está realizando una acción directa o indirectamente).

#### Mensaje al Mismo Objeto



Un objeto se destruye mediante un mensaje de estereotipo «destroy», dando finalizada la línea de vida con un aspa (X).

Ejemplo:



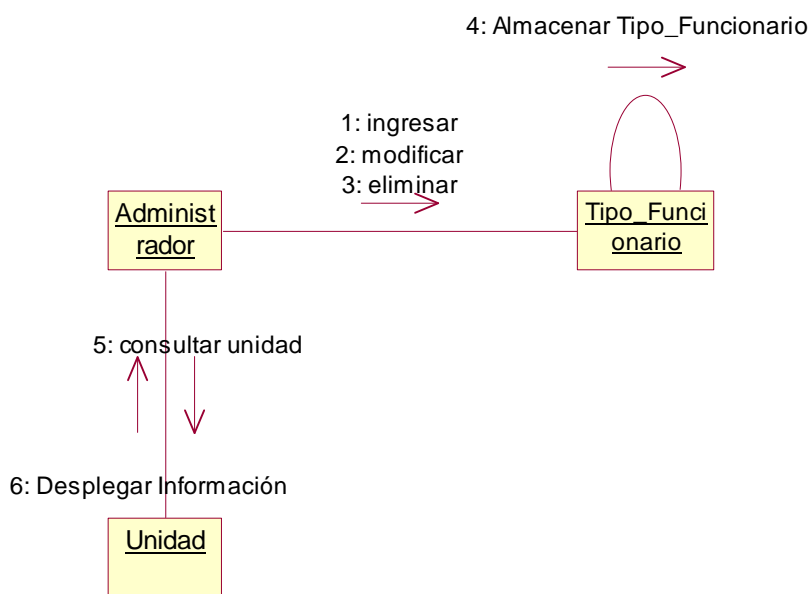
Los diagramas de secuencia permiten mostrar escenarios de forma que sean comprensibles tanto por clientes como por desarrolladores. Sin embargo, una vez que los clientes han dado su visto bueno a los mismos, se suelen refinar añadiendo detalles de diseño que serán útiles a los desarrolladores.

### 2.5.5 DIAGRAMA DE COLABORACIÓN

Los diagramas de colaboración son otro tipo de diagramas de interacción, que contiene la misma información que los de secuencia, sólo que se centran en las

responsabilidades de cada objeto, en lugar de en el tiempo en que los mensajes son enviados. Cada mensaje de un diagrama de colaboración tiene un número de secuencia. El primer nivel de la secuencia es 1, y los mensajes que son enviados durante la misma llamada a un método se numeran 1.1, 1.2 y así sucesivamente para tantos niveles como sea necesario.

Un diagrama de colaboración es un diagrama que muestra una interacción en el cual se destaca la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra los objetos que intervienen en una relación, los mensajes que se intercambian y las relaciones de comunicación que hay entre ellos (para que un objeto pueda comunicarse con otro debe existir algún tipo de enlace o link entre ellos). Ejemplo:



En el diagrama se ve claramente qué objetos participan y las relaciones que existen entre ellos, lo que les hace muy útiles para los desarrolladores. Estos diagramas permiten, por ejemplo, valorar el impacto que supondría un cambio en una clase.

La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás. (Patricio Letelier Torres, 2000).

## 2.5.6 DIAGRAMA DE ESTADOS

Este diagrama muestra la secuencia de los estados de un objeto durante su ciclo de vida, en respuesta a un estímulo recibido. Los estados de los objetos están dados por el valor de sus atributos (estados) lo cual cambia sus comportamientos (métodos).

Los estados hacen referencia a una condición durante la vida de un objeto o a una interacción durante la cual se satisface alguna condición (ejecutar alguna acción, esperar algún evento, etc).

Un objeto permanece en un estado por un tiempo finito, hasta que se cumpla la condición de cambio. Se construyen a partir del Diagrama de Estructura Estática, identificando cuáles objetos cambian de estado, cual es el estado inicial y el final, definiendo a qué eventos puede responder el objeto, y qué transacciones ejecutará

### 2.5.6.1 Estado

Identifica un periodo de tiempo del objeto (no instantáneo) en el cual el objeto está esperando alguna operación, tiene cierto estado característico o puede recibir cierto tipo de estímulos. Se representa mediante un rectángulo con los bordes redondeados.



NombreEstado

### 2.5.6.2 Estados especiales

Estado inicial. Es un pseudoestado que representa el estado en que se encuentra un objeto cuando es creado (inicio de la vida del mismo). Debe aparecer obligatoriamente uno y sólo uno en todo diagrama de estados. Gráficamente se representa:

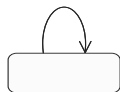
Estado final. Es un pseudoestado que indica la destrucción del objeto (finalización de la vida del mismo). Es opcional y se pueden añadir varios estados de fin para simplificar el diagrama y evitar el cruce de líneas. Gráficamente se representa:

### 2.5.6.3 Transición

Una transición es el paso de un estado a otro. Se muestra gráficamente como una flecha que va del estado origen al estado destino:

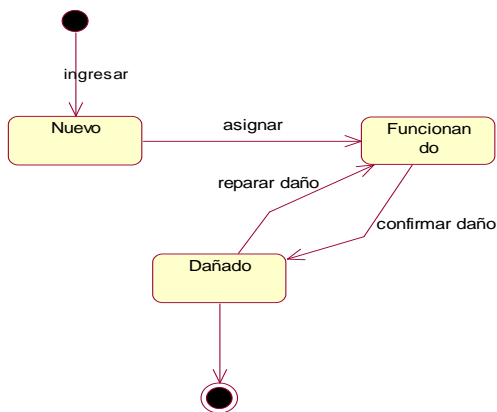


Este movimiento puede ser reflexivo, es decir, el estado origen y el estado destino pueden coincidir, tal y como se muestra a continuación:



Las transiciones pueden incluir especificaciones que indiquen cuándo ocurre la transición o qué acciones se llevarán durante la transición.

En un diagrama de estados aparecen uno o más estados relacionados entre sí por transiciones. Las transiciones, en general, serán ocasionadas por algún evento. Por ejemplo, este podría ser el diagrama de estados de un objeto de la clase Equipo:



## **2.5.7 DIAGRAMA DE ACTIVIDADES**

Los diagramas de actividad permiten modelar el comportamiento de un sistema o alguno de sus elementos, mostrando la secuencia de actividades o pasos que tienen lugar para la obtención de un resultado o la consecución de un determinado objetivo. Opcionalmente, permite mostrar los flujos de información (objetos) producidos como resultado de una actividad y que serán utilizados posiblemente como entrada por la actividad siguiente.

### **2.5.7.1 Actividades**

El elemento fundamental de los diagramas de actividad son las actividades. Una actividad representa la ejecución de una tarea o misión en un flujo de trabajo o la ejecución de una sentencia en un procedimiento, dependiendo del elemento cuyo comportamiento se esté modelando.

La representación es un rectángulo con las puntas redondeadas, en cuyo interior se representa bien una actividad o bien una acción.

### **2.5.7.2 Bifurcaciones**

Un flujo de control no tiene porqué ser siempre secuencial, puede presentar caminos alternativos. Para poder representar dichos caminos alternativos o bifurcación se utilizará como símbolo el rombo. Dicha bifurcación tendrá una transición de entrada y dos o más de salida.

### **2.5.7.3 Transiciones**

Las transiciones reflejan el paso de un estado a otro, bien sea de actividad o de acción. Esta transición se produce como resultado de la finalización del estado del que parte el arco dirigido que marca la transición. Como todo flujo de control debe empezar y terminar en algún momento, podemos indicar esto utilizando dos disparadores de inicio.



### 2.5.7.4 División y unión

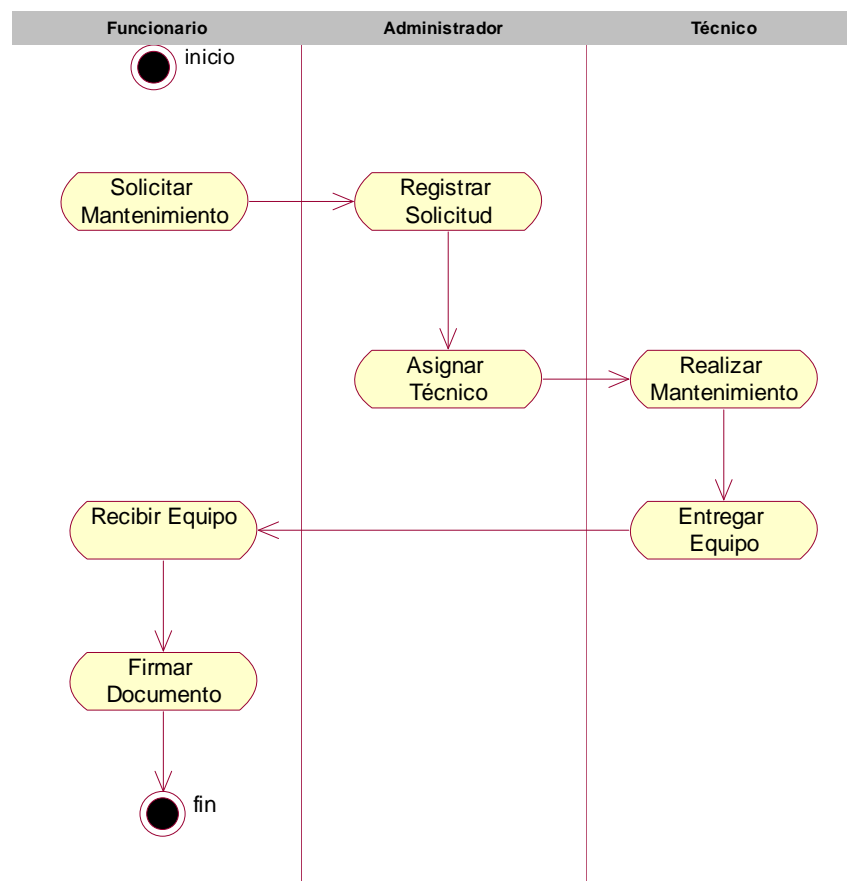
No sólo existe el flujo secuencial y la bifurcación, también hay algunos casos en los que se requieren tareas concurrentes. UML representa gráficamente el proceso de división, que representa la concurrencia, y el momento de la unión de nuevo al flujo de control secuencial, por una línea horizontal ancha.

### 2.5.7.5 Calles

Cuando se modelan flujos de trabajo de organizaciones, es especialmente útil dividir los estados de actividades en grupos, cada grupo tiene un nombre concreto y se denominan calles. Cada calle representa a la parte de la organización responsable de las actividades que aparecen en esa calle

Ejemplo

Actividad: Mantenimiento Equipo



## CAPITULO III

### HERRAMIENTAS UTILIZADAS

#### 3.1 PHP (Preprocesador de Hipertexto)

Es un lenguaje interpretado de alto nivel incluido en páginas HTML y ejecutado en el servidor. El objetivo de este lenguaje es permitir la construcción de páginas Web dinámicas, de manera fácil y rápida. PHP trabaja en conjunto con un servidor Web y puede ser usado con una variedad de Sistemas Operativos como Microsoft Windows y UNIX (multiplataforma), aunque las mejores prestaciones del lenguaje se obtienen trabajando en un entorno Unix o Linux con servidor web Apache.

PHP se diferencia de otros scripts CGI (Interfaz de pasarela común) escritos en lenguajes como Perl o C los cuales necesitan ser creados como programas separados fuera de HTML; PHP es diferente porque está incluido dentro del documento HTML, con etiquetas especiales de inicio y fin que nos permite ingresar y salir de PHP. Esto proporciona un rápido despliegue de las páginas, alta seguridad y transparencia para el usuario final.

La sintaxis de PHP es similar a la de los lenguajes de programación C, C++ y Java. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Es decir, cuando creamos una variable no tenemos que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador. Además PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni estamos limitados en su distribución.

Una de las más fuertes e importantes facilidades que brinda PHP es su amplio soporte para una gran cantidad de bases de datos. Tiene acceso un gran número

de gestores de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, mSQL, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc.

PHP es un lenguaje que se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente y el resultado es enviado al navegador. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Dado que la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.



Cuando un navegador solicita al servidor Web una página PHP, antes de enviar dicha página al cliente, se la pasa al intérprete de PHP. Éste la interpreta y es el resultado de esta interpretación del programa PHP, contenido en la página PHP, lo que termina llegando al cliente.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. (Gracia, 2004).

### 3.2 HTML (Lenguaje de formato de documentos de hipertexto)

Es un lenguaje de marcas (lenguaje de anotaciones o de etiquetas) diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

Nos permite preparar documentos Web insertando en el texto de los mismos una serie de marcas (tags) que controlan los diferentes aspectos de la presentación y comportamiento de sus elementos.

Las marcas o tags que controlan el comportamiento del documento son fragmentos de texto encerrados entre los signos "mayor que" y "menor que" (<marca>). Existen diferentes tipos de marcas: algunas controlan simplemente la presentación del texto del documento; otras, la forma en que se incluirán en él imágenes; otras, finalmente, los hiperenlaces con documentos o con diferentes partes del mismo documento

Las marcas funcionan muchas veces por parejas, una para indicar el inicio de enlace o formato, y otra para señalar el final. La marca de inicio consiste en una letra o una palabra (por ejemplo, estas son marcas de inicio: <B>, <TITLE>). La marca de final es la misma letra o palabra precedida por la barra inclinada o "slash" (es decir, </B>, </TITLE>). Existen, no obstante, algunas marcas que no requieren su pareja de cierre, como <BR> (que fuerza un salto de línea). Es importante señalar que las marcas, en general pueden estar indistintamente en mayúsculas o en minúsculas.

Es recomendable que todo fichero HTML siga la siguiente estructura:

```
<HTML>
<TITLE>Título de la ventana</TITLE>
<BODY>
.....comandos y texto.....
</body>
</HTML>
```

Como puedes comprobar, estos comandos tienen una orden de inicio y otra de

fin, que no es más que el mismo comando con el signo / antecediéndolo. Los comandos pueden figurar en letras mayúsculas o en minúsculas, indistintamente.

La secuencia lógica de estas órdenes es la siguiente:

Inicio de un documento HTML

Inicio del título.

Título (que conviene poner para identificar la página de cara al visitante).

Final del título.

Inicio del cuerpo de la página, esto es, de aquello que queremos visualizar.

Fin del cuerpo de la página.

Fin del documento HTML.

Necesitas dos herramientas, un visualizador de HTML (son populares los productos Netscape o Mosaic), y un editor de texto. Cuando almacenes un fichero, es conveniente que le pongas por extensión (.htm), que es el tipo de fichero que por defecto buscará el visualizador

Gracias a Internet y a los navegadores del tipo Explorer, Mozilla, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como puede ser el sencillo Bloc de Notas de Windows (o Notepad), o cualquier otro editor que admita texto sin formato como Microsoft Word, Wordpad. (Universidad de Murcia, 2001).

### **3.3 JAVASCRIPT**

Javascript es un lenguaje de archivos de comandos (scripts) basado en objetos, desarrollado por Netscape. Su estructura es comparable con la de C o C++. Javascript se integra en un documento HTML y se interpreta por el navegador. Los programas pueden reaccionar ante acciones del usuario lo que hace dinámicos a los documentos web.

JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor. Un script JavaScript insertado en un documento HTML permite

reconocer y tratar localmente, es decir, en el cliente, los eventos generados por el usuario. Estos eventos pueden ser el recorrido del propio documento HTML o la gestión de un formulario. Cuando la página HTML es un formulario que permite acceder a un anuario telefónico, se puede insertar un script que verifique la validez de los parámetros proporcionados por el usuario. Esta prueba se efectúa localmente y no necesita un acceso a la red.

Javascript no tiene dependencia funcional bajo ninguna plataforma y sólo está vinculado al navegador que lo interpreta. Por ello se pueden encontrar diferencias entre los distintos navegadores. (Universidad de las Palmas de Gran Canaria, 2002)

#### Utilización de JavaScript en un documento HTML

La inserción dentro de un documento HTML se realiza mediante la marca SCRIPT utilizando la sintaxis:

```
<SCRIPT>
```

```
Código del script
```

```
</SCRIPT>
```

Es preferible delimitar los scripts insertados en un documento por comentarios HTML para asegurarse de que el contenido del script no aparecerá en los clientes que no reconozcan la marca SCRIPT. Por ejemplo:

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<-- Disimula el contenido del script para navegadores no compatibles
```

```
Código del script
```

```
//-->
```

```
</SCRIPT>
```

### 3.4 JPGRAPH

Una imagen vale más de mil palabras, y al momento de presentar información en una página web, el utilizar graficas en vez de enormes tablas, nos puede ayudar a conceptualizar mejor los datos. PHP es una gran ayuda para esto, puesto que con

el motor de gráficos GD nos permite generar gráficas en tiempo real sobre los datos que podemos tener en nuestras bases de datos.

Para llegar a implementarlo existen librerías de código abierto ya existentes como JpGraph.

JpGraph es una librería que incluye una serie de clases orientadas a objetos para PHP, para la creación dinámica de imágenes a partir de datos. Las gráficas pueden ser de distintos tipos, tales como de barras, de líneas de progreso, de tarta, etc.

Con JpGraph se pueden crear gráficas complejas con un mínimo de código y con un control muy detallado, con el único requerimiento de que se tenga instalado el soporte para la librería GD Graphics library.

### **3.4.1 PRINCIPALES CARACTERÍSTICAS**

Reducido peso en bytes de las imágenes resultado. en promedio una imagen de un tamaño de 300 \* 200 ocupa alrededor de 2K y en general las imágenes rara vez ocuparán más de 4-5K.

Soporte a las librerías GD1 o GD2. La librería auto detecta que versión del GD se tiene instalada.

Soporta formatos gráficos de tipo: PNG, GIF y JPG y puedes seleccionar el mejor formato para una gráfica. Aunque esto depende de la instalación que se tenga de PHP y de la librería que se esté utilizando.

Uso de la Interpolación matemática para obtener curvas a partir unos pocos valores.

Diversos tipos de gráficas 2D o 3D, como de puntos, líneas, tartas, barras, cajas, etc.

Escalas flexibles tanto en el eje X como el Y, que se ajustan al juego de datos que se tenga que representar.

Configurable con distintos tipos de colores, leyendas, tipografías, imágenes de fondo, etc.

### 3.4.2 FUNCIONAMIENTO

El modo de trabajo para usar esta librería es muy simple, se trata de crear una imagen con la etiqueta <img> de HTML, en cuyo atributo src colocaremos la ruta hacia el script PHP que se encargará de generar la gráfica; como se muestra a continuación:

```
<html>
<body>
<imgsrc="grafica.php">
</body>
</html>
```

En el archivo PHP que generará la gráfica tendremos que incluir las librerías apropiadas para el tipo de gráfica que deseemos realizar, también habrá que instanciar el objeto JpGraph correspondiente, cargar los datos a visualizar y llamar a los métodos adecuados para mostrar la imagen.

A pesar de que en esta herramienta los mensajes de error son algo escasos, y la mayoría de las veces cuando algo falla no obtenemos ninguna explicación; se trata de una librería gratuita, fácil de instalar y de fácil manejo. (González, 2003)

### 3.5 BIBLIOTECA GRÁFICA GD

La biblioteca gráfica GD es un añadido de PHP que ofrece una serie de funciones para la creación y manipulación dinámica de gráficos. Dentro de las tareas que nos permite manejar tenemos:

- Tratamiento automatizado de imágenes
- Creación de gráficos mediante el dibujo de formas geométricas o a partir de otros gráficos
- Creación de textos a partir de una tipografía TTF



Una de las mayores ventajas de esta biblioteca es el ahorro de memoria, ya que permite crear las imágenes en memoria que se eliminarán en cuanto se dejen de usar, no ocupando así espacio en el servidor.

Actualmente es posible leer los formatos JPG, GIF y PNG y escribir en JPG, PNG y WBMP.

### **3.6 ENCRIPtar CONTRASEÑAS CON MD5**

Para prevenir posibles vulnerabilidades en nuestro servidor, es una medida eficaz encriptar las contraseñas.

Si en nuestra página web tenemos un sistema de usuarios y queremos proteger las contraseñas para prevenir posibles vulnerabilidades en nuestro servidor, es una medida eficaz encriptar las contraseñas, de manera que si alguien puede acceder a ellas no pueda ver la contraseña si no su encriptación.

Para mejorar este sistema, lo que haremos es usar un algoritmo de encriptación de un solo sentido, es decir que no se puede desencriptar de ninguna manera, como por ejemplo md5.

Para guardar la contraseña encriptada en md5, usaremos la función md5() de PHP:

```
<?
$b_password = md5($b_password);
?>
```

Ya podemos guardar la contraseña en nuestra base de datos o fichero, pero... ¿como haremos para comprobar la contraseña en el inicio de sesión?

Muy fácil, como en la base de datos tenemos la contraseña en md5, encriptaremos la contraseña que escribe el visitante de la misma manera que hemos encriptado la contraseña del usuario en el momento de su registro, ahora ya podremos comparar la contraseña enviada con la almacenada en el servidor, si la encriptación coincide es que la contraseña es correcta.

## 3.7 MYSQL

Es uno de los Sistemas Gestores de bases de Datos (SQL) más populares desarrolladas bajo la filosofía de código abierto.

La desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible.

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

### 3.7.1 CONECTAR MYSQL DESDE PHP

Una vez que tenemos creada la base de datos en nuestro servidor, el siguiente paso es conectarnos a la misma desde una página PHP. Para ello PHP nos proporciona una serie de instrucciones para acceder a bases de datos MySQL.

```
<?php
```

```
#declaracion de variables
```

```
$usql="root";
```

```
$psql="";
```

```
$host="";
```

```
$base="sistec";
```

```
#conexión
$link = mysql_connect("$host", "$usql", "$psql")
or die ("no se ha podido conectar");

#Selección de la base de datos a utilizar
mysql_select_db($base,$link)
or die("Error al tratar de seleccionar esta base");

#cierra la base de datos
mysql_close($link);
?>
```

Al ejecutar la instrucción `mysql_connect` creamos un vínculo entre la base de datos y la página PHP, este vínculo será usado posteriormente en las consultas que hagamos a la base de datos.

Finalmente, una vez que hemos terminado de usar el vínculo con la base de datos, lo liberaremos con la instrucción `mysql_close` para que la conexión no quede ocupada.

### 3.7.2 PHPMYADMIN

Es un programa de libre distribución en PHP. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.

**Home**

- mysql (6)
- sistec (16)
  - equipo
  - equipo\_funcionar
  - funcionario
  - material\_equipo
  - numeracion
  - permisos\_usuari
  - proveedor
  - requerimiento
  - seguimiento
  - tipo\_equipo
  - tipo\_funcionario
  - tipo\_problema
  - transferencia
  - transferenciad
  - unidad
  - usuario
- sistec1 (19)
- test (-)

**Base De Datos sistec - tabla usuario ejecutandose on localhost**

[ Examinar ] [ Seleccionar ] [ Insertar ] [ Vaciar ] [ Eliminar ]

Campo	Tipo	Atributos	Null	Defecto	Extra	Acción
<input type="checkbox"/> id_user	int(3)		No	0		Cambiar Eliminar Primaria I
<input type="checkbox"/> nombre	varchar(50)		No			Cambiar Eliminar Primaria I
<input type="checkbox"/> apellido	varchar(50)		No			Cambiar Eliminar Primaria I
<input type="checkbox"/> user	varchar(50)		No			Cambiar Eliminar Primaria I
<input type="checkbox"/> password	varchar(100)		No	0		Cambiar Eliminar Primaria I
<input type="checkbox"/> tipo	char(1)		No			Cambiar Eliminar Primaria I

↑ With selected:

Indices : [ Documentación ]

Nombre de la clave	Tipo	Cardinality	Acción	Campo
PRIMARY	PRIMARY	1	Eliminar Editar	id_user

Crear un indice en  columnas

Espacio utilizado :

Tipo	Uso
Datos	60 Bytes
Indice	2,048 Bytes
Total	2,108 Bytes

La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

(<http://www.desarrolloweb.com/articulos/844.php>)

### 3.7.3 RESPALDOS DE BASES DE DATOS

El gestor de Bases de datos MySQL incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podremos contar con un respaldo de nuestros datos, para que, en el eventual caso de que se pierdan, poderlos recuperar.

Existen varias opciones para realizar la copia de seguridad de una base de datos de Mysql.

- Se puede utilizar alguna herramienta comercial que se encargue de realizar todo el proceso, pero tenemos el inconveniente que normalmente no se trata de software gratuito y tenemos que pagar una licencia para su uso.
- También es posible utilizar la sentencia sql "backup table" que esta disponible desde la versión 3.23.25 y posteriores.
- Finalmente, se puede realizar la copia de seguridad mediante herramientas que nos proporciona el propio gestor de base de datos, como pueden ser mysqlhotcopy ó mysqldump.
- mysqldump

El comando mysqldump del sistema gestor de base de datos MySQL sirve para hacer copias de seguridad de una o múltiples bases de datos. Además permite que estas copias de seguridad se puedan restaurar en distintos tipos de gestores de bases de datos, sin la necesidad de que se trate de un gestor de mysql.

El comando mysqldump puede ser usado para crear respaldos de todas las bases de datos, algunas bases de datos, sólo una de ellas, o incluso ciertas tablas de una base de datos dada.

El comando mysqldump tiene la siguiente sintaxis:

```
mysqldump -u [user] -p [pass] [dbname] > [backup.sql]
```

Además dispone de una amplia variedad de opciones que permite realizar la copia de la forma más conveniente para el propósito de la misma.

Una opción que permite realizar la copia de seguridad de la base de datos de forma más rápida y efectiva es:

--opt

La cual es equivalente a realizar --quick --add-drop-table --add-locks --extended-insert --lock-tables.

```
mysqldump --opt --password=miclave --user=miuser mibasededatos > archivo.sql  
(http://desarolloweb.com).
```

### **3.7.4 SERVIDORES PARA PÁGINAS WEB**

Un servidor web es un programa que se está ejecutando en un equipo, normalmente un servidor y que proporciona páginas web a los "clientes" que le piden. Los clientes son los navegadores web como Internet Explorer o NetScape. Cuando llamamos a un equipo "servidor" nos referimos a un equipo que va a proporcionar determinados servicios a los usuarios de una red local o de Internet. Lleva un equipamiento de prestaciones más avanzadas que un equipo de sobremesa y sobre todo incorpora un sistema operativo de servidor.

Por tanto si en un navegador o explorador web escribimos una dirección o una página intentará localizar el servidor escrito y buscar la página solicitada.

#### **3.7.4.1 Apache**

Apache es un servidor HTTP altamente configurable de diseño modular, es muy sencillo ampliar las capacidades del servidor, es una tecnología gratuita de código fuente abierta.

Una de las principales características que presenta Apache es que funciona en plataformas virtuales muy utilizadas funciona en la mayoría las versiones de Unix, además funciona en Windows y en muchos otros sistemas operativos.

Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado httpd.conf que se puede utilizar para configurar Apache solo necesita su editor de texto.

Apache fue elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Para poder hacer esto, Apache, una vez que se haya iniciado, crea unos subprocesos para poder gestionar las solicitudes

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero es criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado.

### 3.8 ARQUITECTURA DE TRES CAPAS

En una Aplicación Web el usuario interactúa con las aplicaciones web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente.



Esta distribución se conoce como el modelo o arquitectura de tres capas. (Sigef, 2005).



## **CAPITULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **4.1 CONCLUSIONES**

El uso de una metodología adecuada, en nuestro caso la OOHDM es de gran importancia para el desarrollo de la aplicación ya que nos sirve como guía para su construcción nos ayuda a entender y lograr en cada etapa lo que el usuario realmente necesita, además nos permite llevar un control del desarrollo de las etapas y tener la posibilidad de realizar una rápida detección, corrección de errores, nos permite simplificar y a la vez hacer más eficaz el diseño de la aplicación.

Los diagramas UML también son de gran ayuda para la construcción del sistema ya que nos ayudan a modelar, construir los elementos que constituyen el mismo. Se representa el sistema a partir de los objetos u entidades que lo constituyen, con unos atributos y operaciones asociados, que interactúan con otros objetos para conseguir conjuntamente satisfacer los objetivos del sistema

El uso de software libre, para el desarrollo de software tiene grandes ventajas, ya que se puede construir software de calidad a costos más económicos.

Los sistemas y aplicaciones Web tienen muchas ventajas sobre los sistemas de información tradicionales, ya que pueden funcionar en una Intranet para un grupo de usuarios; o en Internet para el acceso a gran cantidad de personas.

## **4.2 RECOMENDACIONES**

Es necesario que el sistema SISTEC sea implementado en todas las regionales del Ministerio del Ambiente, ya que su uso facilitará de gran manera las actividades diarias de los funcionarios.

Una vez implementado el sistema SISTEC, es importante que se realice un curso de capacitación a las personas que utilizaran el sistema, para que se pueda aprovecharlo al máximo.

El sistema SISTEC cubre las necesidades actuales del Ministerio, pero para futuros requerimientos se recomienda implementar nuevos módulos que manejen información que actualmente no maneja el sistema como por ejemplo en lo que se refiere al software, sus licencias así como también conocer el sw instalado en los equipos de computación.

Para un mejor funcionamiento del sistema, se recomienda llevar un mantenimiento continuo del sistema.

Es de gran urgencia para el área informática, comenzar a utilizar en forma estandarizada una metodología de desarrollo que ayude a crear aplicaciones reusables y fáciles de mantener.

## REFERENCIAS BIBLIOGRAFICAS

### LIBROS

Pressman, R (2002) Ingeniería de Software, 5ª edición, Madrid (España), editorial Concepción Fernández.

Liza Avila, C (2000), Modelando con UML Principios y Aplicaciones.

### INTERNET

Cuenca L (2005). Backup MySQL con mysqldump.  
<http://www.desarrolloweb.com/articulos/1202.php?manual=34>

González P (2003), Creación de gráficas en PHP con JpGraph  
<http://www.desarrolloweb.com/articulos/1987.php>

Makarewicz M (2005). Generar gráficos utilizando la librería GD  
[http://www.programacion.com/php/tutorial/phpsol\\_gd/](http://www.programacion.com/php/tutorial/phpsol_gd/)

Sigef (2005). Modelos o arquitecturas de computación.  
[http://www.sigef.gov.ec/sigef/index.php?option=com\\_content&task=view&id=35&Itemid=35](http://www.sigef.gov.ec/sigef/index.php?option=com_content&task=view&id=35&Itemid=35)

Wipimedia (2006). OOHDM.  
<http://es.wikipedia.org/wiki/OOHDM>

Universidad de Alcalá de Henares 2003. Orientación a objetos en la documentación hipermedia.  
<http://www.ati.es/gt/LATIGOO/OOp96/Ponen6/atiao6p06.html>

Mercerat B. (2002). Construyendo aplicaciones web con una metodología de diseño orientada a objetos.

[http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22\\_art5\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf)

Universidad de Chile (2001). Modelo de clases.

<http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>

Fowler (1999). Diagramas de estructura estática.

<http://www.clikear.com/manuales/uml/diagramasestructuraestatica.asp>

Letelier P (2000). Diagramas de interacción

<http://www.creangel.com/uml/interaccion.php>

Gracia J (2004). Iniciación a php.

<http://www.webestilo.com/php/php00.phtml>

Universidad de Murcia (2001). HTML

<http://www.um.es/psibm/tutorial/#t01>

Universidad de las Palmas de Gran Canaria (2002). Javascript.

<http://www.ulpgc.es/otros/tutoriales/JavaScript/index.htm>

Álvarez A (2002). Phpmyadmin.

<http://www.desarrolloweb.com/articulos/844.php>

Lagrange, M (2004). Encriptar contraseñas con md5

<http://www.maestrosdelweb.com/editorial/md5/>