ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN TECNOLÓGICA

DISEÑO E IMPLANTACIÓN DE UNA APLICACIÓN WEB PARA LA ADMINISTRACIÓN DE PROGRAMA DE EMBARQUES, PROGRAMA BUQUES Y COMPAÑÍAS PARA LA EMPRESA "PETROECUADOR"

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS INFORMÁTICOS

HOLGER DAVID CRUZ TAMAYO

DIRECTOR: ING. MYRIAM PEÑAFIEL

Quito, diciembre 2006

DECLARACIÓN

Yo, Holger David Cruz Tamayo, declaro que el trabajo aquí descrito es de mí autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

HOLGER CRUZ

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Holger David Cruz Tamayo, bajo mi supervisión.

ING. MYRIAM PEÑAFIEL

DIRECTOR DE PROYECTO

CONTENIDO

INDICE DE GRÁFICOS

INDICE DE CUADROS

CAPITULO 1

INTRODUCCIÓN

1.1 AMBITO

La empresa PETROECUADOR, pertenece al estado y su matriz se encuentra ubicada al norte de la Ciudad de Quito; en las calles Alpallana E8-86 y 6 de Diciembre; esta empresa brinda los servicios petroleros al País, la cual tiene un departamento de Sistemas y en él un área de diseño y desarrollo de software.

No existe un medio eficiente de comunicación, entre la matriz donde llegan los futuros acuerdos y operaciones a realizar y las terminales correspondientes, los datos acerca de dichas operaciones se las realizan vía telefónica y fax. La operatividad y el conocimiento que la empresa tiene con sus clientes debe ser mejorada.

La información que poseen los terminales de cada puerto acerca de sus programas de embarques, no es rápidamente ni periódicamente actualizada según las necesidades, de forma óptima y precisa.

Los clientes tienen la dificultad de comunicar a la empresa los servicios que requieren de manera detallada y con las especificaciones del caso de forma inmediata.

Las terminales de los puertos de nuestro País, no tienen una aplicación estándar para el conocimiento y uso de la información enviada desde la matriz acerca de las operaciones a realizar y principalmente no se puede ingresar o modificar los

programas o información de los cllientes desde cualquiera parte que se encuentren.

Se propone crear una aplicación web canalizada a proporcionar información y constante actualización de datos importantes acerca de los clientes, así como de las operaciones realizadas por la empresa, se puede desarrollar: consultas de compañías consignadora/consignataria, terminales, buques y volúmenes de carga o descarga de los productos según lo programado, toda la información necesaria acerca de los nuevos clientes.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Apoyar la comunicación entre los clientes, parte administrativa y la parte operativa de la empresa a través de una aplicación web.

1.2.2 OBJETIVOS ESPECÍFICOS

i) Desarrollar el módulo de Compañías.
ii) Desarrollar el módulo de Programa de Embarques.
iii) Desarrollar el módulo de Programa de Buques.
iv) Desarrollar las páginas para el mantenimiento de los catálogos, de: servicios, actividades, refinadoras, procesos de refinación, esquemas de refinación y mercados.
v) Generar Reportes de cada uno de las compañías, programas de embarques y programa de buques, que están o se han realizado para cada terminal.
vi) Mantener una base de datos de clientes, productos, procesos y operaciones programadas de la empresa que permita realizar consultas dinámicas en cada uno de los módulos.

1.3 JUSTIFICACIÓN

El desarrollo de la aplicación Web para la empresa PETROECUADOR, permitirá que la institución en su parte administrativa se encuentre todo el tiempo comunicada e informada, con su parte operativa y sus relaciones con otras compañías; promoviendo y mejorando el mercado financiero.

1.4 ALCANCE Y LIMITACIONES

1.4.1 ALCANCE

Realizar el diseño y desarrollo de una aplicación web para la empresa PETROECUADOR, canalizada a proporcionar información acerca de los de compañías y sobre operaciones que estas van a realizar en cada terminal del País.

Con el módulo de Compañías se procederá a ser la calificación, según reglamento, a las compañías, teniendo estados de calificación; así, como casos donde se las registrará como operativas o no operativas para realizar las actividades que demanda la empresa. Se mantendrá totalmente asegurada la información con documentos tipo imagen en la base de datos, para que se respalde la calificación y otros procesos importantes del módulo de compañías.

Con respecto a los programas tanto de buques como de embarques, se canalizará la información para que se tenga total administración de la programación en la matriz. En el caso de los embarques la información será reflejada a las regionales, cuando los embarques se encuentren en estado de instrucciones, los cuales podrán en su momento pasar a la ejecución de los embarques.

Por cada sección de cada módulo se contará con sus respectivas consultas y reportes, según criterios relevantes como fechas, productos, buques, etc. Por los que se los pueda filtrar.

La aplicación web posibilitará mejorar la comunicación de la Empresa, principalmente su parte operativa, que se la realiza en las terminales.

1.4.2 LIMITACIONES

Los módulos son totalmente dependientes de otros que ya se han integrado y otros que se los están desarrollando.

1.5 PRESUPUESTO

CUADRO 1. PRESUPUESTO

Costo Hardware	
Computadores Pentium IV 2.4 GHz	3400
Impresora hp Laser Jet 1300n PCL 6	300
Total	**3700**

Costo Software	
Macromedia Dreamweaver MX	850
Oracle	140000
ColdFusion MX 7	6000
Windows XP	500
Total	$ 147350

Costos Indirectos	
CDs	20
Disquetes	5
Papel	10
Suministro de Oficina	4
Agua	30
Luz	21
Teléfono	40
Total	**126**

Costos Recursos Humanos			
Recurso	**Costo/Hora**	**Horas**	**Sub Total**
Diseñador	12	90	1080
Programador	12	90	1080
		Total	**2160**

TOTAL COSTOS DEL PROYECTO	
Costos Hardware	3700
Costos Software	147350
Costos Indirectos	126
Costos Recursos Humanos	2160
Total	**$153336**

ELABORACION: Holger Cruz

CAPITULO 2

MARCO TEÓRICO

2.1 PARADIGMA ESPIRAL ORIENTADO A LA WEB

2.2 METODOLOGÍA

2.2.1 INTRODUCCIÓN

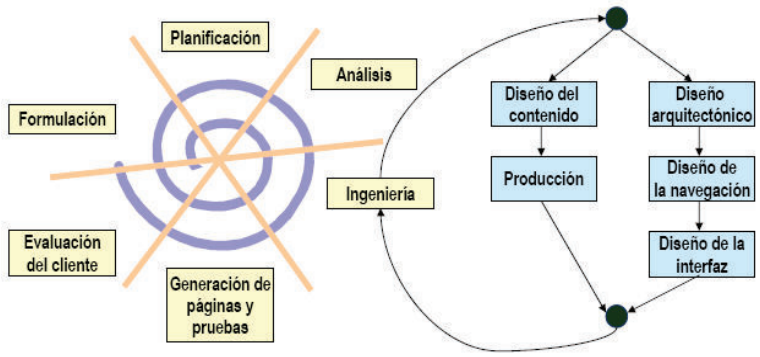
A medida que la evolución de las WebApps (Sistemas de aplicación basadas en Web) pasa de utilizar recursos estáticos de información controlada por el contenido, a utilizar entornos de aplicación dinámicos controlados por el usuario, cada vez es más importante la necesidad de implementar una gestión sólida y unos principios de ingeniería. Para conseguir esto, es necesario desarrollar un marco de trabajo IWeb (Ingeniera Web) que acompañe a un proceso eficaz, popularizado por las actividades del marco de trabajo y por la tarea de Ingeniería,

El modelo en espiral trata de desarrollar incrementalmente el proyecto, dividiéndolo en muchos subproyectos. Uno de los puntos más importantes del proceso es concentrarse primero en los aspectos más críticos del proyecto. La idea es definir e implementar las características más importantes primero, y con el conocimiento adquirido para hacerla, volver hacia atrás y reimplementar las características siguientes en pequeños subproyectos.

El modelo en espiral orientado a la web se divide en un número de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.

Modelo orientado al riesgo. Es el más versátil y flexible, pero también el más complejo. Cada vuelta de la espiral (ciclo) supone una refinación en el desarrollo.

GRÁFICO 1. PARADIGMA ESPIRAL ORIENTADO A LA WEB



Fuente: http://gplsi.dlsi.ua.es/iwad/investigacionWeb.html

2.2.2 ETAPAS

A continuación se indican las etapas que presenta este Modelo:

2.2.2.1 COMUNICACIÓN CON EL CLIENTE O FORMULACIÓN

Las tareas requeridas para establecer comunicación entre el desarrollador y el cliente. Actividad que identifica 'las metas y lo objetivos de la WebApp, y establece el ámbito del primer incremento.

2.2.2.2 PLANIFICACIÓN

La planificación estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo, y define una planificación del desarrollo bien granulada para el incremento final de la WebApp, con una aplicación mas toscamente granulada para los incrementos subsiguientes. Son todos los requerimientos.

2.2.2.3 ANÁLISIS DE RIESGOS

Establece los requerimientos técnicos para la WebApp e identifica los elementos del contenido que se van a incorporar. También se define los requisitos de diseño

gráfico (estética). Es decir se identifica los datos y requisitos funcionales y de comportamiento para la aplicación web.

2.2.2.4 INGENIERÍA

Aquí se incorpora dos tareas paralelas, el Diseño del Contenido y la Producción, son tareas llevadas a cabo por personas no técnicas del equipo IWeb. El objetivo de estas tareas es diseñar, producir y/o adquirir todo el contenido de texto, gráfico y video que se vayan a integrar en la WebApp. Al mismo tiempo se lleva a cabo un conjunto de tareas de diseño.

2.2.2.5 GENERACIÓN DE PÁGINAS

La generación de páginas es una actividad de construcción que hace mucho uso de las herramientas 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 la elaboración de páginas Web ejecutables en HTML, XML y otros lenguajes orientados a procesos (java). Durante esta actividad también se lleva a cabo la integración con el software intermedio (Middleware) de componentes (es decir: COBRA, DCa M o JavaBEan). Las pruebas ejercitan la navegación, intentan descubrir los errores de los applets, guiones y formularios, y ayuda a asegurar que la WeApp funcionará correctamente en diferentes entornas (por ejemplo, con diferentes navegadores)

2.2.2.6 EVALUACIÓN CLIENTE

En este punto es donde se solicitan cambios, tiene lugar la ampliación del ámbito. [1]

---

[1] (Pressman, 2002)

2.3 METODOLOGÍA

La Metodología es el establecimiento de teorías sobre el Método. Entonces la metodología es la descripción y el análisis de los métodos. Podríamos afirmar que la metodología es el estudio analítico y critico de los métodos de investigación y de prueba, esto incluye:

La descripción, el análisis y la valoración crítica de los métodos que conciernen a la investigación.

Lo trascendente de la metodología es que le interesa más el proceso de investigación que los mismos resultados.

En la era espacial en la cual nos desarrollamos, el bombardeo permanente de información demanda de cada uno de los sujetos la búsqueda de diferentes procedimientos y mecanismos de acción, es decir, las herramientas más idóneas para aplicar esa información y transformarla en conocimiento. La información dista mucho del significado de conocimiento, a la primera se la puede conceptuar como los datos que obtenemos por diversos medios y la segunda como la puesta en práctica de aquellos datos que h sido obtenidos.

Al hablar de metodología hacemos referencia a los diversos mecanismos' de rigor lógico-científico, que ayudan a desarrollar el conocimiento, dentro de cada una de las disciplinas científicas. La metodología une y procesa todos los componentes de las áreas del saber, de igual forma, construye sistemas que permiten llegar a los objetivos, metas, desafíos y por ende los consabidos resultados. [2]

Por el interés creciente en el desarrollo de los sistemas orientados a la web y porque estos requieren un tratamiento especial durante el proceso de desarrollo diferente al de los sistemas habituales, es necesario proponer nuevos modelos y técnicas que den soporte al equipo de trabajo asegurando la calidad del producto resultante. Con esta idea, hay diferentes metodologías que ofrecen modelos y técnicas específicas para tratar los sistemas web siendo la OOHDM la que ha sido escogida y a continuación detallaremos.

La fase de diseño para sistemas orientados a la web va a dividirse según la metodología OOHDM en tres fases fundamentalmente: fase de diseño básico, fase de diseño navegacional y fase de diseño de interfaz abstracta.

2.4 METODOLOGÍA OOHDM

2.4.1 INTRODUCCIÓN

Las metodologías tradicionales de ingeniería de software, o las metodologías para sistemas de desarrollo de información, no tienen 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 usabilidad.

Producir aplicaciones en las cuales el usuario pueda aprovechar el potencial del paradigma de la navegación de sitios web, mientras ejecuta acciones sobre bases de información, es una tarea muy difícil de lograr. 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.

Construir la interfaz de una aplicación web es también una tarea compleja; No sólo se necesita especificar cuáles son los objetos de la interfaz que deberían ser implementados, sino también la manera en la cual estos objetos interactuarán con el resto de la aplicación. En hipermedia existen requerimientos que deben ser satisfechos en un entorno de desarrollo unificado.

Por un lado, la navegación y el comportamiento funcional de la aplicación deberían ser integrados. Por otro lado, durante el proceso de diseño se debería poder desacoplar las decisiones de diseño relacionadas con la estructura

---

[2] (C. Villalba, 2004)

navegacional de la aplicación, de aquellas relacionadas con el modelo del dominio.[3]

2.4.2 FASES DE LA METODOLOGÍA OOHDM

2.4.2.1 Captura de Requerimientos

En el desarrollo de software de calidad, es esencial la captura correcta de requerimientos por parte de los diseñadores del sistema a desarrollar. Es necesario incorporar esta fase al OOHDM. Aquí, es primordial identificar a los actores (stakeholders) y las tareas que ellos van a ejecutar. Luego, los escenarios son recolectados para cada tarea y tipo de actor, produciendo los Casos de Uso, los cuales son una representación gráfica concisa de la interacción entre el usuario y el sistema durante la ejecución de una tarea.

2.4.2.2 Diseño Conceptual

En el Diseño Conceptual se realiza un modelo orientado a objetos que representa al dominio de la aplicación en estudio, consiste en realizar los pasos propios del diseño conceptual.

En una aplicación de gestión clásica manteniendo al margen el hecho de que existan necesidades de navegación e interfaz abstracta. Seleccionar la arquitectura que mejor soporte al sistema, así como los casos de uso que se hayan diseñado en fases anteriores del proceso de desarrollo. Además, habrá que obtener el modelo de clases de diseño. Lo que se obtiene por lo tanto en esta etapa es:

La arquitectura abstracta del sistema

La división del sistema en subsistemas

El diseño de los casos de uso

El modelo de clases de diseño

---

[3] http://www.unab.edu. co/e.ditorialunab/revistas/rcc/pdfs/r22 - art5 - c.pdf

Definición de Clases conceptuales

Atributos

Relaciones

Comportamiento

2.4.2.3 Diseño Navegacional

Para la Metodología OOHDM; la navegación es un paso clave en el diseño de aplicaciones web. Luego de obtener el esquema conceptual se establece la segunda fase que se denomina Diseño Navegacional**.** Un Diseño Navegacional es construido como una vista sobre un diseño conceptual, en la que se ha de definir la estructura de navegación a través del hiperdocumento mediante la construcción de modelos navegacionales de acuerdo con los diferentes perfiles de usuarios. En sí depende de las necesidades del usuario, donde lo que se realizará será organizar la información para cumplir con dichos requerimientos

El Diseño de Navegación se expresa en dos esquemas o modelos: Esquema de Clases Navegacionales y Esquema de Contextos Navegacionales. Entre las Clases Navegacionales están un conjunto de tipos predefinidos como: nodos, enlaces y las estructuras de acceso, tales como índices o recorridos guiados, que representan los posibles caminos de acceso a los nodos.

La noción de contexto navegacional es la principal estructura primitiva del espacio navegacional, el contexto es un conjunto de nodos, enlaces, clases de contextos, y otros contextos navegacionales (contextos anidados); que pueden ser definidos por comprensión, por extensión, o por enumeración de sus miembros.

Estos contextos organizan el espacio navegacional en conjuntos adecuados para que sean recorridos en un orden particular y que deberían ser definidos como caminos para ayudar al usuario a lograr la tarea deseada.

Los nodos son enriquecidos con un conjunto de clases especiales que permiten de un nodo observar y presentar atributos (incluidos las anclas), así como métodos (comportamiento) cuando se navega en un particular contexto.

El Diseño Navegacional debe ser claro y consistente para evitar la desorientación y sobresaturación de información; y tiene que estar adaptado al usuario y a la función

2.4.2.4 Diseño de Interfaz Abstracta

Esta es la tercera fase de diseño de la metodología OOHDM, donde luego de definir las estructuras navegacionales, se deben especificar los aspectos de interfaz a través de un modelo, la estructura y el comportamiento de la interface del sistema hipermedia con el usuario. Este modelo es abstracto e independiente de la implementación. Y se basa en las Interfaces Gráficas de Usuario (IGUs). En si se define 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, además qué transformaciones de la interfaz son pertinentes y cuándo es necesario realizarlas.

El modelo de la interface abstracta se expresa en tres tipos de diagramas los cuales se complementan entre sí. El primero es los diagramas de Vistas de Datos Abstractos (Abstract Data Views, ADVs), que OOHDM utiliza para modelar los aspectos estáticos de la interfaz de usuario, este incluyen una vista (ADV) por cada clase navegacional (nodo, enlace o estructura de acceso) que fue establecida en el Diseño Navegacional. Este diagrama se compone de una serie de cajas (una caja es un ADV) que representan las diferentes clases de objetos que aparecerán ante el usuario. El segundo lugar se encuentra el Diagrama de Configuración, que representa los eventos externos (provocados por el usuario, como Clic del ratón o Doble clic) que maneja un ADV, los servicios que ofrece el ADV (como "visualización") y las relaciones estáticas entre las ADVs. Y el tercer diagrama de la metodología OODHM es el Diagrama de Estado que representa el comportamiento dinámico del sistema hipermedial mediante el establecimiento de un diagrama de transición de estados para cada ADVs, en el que se reflejan los posibles estados por los que puede pasar cada objeto de la interface (*oculto, desactivado, ampliado, reducido, normal, etc.)* y los eventos que originan los cambios de estado.

El Diseño de Interfaz Abstracta, simplifica las modificaciones de “percepción” y permite múltiples vistas de un mismo esquema.

2.4.2.5 Implementación

Desarrollada la fase de diseño conceptual y navegacional, sería necesario terminar con la última fase de diseño, el diseño de interfaz abstracta, terminado este el camino a la fase de la implementación resulta más sencillo. Por un lado, al mantenerse la fase de diseño conceptual de forma similar a la propuesta por el Proceso Unificado de UML, las nuevas técnicas de diseño de software basado en métodos formales, se pueden seguir aplicando, manteniéndose así las ventajas que esto conlleva.

Además, con la aplicación de las nuevas fases de diseño, podemos facilitar la tarea del desarrollador. Al diseñar el modelo conceptual, el desarrollador conocerá qué información debe almacenar en el sistema y qué operaciones se podrán realizar sobre él. Pero si a esto le añadimos el modelo navegacional, el desarrollador sabrá además cómo se agrupará esa información a la hora de presentarla y cómo se podrá pasar de un grupo de información a otro. Con el diseño de la interfaz abstracta, el programador podrá, por último, adecuar la presentación de la información a los grupos de usuarios que se requieran.

2.4.2.6 Pruebas

En este caso a medida que se codifique las tareas, se van a ir aplicando las pruebas unitarias a cada una de ellas, con el fin de ir constatando el buen funcionamiento de los módulos, para luego poder ensamblar el sistema y así realizar las pruebas definitivas con el cliente.

2.5 UML

UML es un conjunto de herramientas, que permite modelar (analizar y diseñar) sistemas orientados a objetos. “El cual soluciona el 80% de los problemas usando tan solo el 20% de UML"

2.5.1 QUÉ NO ES UML

UML no es un método de desarrollo. No va a decir cómo pasar del análisis al diseño y de este al código. No son una serie de pasos que te llevan a producir código a partir de unas especificaciones.

UML al no ser un método de desarrollo es independiente del ciclo de desarrollo que se vaya a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

2.5.2 CÓMO NACIÓ UML

Durante los ochenta y principios de los noventa Grady Booch, James Rumbaugh, e Ivar Jacobson trabajaban por separado en desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos. Los tres llegaron por separado a obtener bastante reconocimiento.

Booch había escrito "Object-Oriented Analysis and Design with Applications" un libro de referencia en el análisis y diseño orientado a objetos desarrollando su propia notación.

Por su parte **James Rumbaugh** había desarrollado su propia notación de diseño orientado a objetos llamada OMT (Object Modeling Technique) en su libro "Object-Oriented Modeling and Design".

Por otro lado **Jacobson** se había revelado como un visionario del análisis (padre de los casos de uso) y sobre todo del diseño orientado a objetos, sorprendiendo a todo el mundo en "Object-Oriented Software Engineering: A Use Case Driven Approach".

A mediados de los noventa empezaron a intercambiar documentos y trabajar en conjunto produciendo grandes avances en el modelado de sistemas orientados a objetos.

En 1994 Rational contrató a Rumbaugh en donde ya trabajaba Booch, un año después Jacobson se unía a ellos en Rational.

En 1997 salió a la luz la versión 1.0 de UML

2.5.3 BASES DE UML

UML Semantics.

Define la semántica y sintaxis del UML. Para ello utiliza tres visiones consistentes:

Abstract syntax: Para presentar el meta-modelo UML, sus conceptos (meta-clases), relaciones, y restricciones se utilizan diagramas de clases.

Well-formedness rules: Definen las reglas y restricciones que rigen en los modelos válidos. Las reglas se expresan en lenguaje natural y en OCL (**O**bject **C**onstraint **L**anguage). OCL es un lenguaje de especificación que utiliza lógica para especificar propiedades invariantes de sistemas que se componen de conjuntos y relaciones entre conjuntos.

Semantics: Las semánticas de manejo del modelo se describen en lenguaje natural.

Estas tres visiones constituyen la definición del UML. Se han creado bajo la idea de que una definición más formal conllevaría expresiones matemáticas que poca gente podría entender de una forma directa.

NOTA: Un meta-modelo es un lenguaje para la especificación de un modelo, el modelo propósito. Es decir, es un modelo para elementos de modelado. El propósito del meta-modelo UML es el proporcionar una declaración común y única de la sintaxis y semánticas de los elementos del UML. La presencia de dicho meta-modelo ha hecho posible que los desarrolladores se hallan puesto de acuerdo en las semánticas, además de permitir explorar formas de simplificar el lenguaje de modelado unificando los elementos del UML.

UML Notation Guide.

Define la notación y proporciona ejemplos. La notación será la sintaxis gráfica para expresar la semántica descrita por el meta-modelo UML. La notación gráfica y la sintaxis textual son las partes más visibles, siendo utilizadas por personas y herramientas en el modelado de sistemas. También contiene las semánticas UML, sin embargo, sus definiciones se encuentran en *UML Semantics*.

UML Extension for the Objectory Process for Software Engineering y **UML Extension for Business Modeling**.

Extensiones específicas de UML, *Objectory Process* y *Business Engineering*. El UML es ampliamente aplicable sin extensiones, de forma que las compañías y proyectos sólo deberían definir extensiones cuando es necesario el introducir nueva notación y terminología. Para reducir la confusión se definen:

UML Variant: Un lenguaje con una semántica bien definida que se construye encima del meta-modelo UML. Este lenguaje puede ser una especialización del meta-modelo UML, sin cambios en las semánticas del UML o con redefinición de algunos de sus elementos.

UML Extensión.

Un conjunto predefinido de estereotipos, valores etiquetados, restricciones e iconos de notación que colectivamente extienden y adaptan el UML a un dominio o un proceso específico, por ejemplo, *Objectory Process Extension*.

> **Object Constraint Language Specification (OCL)**: El UML incorpora el lenguaje de restricción de objetos a fin de superar las deficiencias que poseen los elementos UML para definirse a sí mismos. Podría haberse utilizado el lenguaje natural, pero su obvia ambigüedad provocó la necesidad de la incorporación del OCL. Se puede decir que se trata de un lenguaje de expresión, porque se utiliza únicamente para expresar un valor, sin provocar cambios en el modelo; de un lenguaje de modelado, porque no es ejecutable, no es un lenguaje de programación; y de un lenguaje formal, porque todos sus elementos han sido definidos formalmente.

Según los autores, este lenguaje está destinado a la especificación de invariantes en clases y tipos de clases en el modelo de clases; para describir pre y post-condiciones en operaciones y métodos; para describir guardas; para ser empleado como un lenguaje de navegación; para definir reglas de formación; y para definir restricciones de operaciones.

2.5.4 HERRAMIENTAS UML

Pero volviendo a la definición de UML como "*conjunto de herramientas*", si nos imaginamos UML como una caja de herramientas con su martillo, destornillador, alicates, etc. Veamos qué contiene nuestra caja de herramientas:

Diagrama de casos de uso.

Diagrama de clases.

Diagrama de estados.

Diagrama de secuencias.

Diagrama de actividades.

Diagrama de colaboraciones.

Diagrama de componentes.

- Diagrama de despliegue.

2.5.4.1 Diagrama de casos de uso

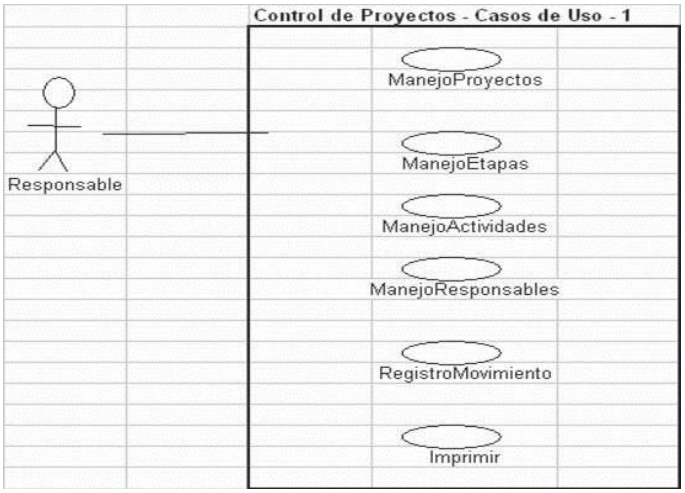
El diagrama de casos de uso modela el comportamiento esencial del sistema en términos de interacción de agentes externos (actores) que interactúan con el sistema.

Se identifican distintos actores, según el acceso a los módulos del programa. Son los siguientes:

Esta representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema. Los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los casos de uso. Se pueden visualizar como las funciones más importantes que la aplicación puede realizar o como las opciones presentes en el menú de la aplicación.

GRÁFICO 2. ACTORES

Operador Contable Colegiado



ELABORADO: Holger Cruz

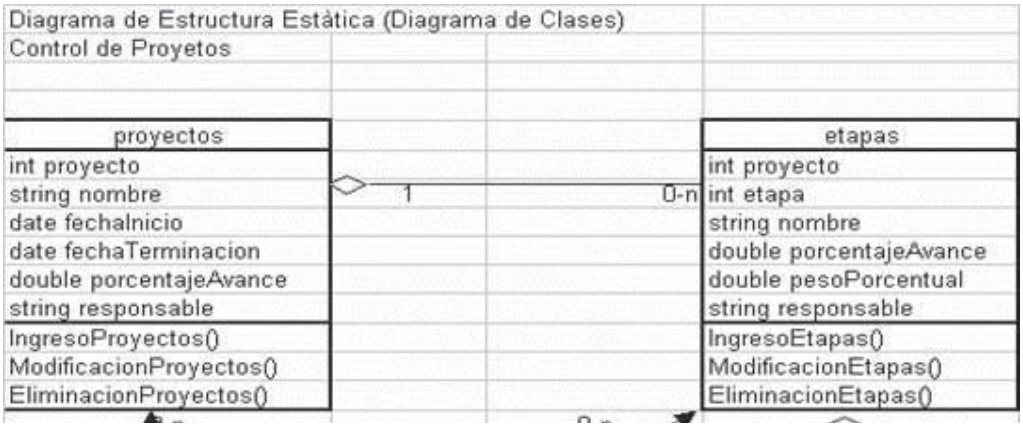
2.5.4.1.1 Subcasos de Uso

Hacen referencia a la descomposición de los casos de uso del punto anterior. Se dan cuando existe una relación entre dos casos de uso. Dicha relación puede ser de extensión, que en términos de la Orientación a Objetos es una relación de herencia, donde el "subcaso" especializa al caso. También puede ser una relación de "uso", donde el caso requiere que el subcaso se realice completamente para que él mismo se realice bien y completamente.

2.5.4.2 Diagrama de clases

Nos muestra una vista de la aplicación en un determinado momento, es decir, en un instante en que el sistema está detenido. Las clases son la plantilla de los objetos, y aquí podemos ver representados a estos con sus atributos o características y su comportamiento o métodos, así como la relación entre ellas.

GRAFICO 3. EJEMPLO DEL DIAGRMA DE CLASES



FUENTE:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295.asp#authorbrief

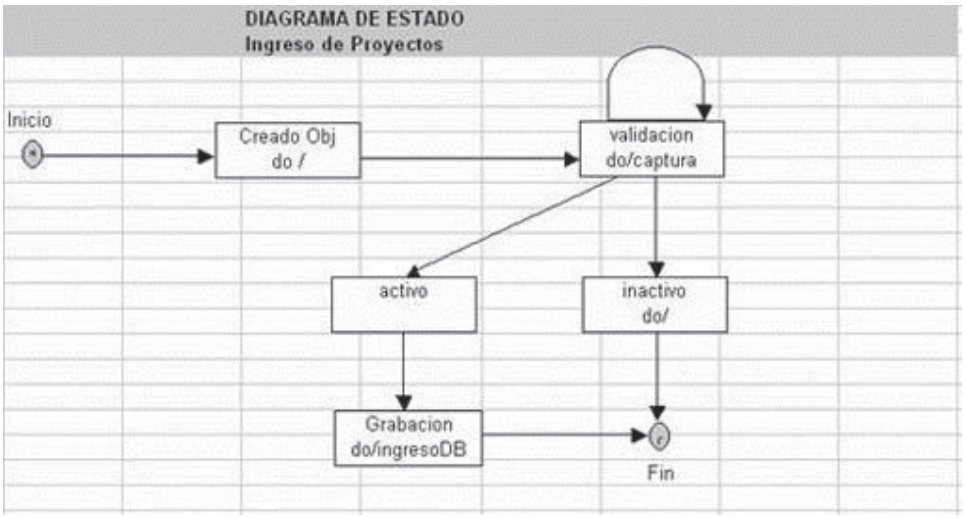
2.5.4.3 Diagrama de estados

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.), por ejemplo una validación de una captura.

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 le estado inicial y el final, definiendo a qué eventos puede responder el objeto, y qué transacciones ejecutará.

GRAFICO 4. EJEMPLO DEL DIAGRAMA DE ESTADOS

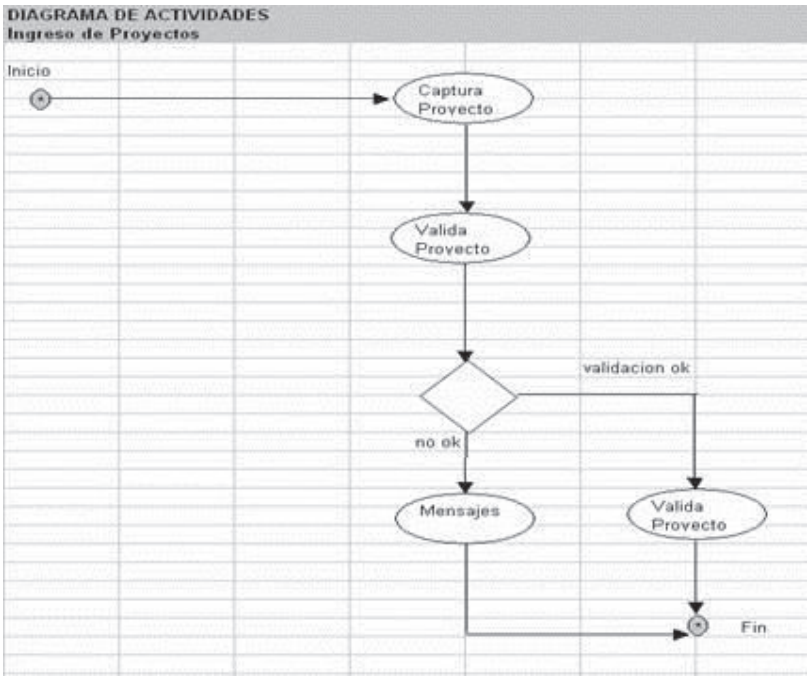


FUENTE:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295.asp#authorbrief

2.5.4.4 Diagrama de actividades

Se utilizan para visualizar, especificar, construir y documentar la dinámica de un conjunto de objetos o simplemente para modelar el flujo de control de una operación (método de una clase). Fundamentalmente es un Diagrama de Flujo que muestra el flujo de control entre las actividades.

GRAFICO 5. EJEMPLO DEL DIAGRAMA DE ACTIVIDADES

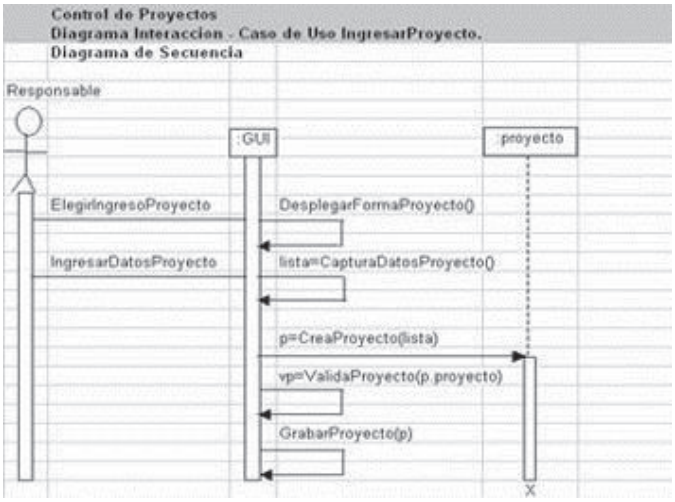


FUENTE:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295.asp#authorbrief

2.5.4.5 Diagrama de secuencia

Los Diagramas de Secuencia representan una interacción entre objetos de manera secuencial en el tiempo. Muestra la participación de objetos en la interacción entre sus “líneas de vida” (desde que sé instancias) y los mensajes que ellos organizadamente intercambian en el tiempo. El responsable o ACTOR es quien inicia el ciclo interactuando inicialmente con la interfaz de usuario: GUI; enseguida se inician todos los objetos que intervienen en el funcionamiento del aplicativo. En este diagrama se comienza a observar el comportamiento del sistema a partir de los eventos generados por los actores. Aquí sé interactúa con instancias, no con clases.

GRAFICO 6. EJEMPLO DEL DIAGRAMA DE SECUENCIA

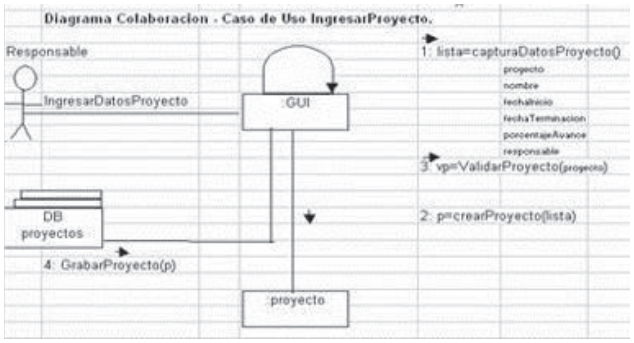


FUENTE:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295.asp#authorbrief

2.5.4.6 Diagrama de colaboración

Los Diagramas de Colaboración dan todas las especificaciones de los métodos. Estos permiten describir una operación específica incluyendo sus argumentos y variables locales creadas durante su ejecución. Se muestran los objetos y mensajes que son necesarios para cumplir con un requerimiento o propósito, o con un conjunto de ellos.

GRAFICO 7. EJEMPLO DEL DIAGRAMA DE COLABORACIÓN



FUENTE:
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_2295.asp#authorbrief

2.5.4.7 Diagrama de componentes

Los diagramas de Componentes permiten visualizar las partes de un sistema, mostrando las diversas formas en que pueden ensamblarse para construir ejecutables. Es decir este diagrama modela la vista estática de los sistemas, osea los componentes y sus conexiones, y no como funcionan.

Siempre que se construya un diagrama de componentes se debe pensar en que estamos modelando una dimensión física del software; esto es, cómo los componentes se almacenan en archivos de disco. Un componente de software constituye en una entidad real y no solo en un concepto. También en este diagrama, debemos mostrar la realización de un conjunto bien definido de interfaces e implementar las clases que trabajan juntas para proveer dichas interfaces.

2.5.4.8 Diagrama de despliegue

El Diagrama de despliegue, modela la topología del hardware sobre el cual correrá nuestra aplicación y nos indica en donde se ejecutará cada uno de nuestros componentes; esto es, muestra las relaciones físicas entre los componentes de software y el hardware de nuestro sistema.

Estos muestran la forma en que físicamente lucirá nuestro sistema, sólo deben mostrarse los nodos y componentes que no existen en tiempo de ejecución no se muestran en este diagrama, sin embargo, pueden ser mostrados en sus respectivos Diagramas de Componentes.

CAPITULO 3

LAS HERRAMIENTAS

3.1 INTRODUCCIÓN

Las Herramientas son fundamentales para complementar el desarrollo de una aplicación. Una vez terminado el análisis del diseño, aplicando toda la metodología, se procede aplicar el diseño y ahí es donde intervienen las herramientas, las cuales son el siguiente paso para poner en ejecución las aplicaciones.

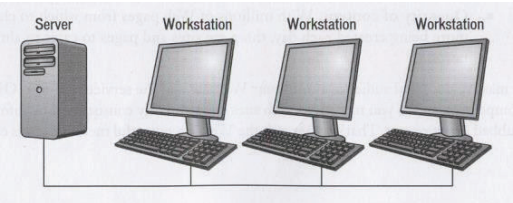
3.1.1 INTERNET

3.1.1.1 Definición

Se puede definir al Internet simplemente como la más grande red de computadoras del mundo.

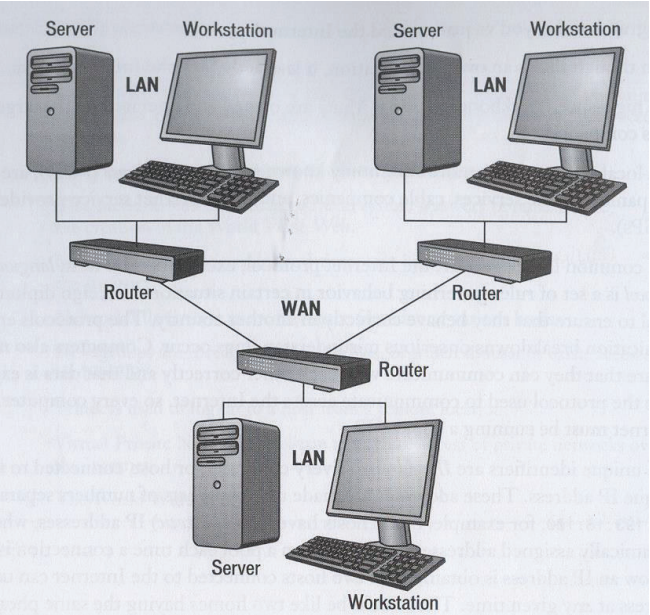
Las redes funcionan actualmente en la mayoría de las oficinas llamadas *local area networks* (LAN), compuesta de un grupo de computadoras relativamente cercanas, enlazadas por un hardware y cableado especial. Algunas computadoras son clientes o también llamadas workstations; otras son los servidores llamados también servidor de archivos. Todas estas computadoras pueden comunicarse con otras para compartir información. El siguiente gráfico muestra una red local.

GRAFICO 8. RED LAN PEQUEÑA ENLAZADA POR CABLES ESPECIALES



FUENTE: Marcomedia Coldfusion MX

GRAFICO 9. RED WAN, FORMADA O INTERCONECTADA POR MÚLTIPLES LANS



Fuente: Macromedia Coldfusion MX

El Internet se puede representar por una sola WAN a escala mundial, la cual puede interconectar varias WANs o LANs a la vez. El lenguaje común que utiliza es el IP (Internet Protocol). UN protocolo es un conjunto de reglas que sirven para controlar situaciones; por cuya razón las computadoras también necesitan protocolos para que entre ellas se puedan comunicar correctamente y los datos sean trasladados correctamente. IP es el protocolo usado para la comunicación a través del Internet, por lo tanto cada computadora conectada al Internet debe correr una copia del protocolo IP. La única identificación son las direcciones IP de cada computadora o host, las cuales son únicas en el Internet. Estas direcciones son compuestas por cuatro conjuntos de números separados por puntos, por ejemplo 208.193.16.100; tal cual se utiliza para direcciones comunes, como números telefónicos o direcciones de calles.

3.1.2 INTRANET

3.1.2.1 Definición

¿Qué es exactamente la Intranet?

En un artículo sobre la implementación de Intranet corporativo, Wagner, Cheng y Baratz (2002) explican el Intranet de una manera general como "una red privada de computadoras que hablan con otras a través de un protocolo o lenguaje común…"[4] . Una red privada quiere decir que únicamente puede ser accesada por miembros de la organización a la cual pertenece esa Intranet; al decir que hablan con otras a través de un protocolo común nos dice que las computadoras de esa red se puede comunicar entre ellas compartiendo información utilizando un mismo lenguaje.

En general la Intranet, así como las redes, pueden estar formadas por únicamente dos computadoras así como por miles de ellas, que pueden estar ubicadas en el mismo lugar o inclusive hasta en diferentes países.

Muchas empresas que están pensando seriamente el hacer uso de esta poderosa tecnología de información y muchas otras que ya están disfrutando de sus beneficios tuvieron que contestar preguntas importantes.

Una de las preguntas que se deben tener en cuenta es la que se refiere a la competencia, clientes y proveedores, si ellos están ya haciendo uso de esta tecnología, la respuesta más común es afirmativa, y aún y que en algunos casos no fuera así, se debe pensar que con el crecimiento tan rápido que está teniendo la Intranet pronto todos van a estar involucrados, y quedar fuera implicaría estar obsoleto.

Otro pregunta importante es si el uso de esta nueva tecnología es segura, si es posible utilizar una red pública como el Internet para aplicaciones de negocios sin tener el control de que o como funciona. Un ejemplo que puede ayudar a responder esta pregunta es el uso del teléfono y del fax que son redes públicas

---

[4] http://www.gestiopoils.com/canales2/gerencia/1/#1

muy parecidas al Internet, y que son utilizados para sacar adelante negocios importantes. Como sabemos la seguridad es muy importante, sin embargo las redes públicas son utilizadas para transacciones importantes.

Y como último la pregunta esperada, ¿se puede esperar un rápido y suficiente retorno de inversión? En realidad los costos para la implementación de la Intranet no son muy altos y sus retornos de inversión suelen ser rápidos y valen la pena, pero se debe ser cuidadoso al implementar un proyecto de Intranet, cuidados en no excederse el tiempo de implementación que se tiene planeado, al momento de hacer la elección de la tecnología a utilizar.

Existen varias razones para empezar a hacer uso del Internet, pero hay dos importantes que menciona en uno de sus artículos Blanc (1998), “Primero, la Intranet puede soportar tus iniciativas de reingeniería y puede mostrar el extra de cualquier análisis costo-beneficio…. La segunda razón es que la Intranet representa un cambio de tecnología fundamental la cual puede convertirse en una herramienta importante de negocios...”[5]

Resumen

Como se ha visto la Intranet puede presentar varios beneficios, aunque casi siempre los ejecutivos buscan encontrar el beneficio que puede traer monetariamente hablando, es decir, el retorno de inversión, (ROI por sus siglas en inglés “Return on investment”), el cual se puede medir de acuerdo al incremento en el nivel de calidad de la comunicación y en cómo se ha logrado reemplazar las formas tradicionales de comunicación, esto se puede ver ya que con la Intranet se logra una gran reducción de las formas tradicionales de comunicación, como lo son los impresos y la comunicación persona a persona, los cuales ocasionan una reducción en los costos.

[5] http://www.gestiopoils.com/canales2/gerencia/1/#2

3.2 INGENIERIA WEB

3.2.1 INTRODUCCIÓN

La Ingeniería de la Web es la aplicación de metodologías sistemáticas, disciplinadas y cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web. [6]

En 1998, Roger Pressman moderó una mesa redonda virtual con representantes la ingeniería software tradicional y del desarrollo software basado exclusivamente en Internet. El debate principalmente se centró en discutir si valía la pena aplicar un proceso de ingeniería a las aplicaciones con base en Internet, o qué características tenían éstas que justificaran el no utilizarlo. La conclusión general fue que aplicar un proceso de ingeniería nunca es una mala idea pero que éste debería adaptarse a los requerimientos de cambio continuo y rapidez siempre presentes en el proceso de desarrollo Web. De iniciativas como ésta y de otras como la organización de congresos y talleres especializados en el desarrollo para la Web, surge el nacimiento de una nueva disciplina denominada Ingeniería Web.

Es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad.

3.2.2 EL PROCESO DE INGENIERÍA WEB

Las actividades que forman parte del proceso son: *formulación, planificación análisis, modelización, generación de páginas, test* y *evaluación del cliente.*

3.2.2.1 La Formulación

Identifica objetivos y establece el alcance de la primera entrega.

---

[6] http://es.wikipedia.org/wikillngenier%C3%ADa_de_la_Web

3.2.2.2 La Planificación

Genera la estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega. El Análisis especifica los requerimientos e identifica el contenido.

3.2.2.3 La Modelización

Se compone de dos secuencias paralelas de tareas. Una consiste en el diseño y producción del contenido que forma parte de la aplicación. La otra, en el diseño de la arquitectura, navegación e interfaz de usuario. Es importante destacar la importancia del diseño de la interfaz. Independientemente del valor del contenido y servicios prestados, una buena interfaz mejora la percepción que el usuario tiene de éstos.

3.2.2.4 La Generación de páginas

Se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de la aplicación, las páginas.

3.2.2.5 El Test.

Busca errores a todos lo niveles: contenido, funcional, navegacional, rendimiento, etc. El hecho de que las aplicaciones residan en la red, y que ínter operen en plataformas muy distintas, hace que el proceso de test sea especialmente difícil.

3.2.2.6 Finalmente

El resultado es sometido a la evaluación del cliente.

3.3 DISEÑO DE PÁGINAS WEB

El diseño web es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del diseño convencional sobre Internet ya que requiere tener en cuenta cuestiones tales como navegabilidad, interactividad, usabilidad, arquitectura de información y la interacción de medios como el audio, texto, imagen y vídeo.

3.3.1 ETAPAS

Para el diseño de páginas web debemos tener en cuenta dos etapas:

La primera es el diseño visual de la información que se desea editar. En esta etapa se trabaja en el papel distribuyendo el texto, los gráficos, los vínculos a otros documentos y otros objetos multimedia que se consideren pertinentes.

La segunda, una vez que se tiene este boceto se pasa al programa la página web. Para esto, y fundamentalmente para manejar los vínculos entre documentos, se creó el lenguaje de marcación de hipertexto o HTML. Los enlaces que aparecen subrayados en este documento y otros de Wikipedia son ejemplos de hipertexto, puesto que al pulsar sobre ellos conducen a otras páginas con información relacionada.

3.4 PROGRAMACIÓN POR CAPAS

3.4.1 CAPAS O NIVELES

3.4.1.1 Capa de Presentación

Es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

3.4.1.2 Capa de Negocio

Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

3.4.1.3 Capa de Datos

Es donde residen los datos. Está formada por uno o más gestor de bases de datos que "realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

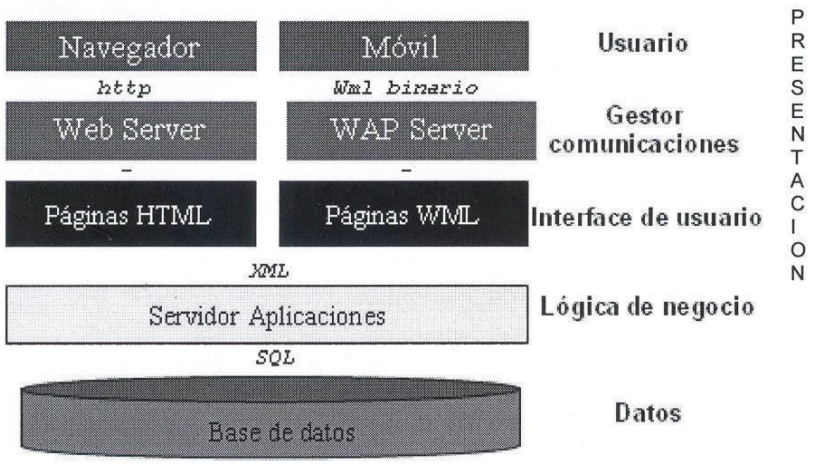
Todas estas capas pueden residir en un único ordenador (no sería lo normal), si bien lo más usual es que haya una multitud de ordenadores donde reside la capa de presentación (son los clientes de la arquitectura cliente / servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de datos, y otra serie de ordenadores sobre los cuales corre la base de datos. En una arquitectura de tres niveles, los términos "capas" y "niveles" no significan lo mismo ni son similares.

El término "capa" hace referencia a la forma como una solución es segmentada desde el punto de vista lógico:

Presentación/ Lógica de Negocio/ Datos.

GRAFICO 10. ARQUITECTURA EN TRES CAPAS



Fuente: http://es .wikipedia.org/wiki/Arquitectura_de_tres_niveles

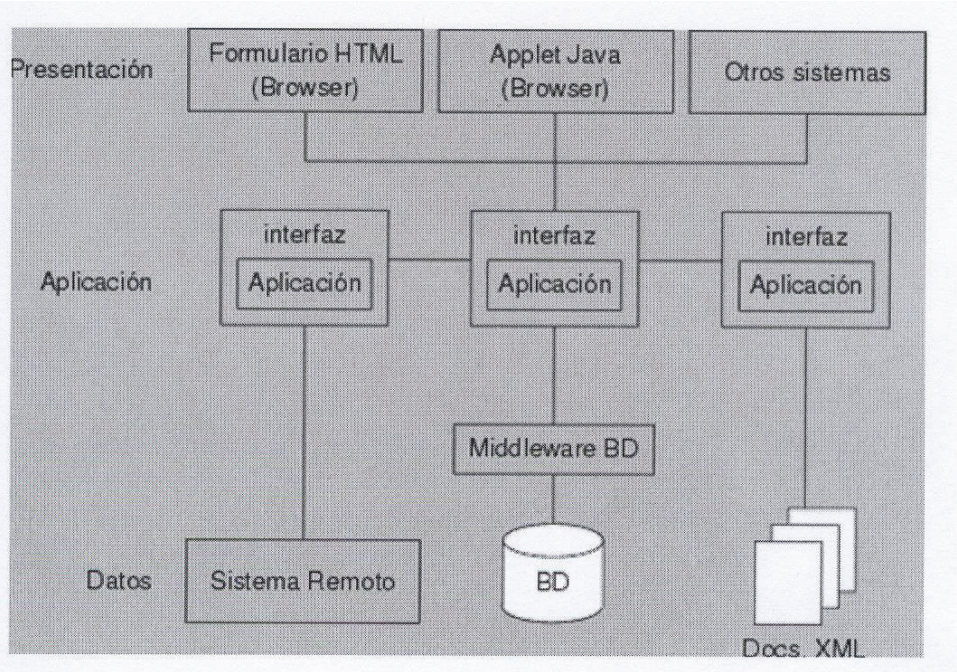
En cambio, el término "nivel", corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

Una solución de tres capas (presentación, lógica, datos) que residen en un solo ordenador (Presentación+lógica+datos). Se dice, que la arquitectura de la solución es de tres capas y un nivel.

- Una solución de tres capas (presentación, lógica, datos) que residen en dos ordenadores (presentación+lógica, lógica+datos). Se dice que la arquitectura de la solución es de tres capas y dos niveles.
- Una solución de tres capas (presentación, lógica, datos) que residen en tres ordenadores (presentación, lógica, datos). La arquitectura que la define es: solución de tres capas y tres niveles.[7]

[7] http://es.wikipedia.org/wiki/Arquitectura_de_tres _niveles

GRAFICO 11. ARQUITECTURA MULTINIVEL



Fuente: http://www.infor.uca.es/-jvegas /cursos/Buendía/pordocente/node21.html

3.5 HERRAMIENTAS PARA LA CONSTRUCCIÓN WEB

3.5.1 TECNOLOGÍA

3.5.1.1 HTML

El HTML fue desarrollado originalmente por Tim Berners-Lee mientras estaba en el CERN, y fue popularizado por el navegador Mosaic desarrollado en el NCSA. Durante los años 90 ha proliferado con el crecimiento explosivo de la Web. Durante este tiempo, el HTML se ha desarrollado de diferentes maneras. La Web depende de que los autores de páginas Web y las compañías compartan las mismas convenciones de HTML. Esto ha motivado el trabajo colectivo en las especificaciones del HTML.

Para publicar información y distribuirla globalmente, se necesita un lenguaje entendido universalmente, una especie de lengua franca de publicación que todas las computadoras puedan comprender potencialmente. El lenguaje de publicación usado por la World Wide Web es el HTML (acrónimo de HyperText Markup Language, Lenguaje para el Formato de Documentos de Hipertexto).

El HTML da a los autores las herramientas para:

- Publicar documentos en línea con encabezados, textos, tablas, fotos, etc.
- Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.
- Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.
- Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.

Cada estructura de texto se encerrará entre una marca de inicio y otra de fin.

Las marcas vienen delimitadas con los signos <(inferior) y >(superior); el final precede por el símbolo /. De este modo el navegador sabe que debe interpretar

código comprendido entre estos símbolos. Los códigos pueden escribirse en procesadores de texto (Word, WordPerfect, Notepad, etc.) que definen el formato de un texto ASCII, y será mostrado en el browser.

Un documento escrito en HTML debe contener los siguientes comandos y la estructura de un documento HTML es la siguiente:

<HTML> Indica el inicio del documento

<HEAD> Inicio de la cabecera

<TITLE> Inicio del título del documento

</TITLE> Final del título del documento

</HEAD> Final de la cabecera del documento

<BODY> Inicio del cuerpo del documento

</BODY> Final del cuerpo del documento

</HTML> Final del documento.

Fuente: (http://www.ilustrados.com/publicaciones/EpyupElVpEQUDHjXOR.php)

3.5.2 COLDFUSION

3.5.2.1 Introducción

ColdFusion es una interfaz creada por Allaire para acceder a bases de datos desde el Web. Es una potente herramienta para realizar las funciones de acceso a la información alojada en bases de datos, utilización de programación personalizada, y presentación de la información utilizando formatos muy avanzados.

Mediante el uso de esta herramienta, se puede distribuir información al nivel de Internet y/o de intranets, ya que permite conectar una base de datos al interior de una red (Intranet) o al nivel de redes más amplias en el Web (Internet).

La conexión con la base de datos es realizada haciendo uso mínimo de programación, generando posteriormente las páginas Web de manera dinámica, cuyo contenido será la información que está alojada en la base de datos. Así mismo, permite introducir nueva información dentro de una base de datos, tener

acceso a datos actualizados periódicamente, automatizando toda la actividad relacionada con dicha base de datos.

La instalación de ColdFusion es muy sencilla y similar a la de otros programas que usan una interfaz gráfica estándar, gracias a que ColdFusion brinda la ayuda necesaria para instalar el software de una forma correcta, aunque el usuario no esté familiarizado con este tipo de productos. El mismo programa de instalación detectará por si solo, con qué tipo de Servidor HTTP se cuenta.

3.5.2.2 Características y funciones

ColdFusion, según Macromedia, combina un lenguaje intuitivo, basado en tags, rico, con herramientas visuales y un servidor de aplicaciones web probadamente confiable, para entregar la manera más rápida de desarrollar poderosas aplicaciones web.

ColdFusion es una herramienta que corre en forma concurrente con la mayoría de los servidores web de Windows, Linux y Solaris (también en servidores web personales en Windows 98 y puede ser usado para intranets). El servidor de aplicaciones web de ColdFusion trabaja con el servidor HTTP para procesar peticiones de páginas web. Cada vez que se solicita una página de ColdFusion, el servidor de aplicaciones ColdFusion ejecuta el script o programa contenido en la página.

ColdFusion es un lenguaje de programación, que puede crear y modificar variables igual que en otros que nos son familiares. Posee controles de flujo de programas, como IF, Switch Case, Loop, etc. Tiene muchas funciones built-in (*Un periférico o un dispositivo que se fabrican como pieza de la computadora, no son agregados por el usuario). Un conductor de software que se combina en el núcleo de un system de funcionamiento como parte del OS, no es agregado por el usuario*)para realizar tareas más complicadas como averiguar que día caerá el 3 de Agosto del 2007 "DayOfWeekAsString(DayOfWeek('2007/08/03'))".

No es un lenguaje de bases de datos, pero interactúa de manera simple con bases de datos (Sybase, Oracle, MySQL, SQL, o Access). Usando SQL estándar,

las páginas y aplicaciones web pueden fácilmente recuperar, guardar, formatear y presentar información dinámicamente.

ColdFusion es un lenguaje basado en tags como CFML (ColdFusion Markup Language). Muchas de las funciones poderosas de ColdFusion, como leer y escribir desde o en discos duros del servidor, son basadas en tags. Así como el tag <Table> puede tener argumentos como 'width' o 'align', el tag <CFFILE> tiene argumentos que especifican 'action=read/write/copy/delete', path=' etc.

ColdFusion integra tecnologías. El tag <CFFORM> construirá automáticamente todo el código JavaScript para verificar los campos requeridos antes de hacer submit al form. ColdFusion también tiene tags para COM, Corba y Applets y Servlets de Java.

Es escalable. ColdFusion fue diseñado para desarrollar sitios complejos y de alto tráfico. A veces, el problema más grande para un diseñador web es que su sitio se vuelve popular. ColdFusion está diseñado para correr en máquinas multiprocesador, y permite construir sitios que pueden correr en clusters de servidores.

Es un lenguaje Server-side. A diferencia de JavaScript y Applets Java, que corren en el cliente o en browsers, ColdFusion corre en el servidor web. Esto significa que los scripts escritos en ColdFusion correrán de la misma manera en cualquier browser.

ColdFusion centra su potencialidad en la confiabilidad y el control del manejo de datos. Reconoce la complejidad del manejo e interacción de escritos CGI, ofreciendo una potente seguridad, veloz carga de datos, procesamiento rápido de escritos CGI que posibilita el cumplimiento de tareas de entrada o devolución de datos.

Utiliza fuentes de datos ODBC de 32-bits, las cuales deberán cumplir con el nivel 1 de los ODBC API y soportar las sentencias SQL.

Entre las *funciones* de ColdFusion están:

- Sirve a cualquier requisición de datos una vez cuente con la instalación y configuración de las fuentes de datos ODBC de 32-bits.
- Detecta errores producidos por la mala configuración o por el registro completo de la bitácora del servidor SQL u ORACLE.
- Funciona correctamente en una máquina remota. Se ejecuta sin problemas en el Microsoft Internet Information Server, aún teniendo gran cantidad de solicitudes. Gracias a ello brinda un correcto funcionamiento tanto en Internet como en Intranets.
- Provee de ayuda para la configuración que permita generar páginas HTML en forma dinámica.
- Crea estructuras condicionales dinámicamente para personalizar la solicitud de datos y el envío de los mismos hacia el cliente. Así mismo, diseña cadenas de datos para crear dinámicamente menús desplegables y para llenar listas de selección y listas de documentos.

Para crear aplicaciones de ColdFusion, se necesita del conocimiento previo de sentencias SQL u ORACLE para la generación de código en la selección de la correcta información alojada en una base de datos. Gracias a las sentencias SQL se tiene un control completo sobre qué, dónde y por qué desplegar los datos dentro de un sitio Web.

3.5.2.3 ¿Cómo funciona Coldfusion?

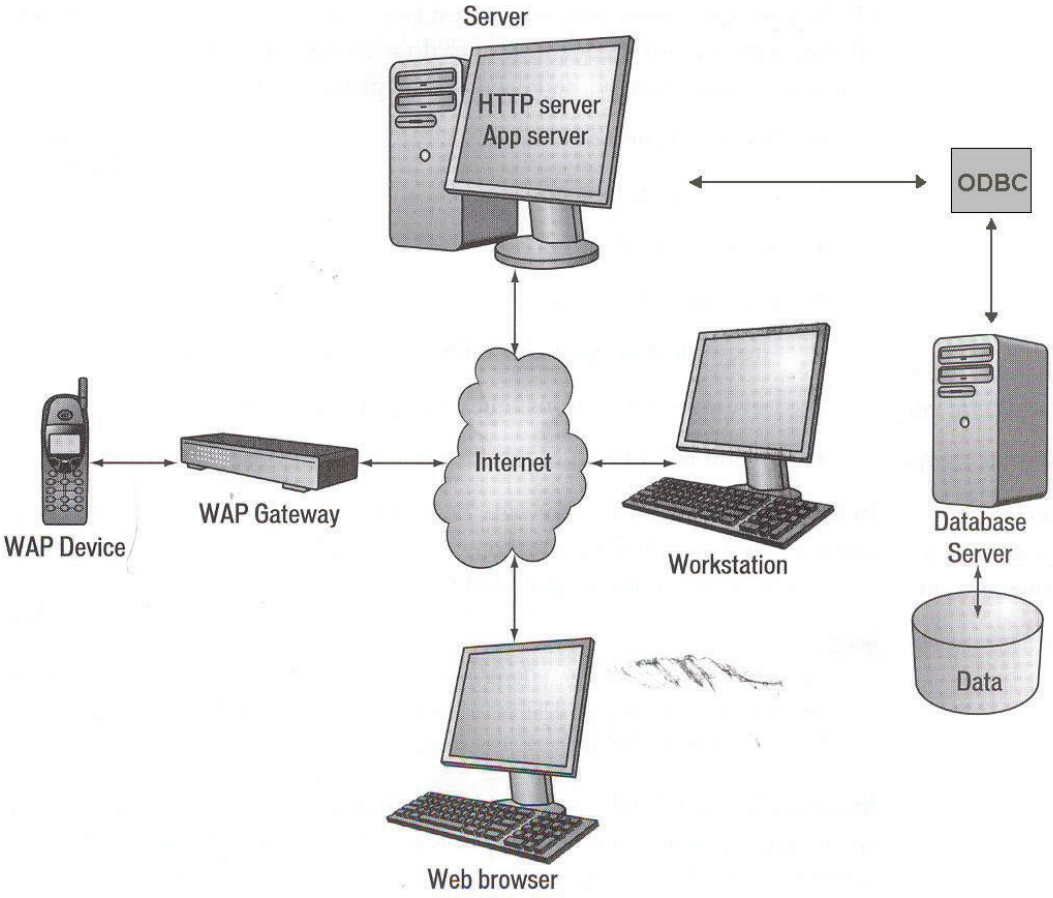
Una vez se ha realizado la instalación de este paquete, se pueden realizar requisiciones a través de un URL, las cuales son enviadas al servidor Web, y éste a su vez la hace a la interfaz de ColdFusion, la que se conecta a una fuente de datos ODBC, a la cual solicita los datos que requiere extraer de la base de datos.

Como puede verse, ColdFusion utiliza fuentes de datos ODBC, de las que incluye una versión dentro del software de instalación, para poder manipular la información dentro de las bases de datos.

Una vez se ha obtenido la información que se ha solicitado, la interfaz envía los datos hacia el Servidor Web y éste al browser, en donde los mismos son desplegados gráficamente.

En la siguiente figura se muestra el proceso que sigue ColdFusion al momento de recibir y responder a una requisición.

GRAFICO 12. ARQUITECTURA DE COLD FUSION PARA ACCEDER BASES DE DATOS EN EL WEB.



Fuente: Macromedia Coldfusion MX

3.5.2.4 Compatibilidad de ColdFusion

Para mostrar la compatibilidad y requerimientos de hardware, se ha elabora el siguiente esquema:

CUADRO 2. CUADRO DE COMPATIBILIDAD DE COLDFUSION

Windows

ColdFusion Product Editions	Enterprise, Standard, Developer
Processor	32-bit x86 processors (Intel Pentium II or AMD Athlon class peformance or better)
Operating System	Microsoft Windows 2003 Web Edition, Standard Edition, Enterprise Edition (SP1 or R2) Microsoft Windows 2000 Advanced Server, Datacenter Server, Server (SP3 and higher) Microsoft Windows 2000 Professional (SP3 and higher) [1] Microsoft Windows XP Professional, XP Home [1]
Memory	512 MB RAM
Hard Disk Space	500 MB
	Linux
ColdFusion Product Editions	Enterprise, Standard, Developer
Processor	32-bit x86 processors (Intel Pentium II or AMD Athlon class peformance or better)
Operating System	Red Hat Enterprise Linux AS & ES 3.0 or 4.0 SuSE Linux Enterprise Server 8, 9 TurboLinux 8 Server (Japanese only)
Memory	512 MB RAM
Hard Disk Space	500 MB
	UNIX
ColdFusion Product Editions	Enterprise, Developer
Processor	SPARC POWER/3 processor
Operating System	Sun Solaris 8, 9, or 10 IBM AIX 5L, 5.1, 5.3 [2]

Memory	512 MB RAM
Hard Disk Space	500 MB

Macintosh

ColdFusion Product Editions	Enterprise, Standard, Developer
Processor	PowerPC
Operating System	Apple Mac OS X 10.3.9, 10.4.2 and 10.4.3
Memory	512 MB RAM
Hard Disk Space	500 MB

Supported Application Servers

- Macromedia JRun 4 (Updater 4 and higher)
- IBM WebSphere Application Server, Version 4, 5.0, 5.0.2, 5.1
- IBM WebSphere Application Server – Network Deployment, Version 5.0, 5.0.2, 5.1
- IBM WebSphere Application Server for Developers 5.0
- BEA WebLogic Server 7, 8.1
- Oracle Application Server 10g

Java Virtual Machine

- Tested and certified with Sun JDK 1.4.2_05, 1.4.2_07, 1.4.2_09, and 1.4.2_11.

Fuente: Análisis bajado de www.adobe.com

Ampliando un poco más el panorama sobre los requerimientos que necesita la aplicación, vemos las siguientes características:

Mínimo en memoria: 512 MB (Live websites) y 256 MB (development purposes). Pero se recomienda 1GB o mayor.

Para los requerimientos del navegador: Se recomienda usar las últimas versiones de Microsoft Internet Explorer, Firefox, o Opera, en Windows. Otros navegadores

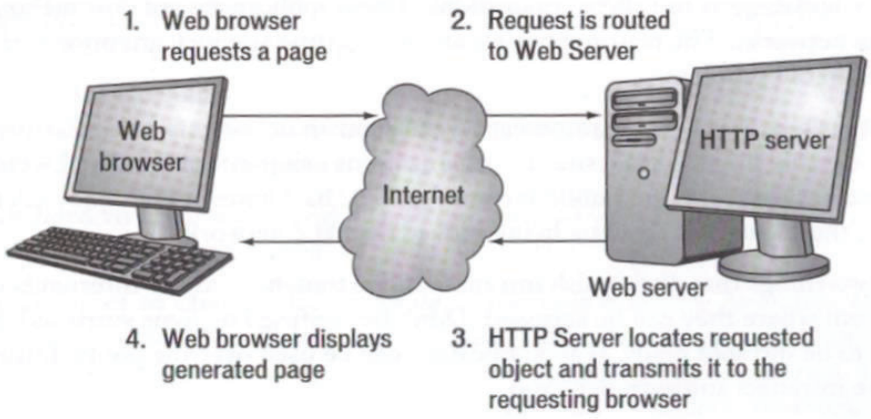
o sistemas operativos pueden ser soportados, pero no han sido probados por adobe.

3.5.2.5 Componentes ColdFusion

3.5.2.5.1 Tecnología del Servidor Web

En el corazón de cada aplicación de ColdFusion hay un servidor ColdFusion, el cual, combina una arquitectura abierta y extensible que se integra fácilmente con sistemas existentes, así como también con aplicaciones built-in y servicios de infraestructura que ayudan a presentar la información de manera elegante y lograr un alto nivel de desempeño y confiabilidad.

GRAFICO 13. EL SERVIDOR WEB HACE UNA PETICIÓN AL WEB APLICATION SERVER, EL CUAL RETORNA EL RESULTADO AL MISMO Y ESTE LO DESPLIEGA.



Fuente: Macromedia Coldfusion MX

3.5.2.5.2 Herramientas de desarrollo

Macromedia ofrece dos herramientas para el desarrollo. Para desarrolladores web, ColdFusion provee una tecnología poderosa de edición de código.

Para diseñadores y desarrolladores web Dreamweaver UltraDev ofrece el mejor ambiente visual de desarrollo. La oferta combinada de ColdFusion 7 y UltraDev 4 Studio, ayuda a mejorar la productividad, asegurar la calidad de la aplicación y

diseñar sofisticados sitios web aprovechando la integración con el servidor ColdFusion.

3.5.2.5.3 Ambiente de programación

ColdFusion soporta un poderoso lenguaje de scripting en el lado del servidor, ColdFusion Markup Language (CFML), que es extremadamente fácil de aprender y se integra limpiamente con todos los lenguajes y tecnologías web populares. ColdFusion trabaja con múltiples arquitecturas a través de la integración de COM, CORBA y EJB. También puede ser fácilmente extendido con nuevos componentes creados con Java Servlets, clases Java, o C/C++.

3.5.2.5.4 Conexiones a bases de datos

Para los sitios ColdFusion la conexión a una base de datos Oracle puede realizarse de forma nativa, a través de JDBC o mediante ODBC.

3.5.2.5.5 ¿Por qué trabajar con ColdFusion?

Permite construir aplicaciones web rápidamente

- Mejora la productividad gracias al lenguaje de scripting del servidor basado en tags, aplicado de manera única en aplicaciones web.
- Acelera el desarrollo con un conjunto poderoso de herramientas poderosas de diseño, programación, depuración e implantación.
- Permite a los equipos de desarrollo colaborar de manera más efectiva compartiendo el mismo servidor y trabajando local o remotamente.

Ensambla soluciones poderosas fácilmente

- El servidor ColdFusion provee funcionalidades built-in como graficar, seguridad y búsqueda.
- Integración completa con la empresa, se conecta con todo el rango de sistemas backend, incluyendo bases de datos, servidores de mail, directorios, y aplicaciones empaquetadas. Se integra con tecnologías de

empresa y de Internet, incluyendo COM, CORBA, EJB, XML, C/C++ y Java.

- Posee inteligencia de negocios. Permite crear planillas y reportes tabulares de calidad profesional.
- Completa búsqueda de texto. Permite indexar fácilmente y buscar muchos tipos de contenido, incluyendo páginas web y documentos Microsoft Office 2000.

Entrega un alto desempeño y confiabilidad

- Arquitectura de alto desempeño. Asegura que las aplicaciones sean de implantación multiplataforma, entrega un avanzado thread pooling, caching de páginas built-in, consultas persistentes y pooling de conexiones a bases de datos.
- Administración fácil. Simplifica la implantación y la administración del servidor a través de una poderosa consola de administración basada en web, reportes robustos de servidor y herramientas de análisis, además de integración con los sistemas de administración de la empresa.
- Clustering de servidor. Provee balance de carga y recuperación automática para asegurar que las aplicaciones se mantengan consistentemente disponibles y se escala fácilmente para manejar tráfico creciente.

3.5.2.6 ColdFusion MX 7

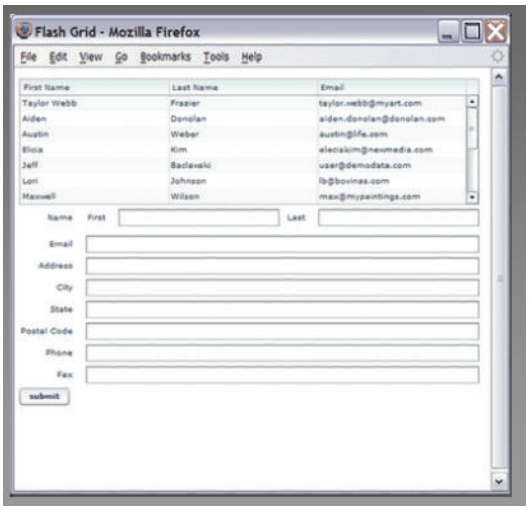
Hablar de Coldfusion 7 es hablar del futuro de las aplicaciones Web, ya que incorpora una gran inteligencia para el desarrollador, lo cual hace el trabajo mucho más fácil y rápido.

- Mejores aplicaciones Web.
- Nuevo administrador para el servidor.
- Nuevas aplicaciones de clases.

La nueva versión extiende Internet a teléfonos móviles y entrega un innovador soporte para formularios dinámicos, generación de informes y soluciones para impresión en Macromedia FlashPaper y PDF.

“Una vez más, Macromedia hace que los complicados problemas de desarrollo sean fáciles de resolver gracias a innovadoras capacidades disponibles en ColdFusion MX 7”, comenta David Mendels, Director General de Macromedia. “ColdFusion MX 7 provee a las organizaciones con la solución idónea para crear rápidamente aplicaciones de Internet que entregan un claro retorno de la inversión a la empresa y una experiencia de usuario mucho más convincente y atractiva.”

GRAFICO 14. FORMULARIO DISEÑADO CON COLDFUSION 7



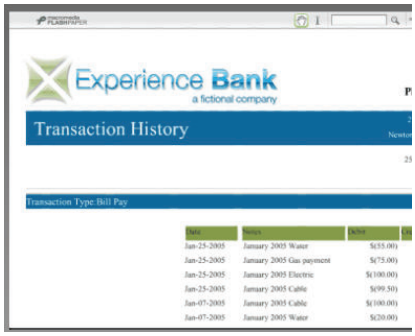
Fuente: www.adobe.com

Además de aplicaciones basadas en Internet, ColdFusion MX 7 permite a los desarrolladores crear una nueva clase de aplicaciones de Internet para el mundo móvil. Estas nuevas capacidades proporcionan una manera fácil y segura de extender aplicaciones a usuarios de telefonía móvil a través de mensajes de texto (SMS). También pueden crearse aplicaciones para que clientes de mensajería instantánea accedan a ellas. Estas innovadoras opciones de despliegue permiten

a las empresas atender a una amplia gama de necesidades empresariales propias del trabajo 'móvil' de hoy en día.

Con ColdFusion MX 7 los desarrolladores pueden crear aplicaciones de Internet más potentes en mucho menos tiempo de lo que nunca había sido posible. La tecnología de nuevos formularios dinámicos permite a los desarrolladores crear formularios en línea mucho mejores y de manera más fácil, sin tener que codificar toda la lógica de presentación. Los desarrolladores también pueden transformar contenido web dinámico en documentos con formato y que pueden ser impresos tanto a Macromedia FlashPaper como PDF.

GRAFICO 15. REPORTE DISEÑADO CON COLDFUSION 7



Fuente: www.adobe.com

Las nuevas características de creación de informes comerciales permiten a los desarrolladores generar informes bien estructurados y accesibles dentro de un contexto de aplicaciones web en la empresa, a la vez que dar acceso a los usuarios a información importante de la empresa en unos formatos fáciles de aprender y usar.

"El mundo digital, tal y como lo conocemos, está cambiando y está generando un crecimiento en el mercado del software de despliegue de aplicaciones de Internet en navegadores y más allá, hasta los límites de la Red", comenta Dennis Byron. Vicepresidente, business process automation and deployment software research, IDC. "La arquitectura abierta en software como ColdFusion es justo lo que se

necesita para hacer realidad la próxima generación de soluciones b2b (business to business).”

ColdFusion funciona tanto como servidor independiente de aplicaciones como sobre servidores de aplicaciones J2EE líderes, incluyendo Macromedia JRun, IBM WebSphere y BEA WebLogic, llevando el legado de la productividad de este software a la plataforma escalable y basada en estándares, Java. ColdFusion MX 7 facilita un Enterprise Manager que permite a los administradores crear fácil y rápidamente múltiples instancias de aplicaciones y, además, añade nuevos niveles de alto rendimiento y accesibilidad. La arquitectura abierta de ColdFusion MX 7 continúa haciendo que sea fácil desplegar aplicaciones en plataformas Windows, UNIX y Linux.

“ColdFusion MX 7 promete ser la versión preferida por los desarrolladores de entre todas las versiones de ColdFusion hasta la fecha”, comenta Joel Cox, IT project manager, The Goodyear Tire & Rubber Company. “La parte más tediosa del desarrollo de interfaces se ve aliviado, haciendo posible que los desarrolladores dediquen más tiempo a trabajar en la lógica subyacente de la Empresa”. [8]

3.5.3 ORACLE

La base de datos de Oracle tiene una capa lógica y otra física. La capa física consiste de archivos que residen en el disco y los componentes de la capa lógica son estructuras que mapean los datos hacia estos componentes físicos.

3.5.3.1 La Capa Física

Ya se dijo que consiste de archivos físicos que se encuentran en los discos. Estos pueden ser de tres tipos diferentes:

Uno o más *datafiles*

---

[8] Fuente: mastermagazine.info

Los *datafiles* almacenan toda la información ingresada en una base de datos. Se pueden tener sólo uno o cientos de ellos. Muchos objetos (tablas, índices) pueden compartir varios *datafiles*. El número máximo de *datafiles* que pueden ser configurados está limitado por el parámetro de sistema MAXDATAFILES.

Dos o más archivos *redo log* (de *deshacer*)

Los archivos del tipo *redo log* almacenan información que se utiliza para la recuperación de una base de datos en caso de falla. Estos archivos almacenan la historia de cambios efectuados sobre la base de datos y son particularmente útiles cuando se necesita corroborar si los cambios que la base de datos ya ha confirmado se han efectuado realmente en los *datafiles*.

Uno o más control files

Estos archivos contienen información que se utiliza cuando se levanta una instancia, tal como la información de dónde se encuentran ubicados los *datafiles* y los archivos *redo log*. Estos archivos de control deben encontrarse siempre protegidos.

3.5.3.2 La Capa Lógica

3.5.3.3 La capa lógica de una base de datos consta de los siguientes elementos:

Uno o más tablespaces

El esquema de la base de datos (*schema*), el cual consiste de objetos como tablas, clusters, índices, vistas, procedimientos almacenados, triggers, secuencias y otros.

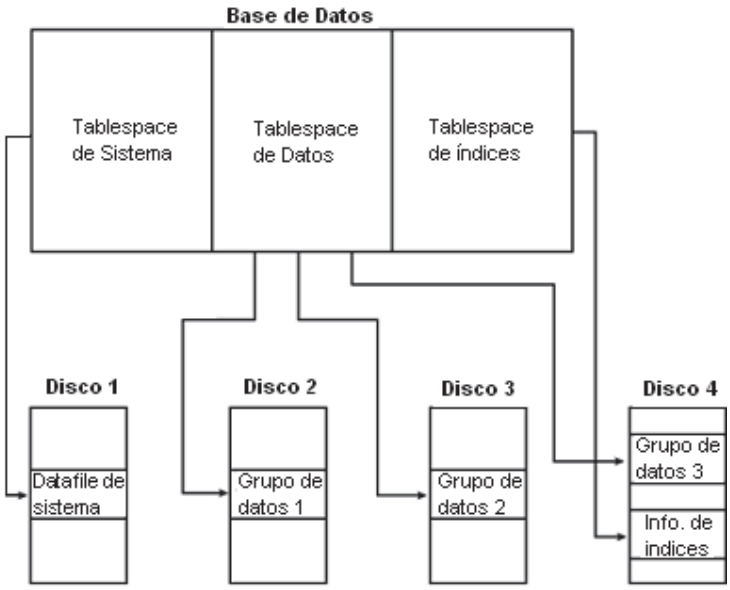
Los Tablespaces y los Datafiles

Como se mencionó, una base de datos se encuentra dividida en una o más piezas lógicas llamadas *tablespaces*, que son utilizados para separar la información en grupos y así simplificar la administración de los datos. Los *tablespaces* pueden ocupar uno o más *datafiles*. Si se decide que utilice varios

datafiles, el administrador del sistema puede gestionar que éstos queden localizados en discos diferentes, lo que aumentará el rendimiento del sistema, principalmente por la mejora en la distribución de la carga de entrada / salida.

En la figura siguiente se aprecia la diferencia entre estos tres conceptos. Una base de datos de ejemplo contiene tres *tablespaces* lógicos (parte superior de la figura) que utiliza para almacenar información del sistema, de los datos del usuario y de los índices de las tablas. Asimismo, existen los espacios físicos (*datafiles*) que guardan esta información en los diferentes discos disponibles y que se señalan en la parte inferior del dibujo.

GRÁFICO 16. RELACIÓN ENTRE LA BASE DE DATOS, LOS TABLESPACES Y LOS DATAFILES



Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

Segmentos, Extensiones y Bloques

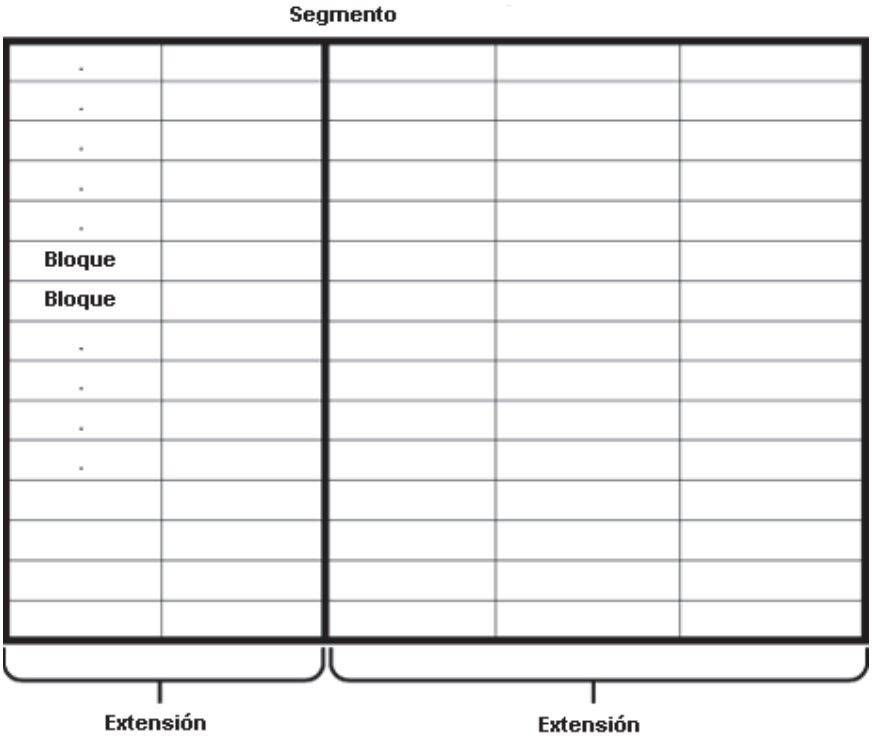
Dentro de los *tablespaces* y *datafiles*, el espacio utilizado para almacenar datos es controlado por el uso de ciertas estructuras; éstas son las siguientes:

Bloques: Un bloque es la unidad de almacenamiento más pequeña en una base de datos Oracle. Contiene una pequeña porción de información (*header*) referente al bloque en sí y el resto a los datos que guarda. Generalmente, un bloque de datos ocupará aprox. 2 KB de espacio físico en el disco (asignación típica).

Extensiones: Es un grupo de bloques de datos. Se establecen en un tamaño fijo y crecen a medida que van almacenando más datos. También se pueden redimensionar para aprovechar mejor el espacio de almacenamiento.

Segmentos: Es un grupo de extensiones utilizados para almacenar un tipo particular de datos. Existen 4 tipos de segmentos: datos, índices, rollback y temporales.

GRÁFICO 17. RELACIÓN ENTRE BLOQUES, EXTENSIONES Y SEGMENTOS



Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

El Esquema de la base de datos

Un esquema es una colección de objetos lógicos, utilizados para organizar de manera más comprensible la información y conocidos como objetos del esquema. Una breve descripción de los objetos que lo componen es la siguiente:

Tabla: Es la unidad lógica básica de almacenamiento. Contiene filas y columnas (como una matriz) y se identifica por un nombre. Las columnas también tienen un nombre y deben especificar un tipo de datos. Una tabla se guarda dentro de un *tablespace* (o varios, en el caso de las tablas particionadas).

Cluster: Un cluster es un grupo de tablas almacenadas en conjunto físicamente como una sola tabla que comparte una columna en común. Si a menudo se necesita recuperar datos de dos o más tablas basado en un valor de la columna que tienen en común, entonces es más eficiente organizarlas como un cluster, ya que la información podrá ser recuperada en una menor cantidad de operaciones de lectura realizadas sobre el disco.

*Índice:*Un índice es una estructura creada para ayudar a recuperar datos de una manera más rápida y eficiente. Un índice se crea sobre una o varias columnas de una misma tabla. De esta manera, cuando se solicita recuperar datos de ella mediante alguna condición de búsqueda (cláusula *where* de la sentencia), ésta se puede acelerar si se dispone de algún índice sobre las columnas-objetivo.

Vista: Una vista implementa una selección de varias columnas de una o diferentes tablas. Una vista no almacena datos; sólo los presenta en forma dinámica. Se utilizan para simplificar la visión del usuario sobre un conjunto de tablas, haciendo transparente para él la forma de obtención de los datos.

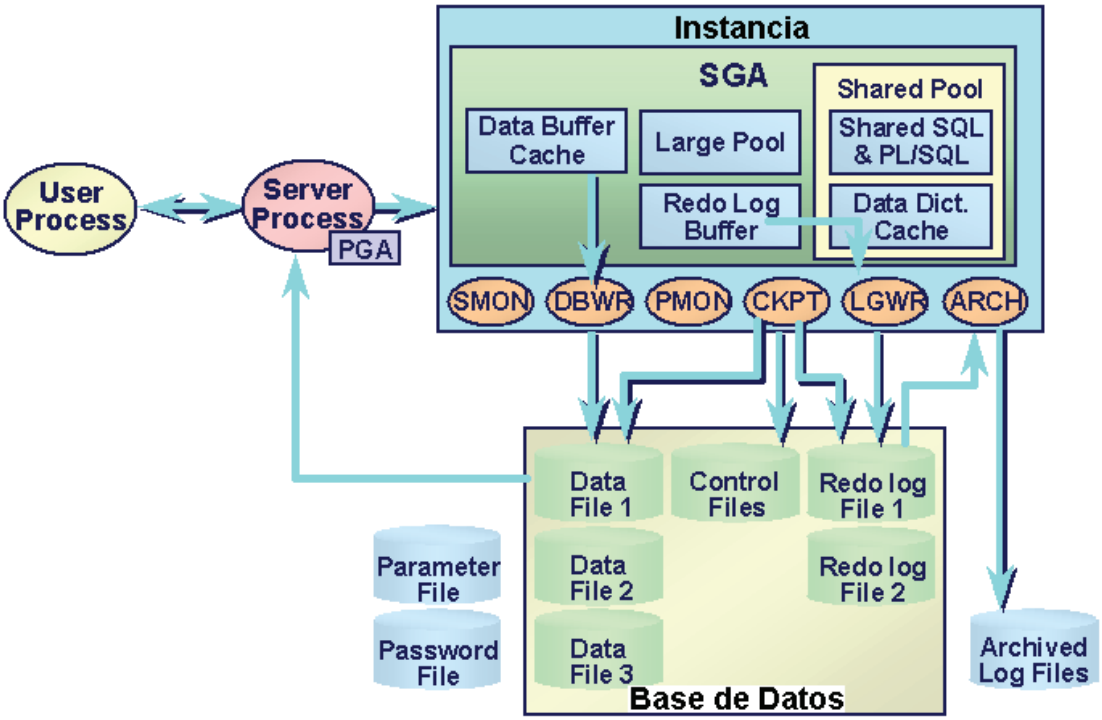
Proced. Almacenado: Son programas que permiten independizar el manejo de datos desde una aplicación y efectuarla directamente desde el motor de base de datos, disminuyendo así el tráfico de información a través de la red y mejorando el rendimiento de los procesos implementados mediante estos programas.

Trigger: Un *trigger* es un procedimiento que se ejecuta en forma inmediata cuando ocurre un evento especial. Estos eventos sólo pueden ser la inserción, actualización o eliminación de datos de una tabla.

Secuencias: El generador de secuencias de Oracle se utiliza para generar números únicos y utilizarlos, por ejemplo, como claves de tablas. La principal ventaja es que libera al programador de obtener números secuenciales que no se repitan con los que pueda generar otro usuario en un instante determinado.

3.5.3.4 Arquitectura de Oracle

GRÁFICO 18. VISTA GENERAL DE LA ARQUITECTURA DE ORACLE



Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

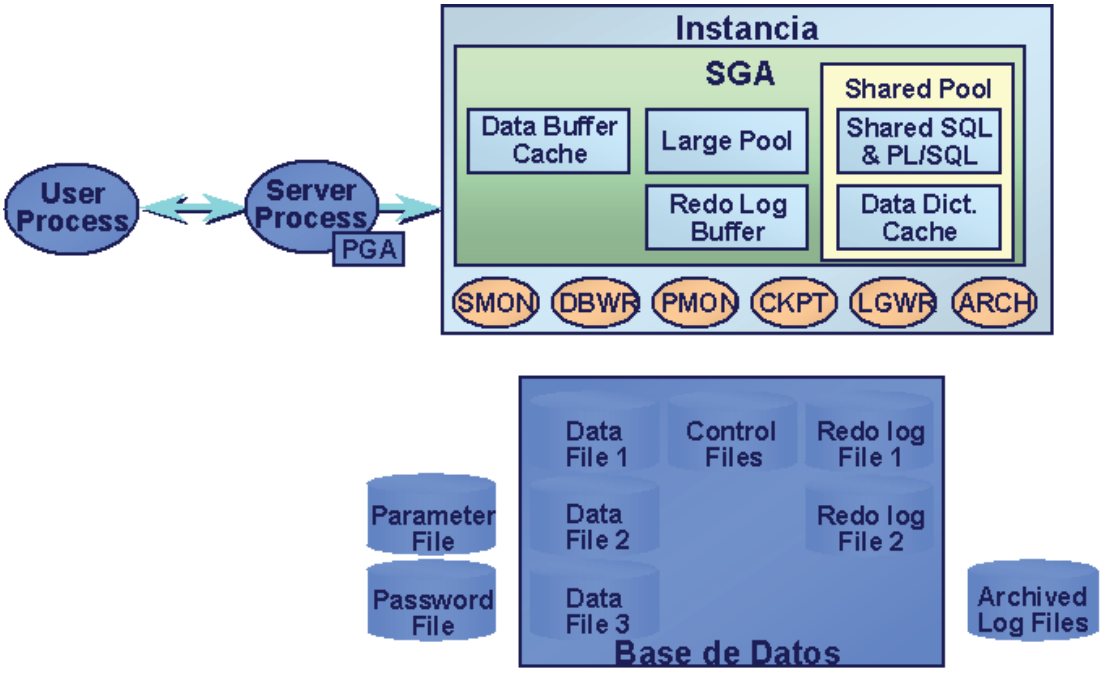
La Arquitectura general de Oracle consiste de varios procesos corriendo en la máquina donde reside la instancia, más los espacios de memoria dedicados a ejecutar procesos específicos o al almacenaje de información de cada proceso y la base de datos física propiamente tal, con sus archivos de control, de datos y de transacciones.

3.5.3.5 LA INSTANCIA ORACLE

Una instancia de Oracle está conformada por varios procesos y espacios de memoria compartida que son necesarios para acceder a la información contenida en la base de datos.

La instancia está conformada por procesos del usuario, procesos que se ejecutan en el background de Oracle y los espacios de memoria que comparten estos procesos.

GRÁFICO 19. ARQUITECTURA DE LA INSTANCIA DE ORACLE



Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

El Área Global del Sistema (SGA)

El SGA es un área de memoria compartida que se utiliza para almacenar información de control y de datos de la instancia. Se crea cuando la instancia es levantada y se borra cuando ésta se deja de usar (cuando se hace *shutdown*). La información que se almacena en esta área consiste de los siguientes elementos, cada uno de ellos con un tamaño fijo:

El buffer de caché (*database buffer cache*)

Almacena los bloques de datos utilizados recientemente (se hayan o no confirmado sus cambios en el disco). Al utilizarse este buffer se reducen las operaciones de entrada y salida y por esto se mejora el rendimiento.

El buffer de *redo log*: Guarda los cambios efectuados en la base de datos. Estos *buffers* escriben en el archivo físico de redo log tan rápido como se pueda sin perder eficiencia. Este último archivo se utiliza para recuperar la base de datos ante eventuales fallas del sistema.

El área *shared pool*: Esta sola área almacena estructuras de memoria compartida, tales como las áreas de código SQL compartido e información interna del diccionario. Una cantidad insuficiente de espacio asignado a esta área podría redundar en problemas de rendimiento. En resumen, contiene las áreas del *caché de biblioteca* y del caché del *diccionario de datos*.

El caché de biblioteca se utiliza para almacenar código SQL compartido. Aquí se manejan los árboles de *parsing* y el plan de ejecución de las *queries*. Si varias aplicaciones utilizan la misma sentencia SQL, esta área compartida garantiza el acceso por parte de cualquiera de ellas en cualquier instante.

El caché del diccionario de datos está conformado por un grupo de tablas y vistas que se identifican la base de datos. La información que se almacena aquí guarda relación con la estructura lógica y física de la base de datos. El diccionario de datos contiene información tal como los privilegios de los usuarios, restricciones de integridad definidas para algunas tablas, nombres y tipos de datos de todas las columnas y otra información acerca del espacio asignado y utilizado por los objetos de un esquema.

Procesos de la Instancia

Según lo que se advierte en la figura 5, los procesos que se implementan en una instancia de Oracle y su función principal son los siguientes:

DBWR (*database writer*): Es el responsable de la escritura en disco de toda la información almacenada en los *buffers* de bloques que no se han actualizado.

LGWR (*log writer*): Es el responsable de escribir información desde el *buffer de log* hacia el *archivo redo log*.

CKPT (*checkpoint*): Es el responsable de advertir al proceso DBWR de efectuar un proceso de actualización en el disco de los datos mantenidos en memoria, incluyendo los *datafiles* y *control files* (para registrar el *checkpoint*). Este proceso es opcional, si no está presente, es el proceso LGWR quien asume la responsabilidad de la tarea.

PMON (*process monitor*): Su misión es monitorizar los procesos del servidor y tomar acciones correctivas cuando alguno de ellos se interrumpe en forma abrupta, limpiando la caché y liberando los posibles recursos que pudieran estar asignados en ese momento. También es responsable por el restablecimiento de aquel proceso que se ha interrumpido bruscamente.

SMON (*system monitor*): Levanta una instancia cuando se le da la instrucción de partida (al comienzo del trabajo, encontrándose previamente en *shutdown*). Enseguida limpia los segmentos temporales y recupera las transacciones que pudieran haberse interrumpido debido a una falla del sistema. Además disminuye la fragmentación del sistema agrupando aquellas extensiones libres que existen dentro de la base de datos.

ARCH (*archiver*): La función de este proceso es la de *respaldar* la información almacenada en los archivos *redo log* cuando éstos se llenan. Este proceso está siempre activo cuando se ha establecido el modo ARCHIVELOG. Si el sistema no está operando en este modo se hace más difícil recuperar el sistema sin problemas luego de una falla general.

El Área Global de Programas (PGA)

Esta área de memoria contiene datos e información de control para los procesos que se ejecutan en el servidor de Oracle (relacionados con la base de datos, por supuesto). El tamaño y contenido de la PGA depende de las opciones del servidor que se hayan instalado.

Las Transacciones

El término transacción describe a una unidad lógica de trabajo que está compuesta de una o más sentencias SQL, que deben terminar con una instrucción *commit* o *rollback*. En ese instante, una nueva transacción dará comienzo y estará activa hasta que se ejecute alguno de esos dos comandos otra vez.

Cabe destacar que una transacción no se considera confirmada hasta que ésta se termina de escribir en el archivo de *redo log*.

3.5.3.6 CREACIÓN DE UNA BASE DE DATOS

Generalidades

En este capítulo no se discutirá en detalle como se debe crear una instancia o activar sus servicios porque se supone conocido el mecanismo de conectarse a una base de datos o instancia ya creada. Sin embargo, se repasarán los principales comandos que un DBA debiera reconocer para configurarla porque es un hecho que siempre utilizará alguna herramienta gráfica que le permita con mucha facilidad crear instancias y cree automáticamente los archivos de configuración. Un repaso no viene nada de mal.

En primer lugar debemos suponer que el software de Oracle ya se encuentra instalado o que estamos en ello. En la misma operación de instalación se nos preguntará si deseamos crear una instancia y, posteriormente, una base de datos dentro de ella.

Si no es el caso y debemos configurar cada una de ellas ya sea porque no existen, porque las que existen no nos satisfacen o están relacionadas con otros temas o porque no disponen de suficiente espacio, entonces la secuencia correcta es la siguiente:

GRÁFICO 21. SECUENCIA DE CREACIÓN DE INSTANCIAS Y BASE DE DATOS

Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

3.5.3.7 Trabajar con datos de tipo BLOB en ORACLE

Se puede trabajar con datos binarios en ORACLE. Los datos binarios nos van a permitir guardar en la base de datos archivos, imágenes, sonidos, etc.

Casi siempre es preferible guardar la ruta del archivo en la base de datos en lugar del propio archivo en modo binario, pero existen ciertas circunstancias en las que no nos queda otra solución.

Lo que vamos a hacer es cargar un fichero existente en el servidor en un campo BLOB de una tabla.

Lo primero que debemos hacer es crear un objeto directorio, esto es necesario ya que el fichero que queremos guardar se encuentra en el servidor (PL/SQL se ejecuta en el servidor), y debemos permitir explícitamente el acceso al directorio en cuestión al usuario que ejecutará el PL. El siguiente script SQL crea el directorio (¡ojo! El objeto ORACLE directorio, no el directorio físico del servidor que debe existir), asignándole el nombre lógico **imágenes**. Para poder crear el directorio debemos haber iniciado session como dba.

CUADRO 3. SENTENCIAS DE INICIO DE SESIÓN EN ORACLE

```
CONNECT sys/&password@ORACLEBD as sysdba;
CREATE OR REPLACE
DIRECTORY IMAGES AS 'C:\ORACLE\BLOB\IMAGES';
```

Lo siguiente que vamos a necesitar es una tabla con un campo BLOB, para almacenar la imagen. En este caso vamos a llamar a la tabla "archivos", y su estructura será la siguiente:

CUADRO 4. SENTENCIAS DE CREACIÓN DE TABLAS

```
CREATE TABLE ARCHIVOS
```

```
(CO_ARCHIVO     VARCHAR2(6)   not null,
 NOMBRE_ARCHIVO VARCHAR2(100) not null,
 BIN            BLOB          null,
 FX_ALTA        DATE          null,
 CONSTRAINT PK_ARCHIVOS PRIMARY KEY (CO_ARCHIVO)
 )
```

El siguiente bloque de PL nos va a permitir cargar una imagen, llamada *"imagen.gif"* en la tabla. Es importante tener claro que el archivo *"imagen.gif"* debe existir físicamente el directorio IMAGES (C:\ORACLE\BLOB\IMAGES) que hemos creado anteriormente.

CUADRO 5. SENTENCIAS DE DECLARACIÓN DE CAMPOS TIPO BLOB

```
DECLARE
 l_bfile  BFILE;
 l_blob   BLOB;
BEGIN
 INSERT INTO ARCHIVOS
  (CO_ARCHIVO, NOMBRE_ARCHIVO, BIN, FX_ALTA)
 VALUES
  ('000001','imagen.gif',EMPTY_BLOB(),SYSDATE)
 RETURN BIN INTO l_blob;



 l_bfile := BFILENAME('IMAGES', 'imagen.gif');
 DBMS_LOB.fileopen(l_bfile, Dbms_Lob.File_Readonly);
 DBMS_LOB.loadfromfile(l_blob,l_bfile,DBMS_LOB.getlength(l_bfile));
 DBMS_LOB.fileclose(l_bfile);
 COMMIT;

EXCEPTION WHEN OTHERS THEN
  ROLLBACK;
  RAISE;
END;
```

Hay tres aspectos a comentar de este código:

El uso de RETURN en la sentencia INSERT. Nos permite establecer una referencia al campo BIN insertado en la variable l_blob, de tipo BLOB.

La función **EMPTY_BLOB**. Nos permite insertar un valor nulo en un campo BLOB.

La función **BFILENAME**. Esta función devuelve un objeto BFILE que representa la ruta del fichero *"imagen.gif"* que queremos almacenar en la tabla.

El uso del paquete predefinido de ORACLE **DBMS_LOB**. Es el paquete que proporciona ORACLE para trabajar con tipos binarios. Utilizamos las siguientes funciones:

Fileopen: Abre el archivo definido por BFILE (l_bfile) en el modo indicado (en nuestro caso solo lectura Dbms_Lob.File_Readonly)

Loadfromfile: Lee un determinado número de bytes (en nuestro caso todos) del fichero definido por BFILE (l_bfile) en un objeto de tipo BLOB (l_blob).

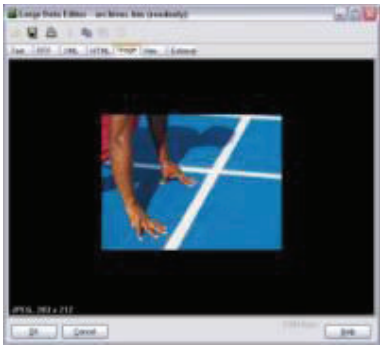
Getlength: Devuelve el tamaño del archivo en bytes.

Fileclose: Cierra el archivo

Llama la atención de este código que a pesar de haber insertado el campo BIN vacío con la función EMPTY_BLOB, finalmente queda cargado sin ejecutar ninguna sentencia UPDATE. Esto ocurre porque estamos utilizando RETURN en la sentencia INSERT y guardando una referencia al campo BIN que posteriormente asignamos al leer el archivo con DBMS_LOB.loadfromfile.

El resultado de todo esto es que hemos conseguido almacenar la imagen en la base de datos.

GRÁFICO 21. EJEMPLO DE CAMPO BLOB EN ORACLE



Fuente: Guía proporcionada por Oracle "*Manual Curso Introductorio a la Administración de Oracle".*

3.5.4 JAVASCRIPT

Se trata de un lenguaje de programación del lado del cliente, utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web; pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Es un lenguaje muy utilizado, debido a su compatibilidad con la mayoría de los navegadores.

Entre las acciones típicas que se realizar en Javascript tenemos dos vertientes. Por un lado los **efectos especiales** sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear **páginas interactivas** con programas como calculadoras, agendas, o tablas de cálculo.

(Las Normas del código en Javascript son:

Todo el código (sentencias) esta dentro de funciones.

Las funciones se desarrollan entre las etiquetas <script> y</script>.

Las etiquetas "<script>" deben colocarse entre las etiquetas <head> y </head>.

Las etiquetas "<title>" no pueden estar colocadas entre las de"<script>".

La llamada a la función se hace a través de un evento de un elemento del documento.)

3.5.5 ACTIONSCRIPT

ActionScript es un lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones web animadas realizadas en el entorno Macromedia Flash, la tecnología de Macromedia para añadir dinamismo al panorama web. Fue lanzado con la versión 4 de Flash, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamicidad y versatilidad muy altos en la versión 8 de Flash.

ActionScript es un lenguaje de script, esto es, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí que ActionScript se parezca tanto a Javascript.

La versión más extendida actualmente es ActionScript 2.0, que incluye clases y es utilizada en la última versión de Macromedia Flash y en anteriores versiones de Flex. Recientemente se ha lanzado la beta pública de Flex 2, que incluye el nuevo ActionScript 3, con mejoras en el rendimiento y nuevas inclusiones como el uso de expresiones regulares y nuevas formas de empaquetar las clases. Incluye, además, Flash Player 8.5, que mejora notablemente el rendimiento y disminuye el uso de recursos en las aplicaciones Flash.

Con estas características ColdFusion ha incorporado ActionScript en formularios tipo Flash, los cuales se corren en tags de contenido de script, haciendo al programa mucho más visual y poder correr las validaciones en el momento de ejecución sin esperar una recarga para ejecutar funciones.

3.6 OTRAS HERRAMIENTAS

Herramientas para el diseño de páginas Web.

El usuario debe tener un conocimiento avanzado en el uso de computadoras; el material que se necesita es el siguiente:

-Computadora PC o Macintosh.

-Para efectos nuestros, todo se hará tomando en cuenta que se trabaja en una Computadora PC con sistema operativo WINDOWS XP, 2000.

-Cualquier programa editor de páginas WEB, en este caso Dream Weaver.

-Dirección en el servidor, en donde se guardará la página creada.

Debemos asegurarnos que el proveedor de servicio de Internet (ISP) asigne el espacio necesario para el Sitio Web y se levante el servicio de ColdFusion.

-*Cualquier programa editor de archivos gráficos.* Entre ellos pueden estar COREL DRAW, FREE HAND, PAGE MAKER, etc.

-Cualquier programa editor de TEXTOS en formato ASCII. Puede ser el programa Wordpad o Block de Notas.

CAPITULO 4

CONLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El uso de la metodología OOHDM como técnica de diseño, a permitido el control e implementación de las fases. La implementación de las fases permite obtener un mejor resultado en el proceso de desarrollo simplificándolo, haciendo su mantenimiento mucho más fácil y reduciendo el tiempo de desarrollo de todas las aplicaciones web; teniendo como objetivo principal la reusabilidad de diseño.
- ColdFusion, como herramienta para el desarrollo web es un punto importante a destacar; ya que fusiona aspectos muy importantes en las aplicaciones Web, como son: la seguridad, el ahorro de líneas de código, la facilidad de aprendizaje y desarrollo, la tendencia a la interconexión de dispositivos móviles y su reusabilidad de código dentro de los componentes.
- Como parte de la seguridad en ambientes Web, es muy importante el aspecto del Servidor Web; como no se podía quedar atrás ColdFusion permite el uso del servidor Apache en el Sistema Operativo Linux, lo cual hace de esta aplicación Web muy segura y su implantación en software Open Source.
- Al termino de la aplicación Web, dió como resultado un aspecto muy importante, que es el fomentar la investigación e implantación de nuevos lenguajes para el desarrollo Web, que permiten mejorar y fortalecer los conocimientos en Diseño, Programación, Seguridad y conexiones a nuevas tecnologías; permitiendo ampliar nuestra visión profesional de la carrera.

4.2 RECOMENDACIONES

- Se recomienda que el módulo de compañías y programa de buques se migre, toda la interfaz y funcionalidad, a la nueva versión de ColdFusion 7. La cual mejoraría el manejo, administración y lo más importante que facilitaría el mantenimiento de la aplicación Web.
- En el módulo de Programación de Embarques, tanto para las Importaciones como para las Exportaciones, se recomienda construir una página para las instrucciones que se dan a los capitanes de los buques en cada regional, como son órdenes de alistamiento, zarpe y otras.
- En el caso de los programas de embarques se recomienda el desarrollo del módulo de ejecución de los mismos para un mejor control de lo que pasa en los terminales.
- Se recomienda que PETROECUADOR, adquiera nuevo equipo que vaya con los requerimientos recomendados de la aplicación, evitando así problemas en los tiempos de espera de la información. También que se desarrollen módulos de integración al sistema como pueden ser Cartas de Crédito, Contratos, Catálogos entre otros.

REFERENCIAS BIBLIOGRAFÍCAS

Metodología:

- Liza Ávila C., Modelando con UML Principios y Aplicaciones, (2001), Primera Edición.
- Pressman Roger, (2002), Ingeniería de Software, Quinta Edición.
- Stevens P. y Pooley R., (2002), Utilización de UML en Ingeniería del Software con Objetos y Componentes, Primera Edición, Madrid, PEARSON EDUACIÓN S.A.
- Investigación Web, http://gplsi.dlsi.ua.es/iwad/investigacionWeb.html
- Metodología OOHDM, http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r22_art5_c.pdf
- OMT, http://pisuerga.inf.ubu.es/icruzado/tfc/OMT_AnaUML_Coinfiti.pdf#search=%22descripcion%20Diagrama%20de%20clases%20%22
- UML, http://www.ingenierosoftware.com/analisisydiseno/uml.php

Herramientas

- ActionScript, http://www.ati.es/novatica/2004/168/168-4.pdf
- Arquitectura campos BLOBs, http://www.devjoker.com/asp/ver_contenidos.aspx?co_contenido=83.
- Ben Forta y Nae Weiss(2003), Macromedia ColdFusion MX Web Application Construction Kit, Quinta Edición, California, PRESS.
- Brooks R. y Bilson (2003), Programming Coldfusion MX, Segunda Edición, California, O' Reilly & Associates.
- ColdFusion, http://cursosgratis.emagister.com/frame.cfm?id_user=57845390203200621496848565351506&id_centro=611740900330666667485065496945 52&id_curso=328356631659347217751924422049 56&url_frame=http://www.tejedoresdelweb.com/307/article-5812.html

- http://www.uca.edu.sv/investigacion/bdweb/reportes/coldfusion.html#_Toc392273171
- Conexión ColdFusion Oracle, http://www.macromedia.com/es/support/dreamweaver/ts/documents/dwmx_dynamic_faq.htm
- Definición e historia HTML ,http://html.conclase.net/w3c/html401-es/intro/intro.html#h-2.2
- Estructura HTML, http://www.ilustados.com/publicaciones/EpyupElVpEQUDHjXOR.php
- JavaScript, http://www.ilustrados.com/publicaciones/EpyupElVpEQUDHjXOR.php
- Ladd Erick (2002), Macromedia ColdFusion MX Development, Indiana, QUE.
- Oracle documento, http://www.zonaoracle.com/manuales-tutoriales-oracle/index.asp?id=1.

ANEXOS

ANEXOS NO. 1 MANUAL TÉCNICO

ANEXOS NO. 2 MANUAL DEL USUARIO

ANEXOS NO. 3 DIAGRAMA DE LA BASE DE DATOS