

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7

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

LUIS JEFFERSON GUALOTO PILLAJO

jefferson_siul2000@hotmail.com

DIRECTOR: ING. MARITZOL TENEMAZA MSc.

rmtenemaza@gmail.com

Quito, Agosto 2012

DECLARACIÓN

Yo, Luis Jefferson Gualoto Pillajo, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he 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 normatividad institucional vigente.

Luis Jefferson Gualoto Pillajo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Luis Jefferson Gualoto Pillajo, bajo mi supervisión

Ing. Maritzol Tenemaza Msc.
DIRECTORA DE PROYECTO

AGRADECIMIENTOS

Agradezco de manera muy especial a mis padres Luis y Nelly, a mis hermanas Viviana, Mishel, a mi primo Alex y a la Srta. Anita que constantemente me brindaron su apoyo, sabios consejos y fuerza para continuar.

También agradezco a mi tutora de tesis la Master Maritzol Tenemaza, por su comprensión, ayuda y ser mi guía en la realización de este proyecto.

Jefferson Gualoto

DEDICATORIA

Dedico este proyecto a mi Dios, a mis padres Luis Gualoto y Nelly Pillajo, a mis hermanas Viviana y Mishel Gualoto por estar constantemente motivándome para ser mejor persona y profesional.

Jefferson

CONTENIDO

DECLARACIÓN.....	I
CERTIFICACIÓN.....	II
AGRADECIMIENTOS.....	III
DEDICATORIA.....	IV
CONTENIDO.....	V
RESUMEN.....	VIII

CAPITULO 1	1
PLANIFICACIÓN O ANTECEDENTES DEL PROYECTO	1
1.1 Introducción – Ámbito	1
1.2 Planteamiento del Problema	1
1.3 Formulación del Problema	2
1.4 Sistematización del Problema	2
1.5 Objetivos de la Investigación	3
1.5.1 Objetivo General	3
1.5.2 Objetivos Específicos	3
1.6 Justificación de la Propuesta	4
1.6.1 Práctica	4
1.6.2 Técnica	4
1.7 Alcance	5
1.8 Aspectos Marco Metodológico	6
CAPITULO II	8
ASPECTOS TEÓRICOS	8
2.1 INGENIERÍA WEB	8
2.1.1 Concepto de Ingeniería Web	8
2.1.2 Proceso de Ingeniería Web	9
2.1.3 El modelo de proceso de la Ingeniería de la Usabilidad	10
2.1.3.1 El análisis de requisitos	10
2.1.3.2 Fase de diseño, evaluación y desarrollo	10
2.1.3.3 La fase de instalación	11
2.1.4 Diseño web centrado en el usuario	11
2.1.4.1 Planificación	12
2.1.4.2 Diseño	13
2.1.4.3 Modelado del Usuario	13
2.1.4.4 Diseño Conceptual	13
2.1.4.5 Prototipado	15
2.1.4.6 Evaluación	16
2.1.4.6.1 Método por Inspección: Evaluación Heurística	16

2.1.4.6.2	<i>Método de test con usuarios</i>	16
2.1.4.6.3	<i>Implementación y lanzamiento</i>	16
2.1.4.6.4	<i>Mantenimiento y Seguimiento.</i>	17
2.2	AQUITECTURA DE DESARROLLO WEB	18
2.2.1	El Navegador Web, Browser	19
2.2.2	El Servidor Web	19
2.2.3	Aplicaciones Multinivel	20
2.3	HERRAMIENTAS DE DESARROLLO	22
2.3.1	Linux	22
2.3.1.1	Distribuciones Linux	24
2.3.1.1.1	<i>Linux RedHat 8.0</i>	25
2.3.2	Apache	30
2.3.2.1	Arquitectura Apache	31
2.3.3	Coldfusion	32
2.3.3.1	Coldfusion y sus Características	32
2.3.3.2	Componentes Coldfusion	33
2.3.3.2.1	<i>Tecnología del servidor</i>	33
2.3.3.2.2	<i>Herramientas de desarrollo</i>	33
2.3.3.2.3	<i>Ambiente de programación</i>	34
2.3.4	ActionScript	37
2.3.4.1	Características generales del ActionScript	37
2.3.5	JavaScript	38
2.3.5.1	Que no es Javascript	39
2.3.5.2	Diferencias entre Javascript y Java	40
2.3.6	Oracle	41
2.3.6.1	Historia	42
2.3.6.2	Esquema de la Base de Datos	42
2.3.6.3	Estructuras de Proceso	44
2.3.6.3.1	<i>System Monitor, SMON</i>	45
2.3.6.3.2	<i>Process Monitor, PMON</i>	45
2.3.6.3.3	<i>Database Writer, DBWR</i>	45
2.3.6.3.4	<i>Log Writer, LGWR</i>	46
2.3.6.3.5	<i>Checkpoint, CKPT</i>	47
2.3.6.3.6	<i>Archiver, ARCH</i>	48
2.3.6.3.7	<i>Recovered, RECO</i>	48
2.3.6.3.8	<i>Lock, LCK</i>	49
2.4	PARADIGMA ESPIRAL ORIENTADO A LA WEB	49
2.4.1	Etapas	49
2.4.1.1	Planificación	49
2.4.1.2	Comunicación con el Cliente o Formulación	50
2.4.1.3	Análisis De Riesgos	50
2.4.1.4	Ingeniería	50
2.4.1.5	Generación de Páginas	50
2.4.1.6	Evaluación Cliente	51
2.5	MODELO OOHDM (OBJECT-ORIENTED HYPERMEDIA DESIGN MODEL) O MÉTODO DE DISEÑO DE HIPERMEDIA ORIENTADO A OBJETOS	51

2.5.1	Diseño Conceptual	52
2.5.2	Diseño Navegacional.....	53
2.5.3	Diseño de Interfaz Abstracta.....	54
2.5.4	Implementación	54
2.6	UML.....	55
2.6.1	Diagramas de Casos de Uso	57
2.6.1.1	Actor	57
2.6.1.2	Casos de Uso.....	57
2.6.1.3	Relaciones.....	58
2.6.1.3.1	Asociación	58
2.6.1.3.2	Dependencia o Instanciación	58
2.6.1.3.3	Generalización	58
2.6.2	Diagrama de Clases	59
2.6.2.1	Clase.....	59
2.6.2.1.1	Relaciones entre clases.....	59
2.6.2.1.2	(Herencia/Generalización).....	60
2.6.2.1.3	Agregación	60
2.6.2.1.4	Asociación	60
2.6.2.1.5	Dependencia o Instancia	61
2.6.3	Diagrama de Objetos.....	61
2.6.4	Diagrama de Secuencia	62
2.6.4.1	Objeto/Actor	62
2.6.4.2	Mensaje a otro Objeto.....	63
2.6.4.3	Mensaje al mismo Objeto	63
2.6.5	Diagrama de Colaboración.....	63
2.6.5.1	Objeto	63
2.6.5.2	Enlaces	63
2.6.5.3	Flujo de Mensajes	64
2.6.5.4	Marcadores de Creación y Destrucción de Objetos	64
2.6.5.5	Objeto Compuesto	64
2.6.6	Diagrama de Actividades	64
2.6.6.1	Estados de Actividad y Estados de Acción	65
2.6.6.2	Transiciones	65
2.6.6.3	Bifurcaciones.....	66
2.6.6.4	División y Unión.....	66
2.6.6.5	Calles.....	67
	ASPECTOS METODOLOGICOS	68
3.1	CONCLUSIONES	68
3.2	RECOMENDACIONES	68
3.3	BIBLIOGRAFÍA	69
3.3.1	Referencias Web	69
3.3.1.1	Paradigmas.....	69
3.3.1.2	Metodología.....	70
3.3.1.3	UML.....	70
3.3.1.4	Java	71

RESUMEN

El Sistema de Gestión de Contratos para Petroecuador Desarrollado en la Herramienta Web Coldfusion MX7; es un sistema orientado a la web que permite la administración de forma eficiente y segura de la información del proceso previo de los contratos, avances físicos y económicos luego del proceso de contratación (contratos, controles periódicos, consultas, informes y reportes); de la Empresa Pública Petroecuador.

El presente trabajo está estructurado de la siguiente forma:

En el capítulo I, se explican los principales motivos que llevaron a desarrollar este trabajo, haciendo una descripción del problema así como de la solución al mismo, se da a conocer los objetivos generales y específicos que se van a cumplir, y el alcance al que se pretende llegar con el desarrollo del proyecto.

En el capítulo II, se realiza una descripción del paradigma que se utilizó a lo largo del desarrollo del proyecto, la metodología y sus correspondientes métodos y herramientas, y de igual manera se hace alusión a ciertos conceptos básicos de temas utilizados en la implementación del proyecto.

En el capítulo III, finalmente se redacta ciertas conclusiones y recomendaciones que pueden servir como lineamientos para futuras mejoras del Sistema de Software, además de la bibliografía utilizada para el desarrollo de este proyecto.

CAPITULO I

PLANIFICACIÓN O ANTECEDENTES DEL PROYECTO

1.1 Introducción – Ámbito

La Gerencia de Desarrollo Organizacional de la Empresa de Hidrocarburos del Ecuador Petroecuador es la unidad encargada de la gestión y verificación del cumplimiento de la contratación de servicios, tanto Técnicos como Legales y Convenios con instituciones públicas o privadas, de la empresa. Facilita información y consultas, al personal de la empresa Petroecuador o entidades externas, de procesos realizados previos a la contratación de servicios que esta requiere. La Gerencia realiza un control periódico del avance físico y económico de todos los contratos efectuados por las gerencias; lleva a cabo el seguimiento y la verificación del cumplimiento de las condiciones establecidas en cada contrato y el control de plazos para entrega de obras y servicios, además pone multas por retrasos ocurridos en la entrega de obras o servicios.

Además la unidad es la encargada de asignar fiscalizadores para cada contrato, los cuales, desarrollan reportes periódicos de avances físico y económico de las obras de acuerdo a los datos tomados de cada uno de los contratos.

1.2 Planteamiento del Problema

Actualmente la Gerencia de Desarrollo Organizacional utiliza la herramienta Microsoft Office Excel para el control, seguimiento y gestión de los contratos, realizados por todas las Gerencias pertenecientes a Petroecuador. Las Gerencias realizan toda la gestión de los Contratos, tales como: registro del contrato, reportes, consultas, modificaciones o actualizaciones y eliminación de contratos,

de forma dificultosa porque, al momento el Sistema actual genera los siguientes problemas:

- El sistema actual no cuenta con una base de datos confiable.
- Pérdida de tiempo, tanto para el usuario como para la Empresa
- Demoras en el acceso y búsquedas de información.
- La información almacenada no puede ser manipulada de acuerdo a las necesidades de los usuarios y autoridades de la Empresa.
- Equivocaciones al realizar cálculos.
- El sistema actual no permite generar reportes confiables.
- El acceso y transferencia de datos no es confiable.
- Al no contar con una base de datos, la información de los contratos no se encuentra debidamente organizada.

Petroecuador ha decidido desarrollar todas sus aplicaciones en plataforma Web, considerando que el constante crecimiento de aplicaciones y sistemas en ambientes Web ha proporcionado novedosas ventajas y facilidades en el manejo de información a través de una red o intranet sin importar la plataforma o sistema operativo en que esté la aplicación.

Por lo expuesto, se ha visto la necesidad de desarrollar el Sistema Gestión de Contratos en plataforma Web.

1.3 Formulación del Problema

¿Cómo mejorar la gestión y control de los Contratos realizados por PETROECUADOR de modo que su eficiencia, confiabilidad y seguridad de los datos puedan manipularse de acuerdo a los requerimientos de la Gerencia de Desarrollo Organizacional?

1.4 Sistematización del Problema

- ¿Cómo se va a mantener la funcionalidad que presta el software actual?

- ¿Cómo administrar la información de contratos de acuerdo a Gerencias, y sus controles periódicos?
- ¿Cómo administrar la información de consultas e informes de contratos?
- ¿Cómo se validará la autenticidad de los usuarios?
- ¿Cómo se asegurará la discrecionalidad en el acceso a la información?
- ¿Cómo se asegurará que la información del Seguimiento de Contratos de Petroecuador pueda ser consultada por personas ajenas?

1.5 Objetivos de la Investigación

1.5.1 Objetivo General

Implementar una aplicación Web para la gestión y control del cumplimiento de avance económico y el porcentaje físico establecidos en los Contratos de Petroecuador con los Oferentes Nacionales e Internacionales registrados para prestar sus servicios a la empresa.

1.5.2 Objetivos Específicos

- Desarrollar completamente el sistema Gestión de Contratos en la herramienta Coldfusion MX para el desarrollo de aplicaciones Web tomando en cuenta las políticas manejadas por las Gerencias.
- Construir un módulo para administrar la información de contratos creados y asignados a un responsable y fiscalizador.
- Construir un módulo para administrar la información de consultas e informes de contratos.
- Establecer claves personales, a cada usuario, para controlar el acceso, a la Aplicación.
- Construir un módulo para validación de usuario y contraseña, que permita desplegar la información de acuerdo al perfil del usuario y la gerencia a la que pertenece, para mantener la discrecionalidad de la información.

- Construir el aplicativo en la intranet y establecer políticas de seguridad y transparencia en el manejo de la información de acuerdo a la funcionalidad de los usuarios, como jefe de gerencia, funcionario o secretaria.

1.6 Justificación de la Propuesta

1.6.1 Práctica

Con la implementación de este sistema Petroecuador podrá contar con el manejo de la información de sus Contratos, de modo que sus datos puedan ser administrados de forma segura, para así garantizar la transparencia en el manejo de la información. Además facilitará el acceso al usuario mediante claves personales independiente del lugar donde se encuentre, ya que la principal fortaleza de una aplicación Web está en la accesibilidad universal, mediante un navegador Web y que no requieren de alguna instalación previa, y eso facilita las actualizaciones del mismo que se realiza exclusivamente en el servidor.

1.6.2 Técnica

Para el análisis, diseño y construcción de este proyecto se ha tomado en cuenta el paradigma espiral incremental para tener un continuo mejoramiento del software hasta un límite determinado, permitiendo llevar un control del mejoramiento del software y una prevención a riesgos en cada mejoramiento; la metodología OMT (Técnica de Modelamiento de Objetos) y los diagramas de UML (Lenguaje Unificado de Modelamiento) que nos indican los pasos a seguir y como representarlos visualmente para desarrollar de forma eficiente un sistema de software.

Es así que se ha tomado en cuenta la especificación de requerimientos a través de reuniones de trabajo con el usuario, para así tener muy claros los resultados que el usuario quiere del sistema de software, es decir, una idea en común y plasmar estos requerimientos a través del modelamiento de diagramas para que

puedan ser interpretados por todos los participantes en el análisis, diseño y construcción del sistema de software.

Se definió la funcionalidad requerida del sistema de software a través de diagramas de casos de uso, con la especificación de cada caso de uso se definió los procesos necesarios y se los representa en los diagramas de secuencia y actividades; para determinar el comportamiento interno (flujo de tareas y modelamiento de operaciones) de cada proceso, los objetos participantes, las interacciones y los mensajes que se intercambian ordenados según su secuencia en el tiempo de cada objeto; el almacenamiento de los datos (tipos, longitud, valores nulos, valores por defecto, etc.), la distribución y las relaciones que se generan entre ellos y así definir las estructuras de datos (Base de Datos) para el almacenamiento de la información a través de los diagramas de objetos y de clases.

1.7 Alcance

El presente proyecto **“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARRROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7”** tiene como finalidad desarrollar una aplicación WEB, para apoyar la administración de forma eficiente y segura de la información de gestión de contratos (usuarios, ingreso, controles periódicos, consultas y reportes), modificaciones y actualizaciones a los contratos de la empresa Petroecuador.

El proyecto parte de un análisis de la situación actual del proceso que se lleva para el registro de contratos, generación de consultas y reportes, el cual propone un nuevo procedimiento en base al cual se diseñará y construirá una aplicación WEB, la misma que permitirá automatizar el flujo del procedimiento, a fin de apoyar en la administración y optimizar los recursos materiales, humanos, el

tiempo y la manipulación de la información, y de esta manera mejorar la calidad del servicio.

El proceso objeto de estudio cubre las siguientes partes:

- Administrar información de contratos, su registro y sus controles periódicos.
- Administrar la información de personal responsable de los contratos y asignados a un fiscalizador, con sus respectivos avances económico y físico.
- Administrar la información de consultas y reportes por usuario.
- Generar reportes y estadísticas de toda la información registrada.
- Controlar el acceso al sistema y la manipulación de la información de contratos (usuarios, ingreso, controles periódicos, consultas y reportes), tipo de contrato y empresas registradas para prestar sus servicios a la Petroecuador.

1.8 Aspectos Marco Metodológico

PARADIGMA Espiral Incremental	METODOLOGÍA OOHDM	DIAGRAMAS UML	
Análisis	Diseño Conceptual	Descripción de Actores Diagrama de Casos de Uso Descripción de Casos de Uso Diagrama de Clases Diccionario de Clases Diagrama de Objetos	
Diseño	Modelo Dinámico	Diagrama de Interacción	Diagrama de Secuencia Diagrama de Colaboración
	Modelo Funcional	Diagrama de Actividades	
	Diseño Navegacional	Diagrama Navegacional	
Construcción	Herramientas de Software	Plataforma Web Macromedia Dreamweaver Oracle 10g ColdFusion MX7 Colfusion Report Builder Power Designer 12.5	

PARADIGMA Espiral Incremental	METODOLOGÍA OOHDM	DIAGRAMAS UML
Pruebas	Prueba Funcional	Descripción de la Prueba Procedimiento de la Prueba
Mantenimiento	La documentación del desarrollo y seguimiento de los modelos facilita el posterior mantenimiento.	

CAPITULO II

ASPECTOS TEÓRICOS

2.1 INGENIERÍA WEB

2.1.1 Concepto de Ingeniería Web

La Ingeniería Web es el proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad. Está relacionada con el establecimiento y utilización de principios científicos, de ingeniería y gestión, y con enfoques sistemáticos y disciplinados del éxito y desarrollo, empleo y mantenimiento de sistemas y aplicaciones basados en el Web de alta calidad.

La ingeniería Web no es un clone de la Ingeniería del Software aunque ambos involucren programación y desarrollo de software.

Cabe destacar que la ingeniería de la web hace una diferencia entre un sitio web y un aplicativo, ya que la ingeniería de la web no se dedica a la construcción de sitios web si no a la construcción de aplicativos web, la principal característica que los distingue (aplicativos de sitios web) es que los sitios web son sitios en la web en donde se publica contenido generalmente estático o un muy bajo nivel de interactividad con el usuario, mientras que los aplicativos son lugares con alto contenido de interactividad y funcionalidades que bien podrían ser de un software convencional, el aplicativo web más sencillo sería uno que contenga formularios y subiendo de nivel encontramos los que realizan conexión con bases de datos remotas, y administradores de contenidos entre otras.

2.1.2 Proceso de Ingeniería Web

Es un modelo de proceso de software evolutivo que proporciona el potencial para el desarrollo rápido de versiones incrementales del software, permite al desarrollador y al cliente comprender y reaccionar mejor ante riesgos en cada uno de los niveles evolutivos del software. Características como inmediatez y evolución y crecimiento continuos, nos llevan a un proceso incremental y evolutivo, que permite que el usuario se involucre, facilitando el desarrollo de productos que se ajustan mucho lo que éste busca y necesita.

Existen algunas actividades importantes y que forman parte de este proceso, estas son: formulación, planificación análisis, modelización, generación de páginas, test y evaluación del cliente.

Formulación: la Formulación identifica objetivos y establece el alcance de la primera entrega.

Planificación: la Planificación genera a estimación del coste general del proyecto, la evaluación de riesgos y el calendario del desarrollo y fechas de entrega.

Análisis: el Análisis especifica los requerimientos e identifica el contenido.

Modelización: 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.

Generación de Páginas: en la generación de páginas se integra contenido, arquitectura, navegación e interfaz para crear estática o dinámicamente el aspecto más visible de las aplicaciones, las páginas.

Test: el Test busca errores a todos los niveles: contenido, funcional, navegacional, rendimiento, etc.

Evaluación del Cliente: aquí se realizan todas las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software generadas durante todo el proceso.

2.1.3 El modelo de proceso de la Ingeniería de la Usabilidad

En este modelo hay tres grandes fases: el análisis de los requisitos; el diseño, evaluación y desarrollo; y por último, la de instalación.

2.1.3.1 El análisis de requisitos

Se inicia estableciendo el perfil de los usuarios del sistema y llevando a cabo el análisis conceptual de las tareas, la definición de las restricciones y necesidades de la plataforma de uso y de los principios de diseño a aplicar.

Todo ello da lugar a una serie de objetivos de usabilidad que el sistema debe cumplir y que, junto a las guías de estilo, conforman el conjunto estable de requisitos a considerar en el proceso de desarrollo.

2.1.3.2 Fase de diseño, evaluación y desarrollo

Se definen tres niveles de desarrollo que se corresponden con distintos niveles de abstracción, cada uno de los cuales hace uso de maquetas y prototipos como artefactos destinados a ser evaluados.

En el primer nivel, se realiza el diseño conceptual y se construyen maquetas del sistema y que se evalúan para El segundo nivel se centra en el diseño de las ventanas y en la construcción de los prototipos que van a permitir comprobar que se han cubierto todos los aspectos identificados en la fase anterior que representan la interacción con él, (garantizar de manera iterativa si los objetivos de

usabilidad se cumplen). Por último, la interfaz se va refinando de manera iterativa hasta conseguir un sistema que incorpore todas las funcionalidades requeridas.

2.1.3.3 La fase de instalación

La fase de instalación del mismo en el entorno de exploración en el que el usuario real interactuará con el producto.

El objetivo principal de esta fase es detectar y corregir posibles problemas.

2.1.4 Diseño web centrado en el usuario

El Diseño Web Centrado en el Usuario se caracteriza por asumir que todo el proceso de diseño y desarrollo del sitio web debe estar conducido por el usuario, sus necesidades, características y objetivos. Centrar el diseño en sus usuarios (en oposición a centrarlo en las posibilidades tecnológicas o en nosotros mismos como diseñadores) implica involucrar desde el comienzo a los usuarios en el proceso de desarrollo del sitio; conocer cómo son, qué necesitan, para qué usan el sitio; testar el sitio con los propios usuarios; investigar cómo reaccionan ante el diseño, cómo es su experiencia de uso; e innovar siempre con el objetivo claro de mejorar la experiencia del usuario.

El proceso de Diseño Web Centrado en el Usuario propuesto en este trabajo se divide en varias fases o etapas, algunas de las cuales tienen carácter iterativo. Sirva como aproximación el siguiente esquema:

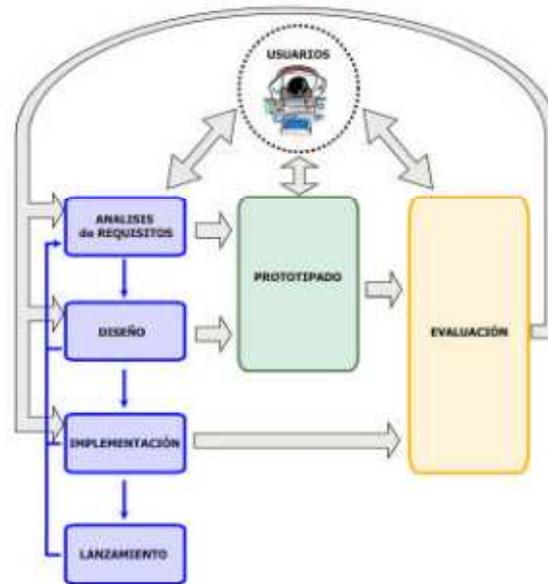


Fig. 2.1 Diseño web centrado en el usuario

Fuente: <http://lsi.ugr.es/~mgea/workshops/coline02/Articulos/toni.pdf>

2.1.4.1 Planificación

En esta etapa se identifican los objetivos del sitio, así como las necesidades, requerimientos y objetivos de la audiencia potencial. Confrontando esta información se definen los requerimientos del sitio web, entre los que podemos contar requerimientos técnicos (back-end y front-end), recursos humanos y perfiles profesionales necesarios, y adecuación del presupuesto disponible.

El diseñador debe obtener información precisa tanto de las necesidades y objetivos del proveedor como del usuario. Más difícil, pero al mismo tiempo más importante, es obtener esta información del usuario: Qué necesita, cuáles son sus objetivos, cómo se comporta y actúa, cuál será el contexto de uso y cómo afectará a la interacción, experiencia y conocimientos previos.

Como se puede ver, la etapa de planificación se basa casi completamente en la recogida, análisis y ordenación de toda la información posible, con el objetivo de

tener una base sólida sobre la que poder tomar decisiones de diseño en las siguientes etapas del proceso.

2.1.4.2 Diseño

La etapa de Diseño es el momento del proceso de desarrollo para la toma de decisiones acerca de cómo diseñar o rediseñar, en base siempre al conocimiento obtenido en la etapa de planificación, así como a los problemas de usabilidad descubiertos en etapas de prototipado y evaluación.

2.1.4.3 Modelado del Usuario

Consiste en la definición de clases o perfiles de usuarios en base a atributos comunes. Los atributos sobre los que se hará la clasificación dependen de la información que se tenga de la audiencia, pero normalmente se tratarán de atributos tales como necesidades de información, condiciones de acceso, experiencia y conocimientos.

La función de esta técnica es la de servir de soporte para la toma de decisiones en el diseño del sitio, permitiendo al desarrollador realizar un diseño centrado en el usuario, o más correctamente, en "algún" usuario.

Es demasiado común que el diseñador se imagine a sí mismo usando el sitio y por tanto sea incapaz de comprender por qué a alguien le puede resultar difícil, incómodo y hasta frustrante su uso. Estos arquetipos de usuarios conseguirán precisamente que el diseñador tenga en mente a un usuario 'real', con limitaciones, habilidades y necesidades reales.

2.1.4.4 Diseño Conceptual

El objetivo de la fase de Diseño Conceptual es definir el esquema de organización, funcionamiento y navegación del sitio. No se especifica qué apariencia va a tener el sitio, sino que se centra en el concepto mismo del sitio: su arquitectura de información.

La "estructura" del sitio web se refiere precisamente a las conexiones y relaciones entre páginas, a la topología de la red de páginas, así como a la granularidad de los elementos de información contenidos en las páginas; y la "navegación" a las posibilidades y forma en que cada página presenta las opciones de desplazamiento hacia otras páginas.

La definición de la estructura del sitio puede hacerse desde dos enfoques diferentes y complementarios: aproximación descendente y ascendente. En la descendente se trata de estructurar del "todo" a las "partes", dividir los contenidos en páginas y definir los enlaces entre páginas. En la Ascendente, por el contrario, se definen los bloques mínimos de información, estructuración que va más allá de la propia segmentación de información en páginas.

Una vez definida la estructuración del sitio es necesario documentarla, para así tener un modelo de referencia sobre el que sustentar el desarrollo del sitio. La forma de documentar arquitecturas se suele hacer a través de grafos y esquemas, con el objetivo de que sean de fácil y rápida comprensión por todos los miembros del equipo de desarrollo.

Si la arquitectura es ascendente normalmente se documentará a través de diagramas entidad-relación. Por otro lado, cuando la arquitectura a documentar es la descendente, para sitios web proponemos el uso del vocabulario gráfico de **Garret (2002)**.

Otras tareas a llevar a cabo por el Arquitecto de Información o diseñador en la fase de Diseño Conceptual son: Definir sistemas de clasificación para los contenidos; Elaborar índices y mapas del sitio; Aplicar metadatos a cada una de las páginas y sub-elementos de información; y Definir el Sistema de Rotulado **(Rosenfeld, Morville; 2002)**.

Entre las técnicas de Diseño Centrado en el Usuario a aplicar en la etapa de Diseño Conceptual destacamos, por su utilidad y facilidad de ser llevada a cabo, la técnica de "card sorting" u ordenación de tarjetas. Ésta se basa en la observación

de cómo los usuarios agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con las diferentes categorías o secciones temáticas del sitio web.

2.1.4.5 Prototipado

La etapa de prototipado se basa en la elaboración de modelos o prototipos de la interfaz del sitio. Su aspecto no se corresponde exactamente con el que tendrá el sitio una vez finalizado, pero pueden servir para evaluar la usabilidad del sitio sin necesidad de esperar a su implementación.

Según **Floría Cortés (2000)**, podemos clasificar los tipos de prototipado según el nivel de funcionalidad reproducida:

- Prototipado horizontal: Se reproduce gran parte del aspecto visual del sitio, pero sin que esos modelos de interfaz estén respaldados por la funcionalidad real que tendrá finalmente el sitio.
- Prototipado vertical: Se reproduce únicamente el aspecto visual de una parte del sitio, pero la parte reproducida poseerá la misma funcionalidad que el sitio web una vez implementado.

Según el grado de fidelidad o calidad del prototipo se distingue entre:

- Prototipado de alta fidelidad: El prototipo será muy parecido al sitio web una vez terminado.
- Prototipado de baja fidelidad: El aspecto del prototipo distará bastante del que tenga el sitio web final.

Otra forma de realizar prototipos es mediante la reproducción del aspecto del sitio a través de herramientas software. Mediante el procesador de textos o un simple editor HTML podemos esbozar cómo será la interfaz del sitio.

La utilidad real del prototipado se fundamenta en que no tendría sentido empezar a implementar una interfaz web si no nos hemos asegurado antes de que el diseño es usable.

2.1.4.6 Evaluación

Existe una gran variedad de métodos para evaluación de usabilidad como se detalla a continuación:

2.1.4.6.1 Método por Inspección: Evaluación Heurística

La Evaluación Heurística es un tipo de método de inspección, que tiene como ventaja la facilidad y rapidez con la que se puede llevar a cabo.

Este tipo de evaluación normalmente la lleva a cabo un grupo reducido de evaluadores que, en base a su propia experiencia, fundamentándose en reconocidos principios de usabilidad (heurísticos), y apoyándose en guías elaboradas para tal fin, evalúan de forma independiente el sitio web, contrastando finalmente los resultados con el resto de evaluadores.

2.1.4.6.2 Método de test con usuarios

El test con usuarios es una prueba de usabilidad que se basa en la observación y análisis de cómo un grupo de usuarios reales utiliza el sitio web, anotando los problemas de uso con los que se encuentran para poder solucionarlos posteriormente.

La ventaja que ofrecen los test de usuarios frente a otro tipo de evaluaciones es que por un lado es una demostración con hechos, por lo que sus resultados son más fiables, y por otro porque posibilitan el descubrimiento de errores de diseño imposibles o difíciles de descubrir mediante la evaluación heurística.

2.1.4.6.3 Implementación y lanzamiento

En la implementación del sitio es recomendable utilizar estándares (HTML, XHTML...) para asegurar la futura compatibilidad y escalabilidad del sitio.

Igualmente es recomendable separar en la implementación contenido de estilo, mediante el uso de hojas de estilo (CSS) del lado del cliente y uso de bases de datos del lado del servidor. De esta forma se facilitará tanto el rediseño del sitio como la posibilidad de adaptación dinámica del diseño a las necesidades de acceso de cada tipo de usuario.

En esta etapa del desarrollo se debe llevar, así mismo, un control de calidad de la implementación, supervisando que todo funcione. Si algo no funciona, sencillamente no se puede usar.

Una vez implementado el sitio y testada su funcionalidad se procede al lanzamiento del sitio, que consiste en su puesta a disposición para los usuarios.

Para asegurar que el sitio llega a su audiencia potencial se hace uso de la promoción. La forma de llevar a cabo una campaña de publicidad o promoción dependerá de la naturaleza y características del sitio web.

Se debe crear expectación, un conocimiento previo del sitio en los potenciales usuarios. Para ello es recomendable que antes del lanzamiento, desde la misma URL que tendrá finalmente el sitio, se ofrezca una página web explicativa de lo que será el sitio, cuándo estará disponible, así como información de contacto.

2.1.4.6.4 Mantenimiento y Seguimiento.

Un sitio web no es una entidad estática, es un objeto vivo cuyos contenidos cambian; cuya audiencia, necesidades y perfiles cambian, y que por lo tanto requiere de continuos rediseños y mejoras.

Estos rediseños deben ser muy sutiles, no se puede cambiar el aspecto y diseño de forma drástica de un día para otro, pues aunque estos cambios estén fundamentados en problemas de usabilidad descubiertos post-lanzamiento, los cambios pueden resultar dramáticos para los actuales usuarios que ya estaban acostumbrados y familiarizados con el actual diseño.

Los problemas de uso no detectados durante el proceso de desarrollo pueden descubrirse a través de varios métodos, principalmente a través de los mensajes y opiniones de los usuarios, y su comportamiento y uso del sitio.

2.2 AQUITECTURA DE DESARROLLO WEB

El modo de crear los documentos HTML ha variado a lo largo de la corta vida de las tecnologías Web pasando desde las primeras páginas escritas en HTML almacenadas en un fichero en el servidor Web hasta aquellas que se generan al vuelo como respuesta a una acción del cliente y cuyo contenido varía según las circunstancias.

Además, el modo de generar páginas dinámicas ha evolucionado, desde la utilización del CGI ,*Common Gateway Interface*, hasta los *servlets* pasando por tecnologías tipo *JavaServer Pages*. Todas estas tecnologías se encuadran dentro de aquellas conocidas como *Server Side*, ya que se ejecutan en el servidor web.

Otro aspecto que completa el panorama son las inclusiones del lado del cliente, *Client Side*, que se refieren a las posibilidades de que las páginas lleven incrustado código que se ejecuta en el cliente, como por ejemplo JavaScript y programas Java.

El esquema general de la situación se puede ver en la figura, donde se muestran cada tipo de tecnología involucrada en la generación e interacción de documentos Web.

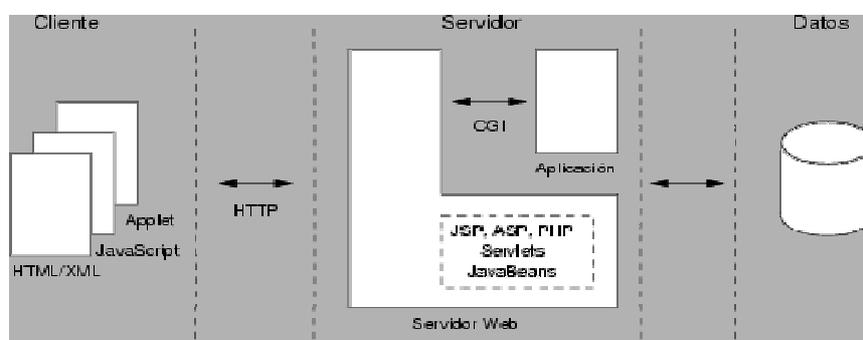


Fig. 2.2 Esquema general de las tecnologías Web.

2.2.1 El Navegador Web, Browser

El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Para todo esto, los fabricantes de navegadores les han dotado de posibilidades de ejecución de programas de tipo *script*, con modelos de objetos que permiten manipular los contenidos de los documentos. Estos lenguajes de programación son VBScript, JScript (ambas de Microsoft) y JavaScript (de Netscape), y proporcionan las soluciones llamadas del lado del cliente, *client side* y permiten realizar validaciones de datos recogidos en las páginas antes de enviarlos al servidor y proporcionan un alto grado de interacción con el usuario dentro del documento.

Otras de las posibilidades de los navegadores es la gestión del llamado HTML dinámico (*Dinamic HTML*, DHTML). Éste está compuesto de HTML, hojas de estilo en cascada, (*Cascade Style Sheets*, CSS), modelo de objetos y *scripts* de programación que permiten formatear y posicionar correctamente los distintos elementos HTML de las páginas Web, permitiendo un mayor control sobre la visualización de las páginas.

2.2.2 El Servidor Web

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados.

El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor, *server side applications*, que vayamos a construir, ya que se ejecutarán en él.

2.2.3 Aplicaciones Multinivel

Al hablar del desarrollo de aplicaciones Web resulta adecuado presentarlas dentro de las aplicaciones multinivel. Los sistemas típicos cliente/servidor pertenecen a la categoría de las aplicaciones de dos niveles. La aplicación reside en el cliente mientras que la base de datos se encuentra en el servidor. En este tipo de aplicaciones el peso del cálculo recae en el cliente, mientras que el servidor hace la parte menos pesada, y eso que los clientes suelen ser máquinas menos potentes que los servidores. Además, está el problema de la actualización y el mantenimiento de las aplicaciones, ya que las modificaciones a la misma han de ser trasladada a todos los clientes.

Para solucionar estos problemas se ha desarrollado el concepto de arquitecturas de tres niveles: interfaz de presentación, lógica de la aplicación y los datos.

La capa intermedia es el código que el usuario invoca para recuperar los datos deseados. La capa de presentación recibe los datos y los formatea para mostrarlos adecuadamente. Esta división entre la capa de presentación y la de la lógica permite una gran flexibilidad a la hora de construir aplicaciones, ya que se pueden tener múltiples interfaces sin cambiar la lógica de la aplicación.

La tercera capa consiste en los datos que gestiona la aplicación. Estos datos pueden ser cualquier fuente de información como una base de datos o documentos XML.

Convertir un sistema de tres niveles a otro multinivel es fácil ya que consiste en extender la capa intermedia permitiendo que convivan múltiples aplicaciones en lugar de una sola como muestra la figura.

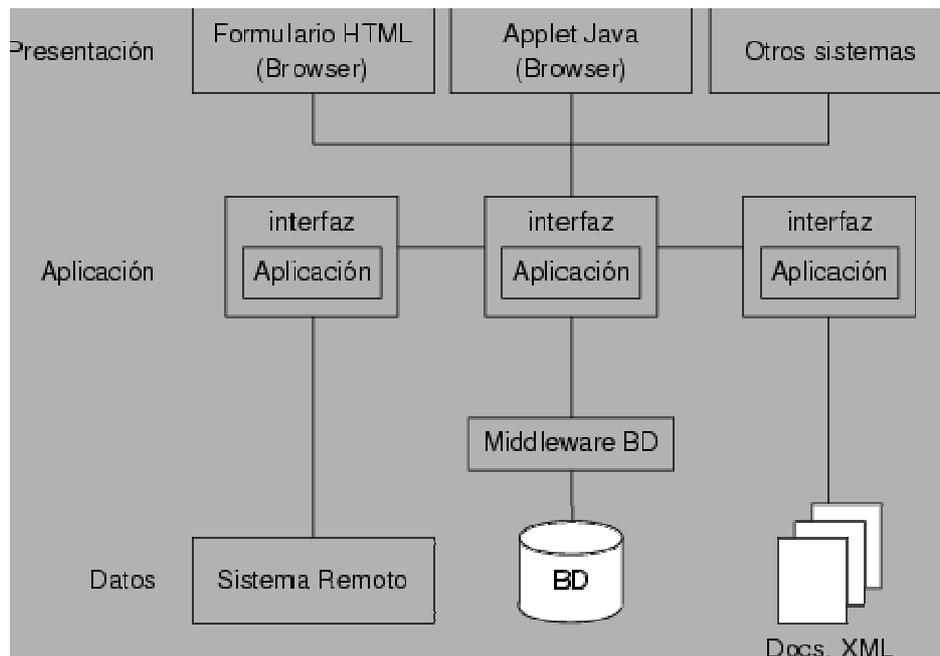


Fig 2.3 Arquitectura Multinivel.

Fuente: <http://www.infor.uva.es/~ivegas/cursos/buendia/pordocente/node21.html>

La arquitectura de las aplicaciones Web suelen presentar un esquema de tres niveles. El primer nivel consiste en la capa de presentación que incluye no sólo el navegador, sino también el servidor web que es el responsable de dar a los datos un formato adecuado. El segundo nivel está referido habitualmente a algún tipo de programa o *script*. Finalmente, el tercer nivel proporciona al segundo los datos necesarios para su ejecución.

Una aplicación Web típica recogerá datos del usuario (primer nivel), los enviará al servidor, que ejecutará un programa (segundo y tercer nivel) y cuyo resultado será formateado y presentado al usuario en el navegador (primer nivel otra vez).

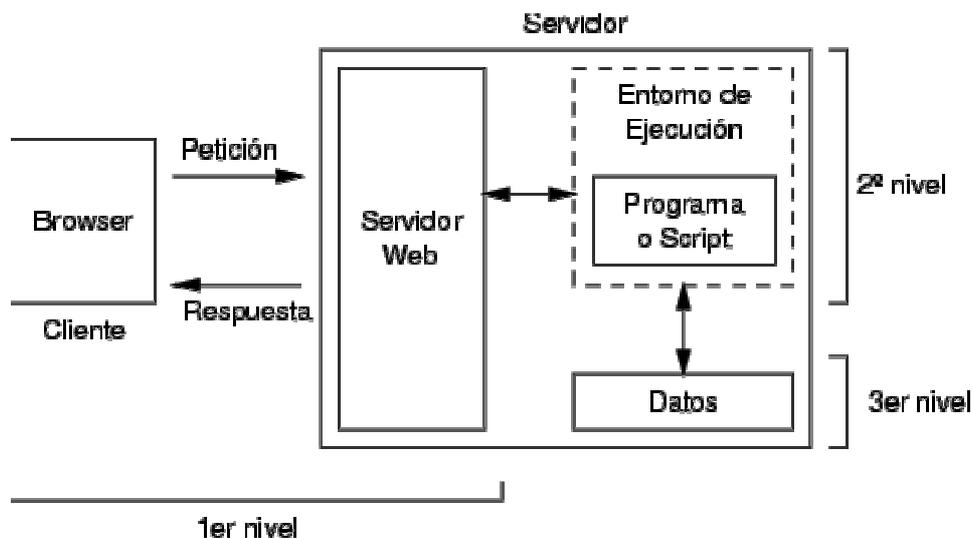


Fig. 2.4 Arquitectura Web de tres niveles.

Lamentablemente, el uso de toda esta tecnología pasa por el dominio de técnicas de programación y de acceso a bases de datos, condición esta que no se puede presuponer en un curso de divulgación como éste. Así, nos vamos a restringir al uso de herramientas básicas a la hora de la construcción de un portal docente.

2.3 HERRAMIENTAS DE DESARROLLO

2.3.1 Linux

Linux es uno de los términos empleados para referirse a la combinación del núcleo o *kernel* libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU) y otra serie de licencias libres.

El núcleo Linux es un sistema operativo libre tipo Unix. Es usualmente utilizado junto a las herramientas GNU como interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar un computador. A la unión de ambas tecnologías, más la inclusión de algunas otras, (como entornos de escritorio e interfaces gráficas) se le conoce como distribución GNU/Linux. Fue

lanzado bajo la licencia pública general de GNU y es desarrollado gracias a contribuciones provenientes de colaboradores de todo el mundo, por lo que es uno de los ejemplos más notables de software libre. Debido a su naturaleza de contenido libre, ambos proyectos invitan a colaborar en ellos de forma altruista.

Se trata de un sistema operativo de 32 bits de libre distribución, desarrollado originalmente por Linus Torvalds, un estudiante de la universidad finlandesa de Helsinki, quien, en 1991, se abocó a la tarea de reemplazar a Minix, un clon de Unix de pequeñas proporciones y finalidad académica desarrollado años antes por Andrew Tannenbaun.

A medida que avanzaba en su desarrollo, Linux fue dejando el código fuente de las sucesivas versiones del kernel y utilidades de Linux a disponibilidad de los usuarios de Internet. Este fue sin duda un gran acierto, ya que hizo posible que una multitud de desarrolladores de todo el mundo se familiarizaran con el código, lo cual en primera instancia significó un gran aporte de sugerencias, evolucionado luego hacia un espectacular ejemplo de desarrollo distribuido de software: centenares de desarrolladores independientes, desde diferentes puntos del planeta tomaron a su cargo la producción de software para Linux, ya sea escribiéndolo desde cero o portándolo desde otras plataformas Unix. Esta modalidad de desarrollo continúa aún hoy y ha permitido a Linux alcanzar un alto nivel de desarrollo y madurez, así también como un amplio grado de aceptación.

Actualmente, Linux posee todas las características que pueden encontrarse en cualquier sistema Unix moderno, incluyendo direccionamiento lineal de 32 bits, memoria virtual, multitarea real, *shared libraries*, módulos de kernel cargables *on-demand*, soporte TCP/IP (incluyendo SLIP, PPP, NFS, etc.), y sistema X-Windows (incluyendo servidores X para los adaptadores de video más populares, y clones de *Motif*, *OpenLook*, *NextStep* y *Windows95* como *window managers*).

También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0.

2.3.1.1 Distribuciones Linux

Es un sistema operativo descendiente de UNIX. Unix es un sistema operativo robusto, estable, multiusuario, multitarea, multiplataforma y con gran capacidad para gestión de redes, Linux fue creado siguiendo estas características. En la década de los ochenta apareció un nuevo sistema, era una versión básica y reducida de Unix llamada Minix, su autor fue Andrew Tanenbaum, el objetivo era crear un acceso a este sistema sin tener que pagar licencias, basados en este sistema el señor Linus B. Torvalds, a mediados de 1991 empezó a trabajar en un proyecto para mejorar las deficiencias de Minix, Torvalds creó la primera versión de Linux (Contracción de Linus y Unix) numerada como versión 0.01. Esta versión solo contenía un Kernel muy rudimentario y para poder realizar cualquier operación se requería que la máquina tuviera instalado Minix. El 5 de Octubre de 1991 fue creada y publicada la versión 0.02 cuando Torvalds logró ejecutar programas como el Bash y el Gcc, después de esta publicación se distribuyó en forma gratuita el código de Linux e invitó a todo aquel que pudiera aportar ideas nuevas y mejorar el código vía Internet, gracias a estos aportes Linux evolucionó rápidamente a las versiones 0.03, 0.10, 0.11 y 0.12. En Marzo de 1992 fue creada la versión 0.95

LINUX es un sistema operativo, compatible Unix. Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado, la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente. El sistema lo forman el núcleo del sistema (kernel) más un gran número de programas / librerías que hacen posible su utilización.

En este caso haremos referencia a Linux RedHat.

2.3.1.1.1 *Linux RedHat 8.0*

Red Hat Linux es instalado con un ambiente gráfico llamado Anaconda, diseñado para su fácil uso por novatos. También incorpora una herramienta llamada Lokkit para configurar las capacidades de Cortafuegos.

Ha salido al mercado oficialmente la nueva versión del sistema operativo para Linux Red Hat 8.0, con la novedad principal de que la tradicional opción a instalar una interfaz tipo GNOME o tipo KDE se sustituye por otra denominada Bluecurve, desarrollada por Red Hat, que permite personalizar más la interfaz de usuario. La interfaz Bluecurve también se incorporará en el software servidor Red Hat Advanced Server, si la respuesta de los usuarios de Red Hat 8.0 es favorable.

Red Hat Linux carece de muchas características debido a posibles problemas de copyright y patentes. Por ejemplo, el soporte al formato MP3 está desactivado tanto en Rhythmbox como en XMMS; en su lugar, Red Hat recomienda usar Ogg Vorbis, que no tiene patentes. Sin embargo, el soporte para MP3 puede ser instalado luego, aunque se requiere el pago de regalías en los Estados Unidos. El soporte al formato NTFS también está ausente, pero también puede ser instalado libremente.

Red Hat se ha comprometido a respetar el *Estándar de Jerarquía del Sistema Ficheros (FHS)* del inglés Filesystem Hierarchy Standard), un documento de consenso que define los nombres y la situación de muchos ficheros y directorios. En el futuro se seguirá el estándar para asegurar la compatibilidad de Red Hat Linux

El documento que define el FHS es la referencia autorizada para cualquier sistema compatible FHS, sin embargo el estándar da pie a la extensibilidad de unas áreas o no define otras. En esta sección se proporciona un resumen del estándar y una descripción de aquellas partes del sistema de ficheros que no cubre el estándar.

Organización de FHS

Los directorios y ficheros aquí anotados, son sólo un subconjunto de los especificados por el FHS.

En el directorio raíz contiene toda la estructura jerárquica del sistema.

Directorio /bin

Este directorio contiene archivos binarios esenciales del sistema, disponibles para todos los usuarios. Por lo que es estático y no se puede compartir. Normalmente no incluye carpetas.

Directorio /boot

Los archivos necesarios para el arranque de Linux se encuentran en esta carpeta. Estático y no puede compartirse.

Directorio /dev

El directorio /dev contiene archivos que representan dispositivos del sistema. Estos archivos son esenciales para el correcto funcionamiento del sistema.

Directorio /etc

El directorio/etc está reservado para archivos de configuración que afectan directamente a su ordenador. No deben colocarse ejecutables en /etc. Los ejecutables que antiguamente se colocaban en /etc deberían estar en /sbin o posiblemente en /bin. Los directorios X11 y skel deben ser subdirectorios de /etc:

/etc

|- X11

|- skel

El directorio X11 es para archivos de configuración de X11 como XF86Config. El

directorio skel es para archivos "esqueleto" (del inglés "skeleton") para usuarios, archivos que se utilizan para rellenar el directorio raíz de un usuario cuando éste es creado.

Directorio /lib

El directorio /lib debería contener sólo las librerías necesarias para ejecutar los binarios en /bin y /sbin. Estas imágenes de librerías compartidas son particularmente importantes para arrancar el sistema y ejecutar comandos en el sistema de ficheros de root.

Directorio /mnt

El directorio /mnt se refiere a sistemas de ficheros montados temporalmente, tales como CD-ROMs y disquetes.

Directorio /opt

El directorio /opt proporciona un área para almacenar habitualmente paquetes de software de una aplicación estática y amplia.

Para paquetes en los que se evite poner ficheros a través del sistema de ficheros, proporciona un sistema de organización predecible y lógico bajo el directorio de paquetes. Esto le aporta al gestor del sistema un modo sencillo de determinar el rol de cada fichero en un paquete particular.

Por ejemplo, si sample fuese el nombre de un paquete de software particular localizado en /opt, todos sus ficheros podrían ser emplazados en directorios dentro de /opt/sample, tales como /opt/sample/bin para binarios y /opt/sample/man para páginas de manual.

Los paquetes grandes que abarcan diferentes subpaquetes, cada uno de los cuales desempeñan una tarea específica, también funcionan con /opt, aportando a este gran paquete un modo estándar de organizarse. De este modo, el paquete

sample tendrá diferentes herramientas cada una de las cuales irá a sus subdirectorios, tales como /opt/sample/tool1 y /opt/sample/tool2, cada uno de los cuales puede tener su propio bin, man y otros directorios similares.

Directorio /root

Dentro está el directorio personal del administrador del sistema. No estático, incompatible.

Directorio /proc

El directorio /proc contiene "ficheros" especiales que o bien extraen información del kernel o bien la envían a éste.

Debido a la gran cantidad de datos que contiene el directorio /proc y a la gran cantidad de maneras utilizadas para comunicar con el kernel, se ha dedicado un capítulo entero a este tema.

Directorio /sbin

El directorio /sbin es para ejecutables usados sólo por el usuario root. Los ejecutables en /sbin sólo se usan para arrancar y montar /usr y ejecutar operaciones de recuperación del sistema. El FHS dice: "/sbin contiene típicamente archivos esenciales para arrancar el sistema además de los binarios en /bin. Cualquier archivo ejecutado tras /usr, será montado (si no surge ningún problema) y ubicado en /usr/sbin. Los binarios de administración de sistema sólo local, deberían ser ubicados en /usr/local/sbin.

Directorio /usr

El directorio /usr es para archivos que puedan ser compartidos a través de todo el sitio. El directorio /usr habitualmente tiene su propia partición y debería ser montable en sólo lectura. Los siguientes directorios deberían ser subdirectorios /usr:

/usr

- | - bin
- | - dict
- | - etc
- | - games
- | - include
- | - kerberos
- | - lib
- | - libexec
- | - local
- | - sbin
- | - share
- | - src
- | - X11R6

El directorio bin contiene ejecutables, doc contiene páginas de documentación incompatibles con FHS, etc contiene ficheros de configuración de sistema, games es para juegos, include contiene los ficheros de cabecera C, kerberos contiene binarios y muchos más ficheros de Kerberos y lib contiene ficheros objeto y librerías que no están diseñadas para ser directamente utilizadas por usuarios o scripts de shell. El directorio libexec contiene programas de pequeña ayuda llamados por otros programas, sbin es para los binarios de administración del sistema (aquéllos que no pertenecen a /sbin), share contiene ficheros que no son de una arquitectura específica, src es para el código fuente y X11R6 es para el sistema X Window (XFree86 de Red Hat Linux).

Directorio /usr/local

El **FHS** dice:

"La jerarquía /usr/local es para uso del gestor del sistema al instalar localmente el software. Necesita ser seguro para ser sobrescrito cuando el software del sistema es compatible entre un grupo de hosts, pero no se encuentra en /usr."

El directorio /usr/local es similar en estructura al directorio /usr.

Directorio /var

Ya que el FHS requiere que sea capaz de montar /usr en sólo lectura, cualquier programa que escriba ficheros log o que necesite los directorios spool o lock debería escribirlos en el directorio /var. El FHS constata que /var es para ficheros de datos variables. Esto incluye ficheros spool, de administración, de registro y ficheros temporales.

Para X Window.

/usr/bin

Contiene los ejecutables no necesarios para la adm. y mantenimiento del sistema.

/usr/sbin.

Librerías necesarias para los binarios.

/usr/local.

Archivos de las aplicaciones.

/var

Contiene archivos que pueden modificarse durante el uso del sistema pero que deben quedar como estaban al volver a arrancar Linux.

Los archivos de Linux pueden tener todos los caracteres que quieras, no existe un formato estándar para sus nombres y no necesitan extensión porque en sus propiedades especifican el tipo de archivo.

Para Linux todo es un archivo. Una partición, un directorio, etc.

2.3.2 Apache

Originalmente basado en el servidor más popular de la época: NCSA httpd a principios de 1995. El nombre Apache es "A PatCH y server".

Desde su origen a evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.

Apache es de acuerdo al estudio hecho por Netcraft, el servidor WWW más popular del momento.

Apache ha demostrado ser substancialmente más rápido que muchos otros servidores libres y compite de cerca con los mejores servidores comerciales

2.3.2.1 Arquitectura Apache

El servidor Apache es un software que está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas del Apache .
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, acepando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

2.3.3 Coldfusion

ColdFusion fue desarrollado por Allaire Corporation para ser una alternativa poderosa y simple de usar a Perl y otras tecnologías CGI. Con cientos de miles de desarrolladores en el mundo que actualmente lo usan, se puede asegurar que ColdFusion ha sido exitoso.

2.3.3.1 Coldfusion y sus Características

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, puede crear y modificar variables igual que en otros lenguajes de programación. Posee controles de flujo de programas, como IF, Switch Case, Loop, etc. 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.

Muchas de las funciones poderosas de ColdFusion, como leer desde y escribir 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. 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 multi-procesador, 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.

2.3.3.2 Componentes Coldfusion

ColdFusion posee los siguientes componentes los cuales lo hacen una herramienta muy eficaz para cualquier tipo de aplicación.

2.3.3.2.1 Tecnología del servidor

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.

2.3.3.2.2 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.

2.3.3.2.3 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++.

- Permite construir aplicaciones web rápidamente.
- 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.
- 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.
- 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.

ColdFusion puede ser usado en un sitio cada vez que se necesita interacción con el usuario. Procesa formularios, hace seguras algunas partes del sitio, y recolecta o publica datos. Se puede usar para construir diarios murales, clientes de POP mail, calendarios en línea, y salas de chat. Se pueden escribir scripts para rastrear estadísticas. Usando ColdFusion se ahorra dinero en mantenimiento. Se gasta mucho más dinero en ajustar el software a nuevos requerimientos que en el desarrollo inicial, ColdFusion es experto en el área de mantenimiento sobre otras herramientas middleware para crear sitios web dinámicos, ya que: Esconde la complejidad, usa menos líneas y son más intuitivas para alcanzar resultados, permite al usuario migrar a otros servidores web y motores de bases de datos con pocos cambios y sin plug-ins externos. Permite setear y olvidarse de los defaults para el acceso a bases de datos, el programador puede setear los detalles de la conexión a la base de datos una vez y después sólo referirse a la fuente de los datos con un simple nombre. En cambio, con ASP, los detalles de la conexión, como username y password, deben ser repetidos en el código cada vez que se utiliza la fuente de los datos. Permite setear y olvidarse de la administración de sesiones. Un archivo global, que es transparentemente incluido al comienzo de cada página puede asignar un ID de sesión y un símbolo que hace difícil perder la sesión y puede asociarla transparentemente (a través de cookies o URLs) con todas las páginas accedidas por un cliente. La creación de símbolos y la verificación de que el símbolo está amarrado a el ID de la sesión se maneja automáticamente. Otros middleware forzan al programador a manejar estos detalles. ColdFusion simplifica el almacenamiento de variables, el programador puede manipular fácilmente las variables apropiadas a su sesión

lógica. Lo mismo pasa con las variables en el servidor, en la aplicación y al nivel de página. Otros middleware necesitan más compromiso del programador y más trabajo para que sea escalable. ColdFusion hace loop implícito sobre consultas y listas. Sabe cómo hacer loop sobre las columnas de una consulta. Obtiene automáticamente la siguiente columna cuando no hay nada más que hacer con la columna actual. Se detiene automáticamente cuando no hay más columnas. Estos detalles no se codifican y no necesitan ser revisados durante el mantenimiento. ColdFusion genera y envía javascript transparentemente on the fly cuando ciertos tags de input son utilizados. Esto facilita el chequeo de inputs del lado del cliente sin forzar al programador a escribir, revisar y modificar javascript para hacer esto. ColdFusion usa menos líneas de código y son más intuitivas. supongamos que se necesita desplegar información de una base de datos ColdFusion tiene un buen manejo de errores y depuración. Cuando un motor de bases de datos arroja un error, ColdFusion sugiere causas posibles. Le permite al programador crear sus propios manejadores de errores cuando se necesita cuidado especial. Con ColdFusion se pueden personalizar los mensajes de error para situaciones específicas y puede proveer un nivel de detalle para los usuarios y uno diferente para los mantenedores. Las aplicaciones en ColdFusion pueden cambiar de plataformas y motores de bases de datos. Se pueden cambiar las aplicaciones de ColdFusion a diferentes sistemas operativos y servidores web con pequeños cambios, y se puede incluso cambiar los motores de bases de datos con un poco más de esfuerzo. Los proyectos pequeños en ColdFusion pueden crecer sin abandonar código ni empezar del comienzo. Ventajas inigualables de ColdFusion viene con habilidades que otros middleware no pueden alcanzar sin necesitar agregados. Viene con un motor para indexar sitios web. Se puede realizar balance de carga dinámico, se pueden mantener porciones de código como propietarias. Se pueden hacer consultas persistentes para mayor velocidad. Se integra con el monitor de desempeño y el monitor de seguridad de NT. Se puede modificar el registro. Permite continuar usando scripts CGI existentes. El código ColdFusion puede

ser generado on the fly con herramientas que vienen con su usual editor, ColdFusion Studio. El lenguaje es extensible, convierte datos hacia y desde XML y permite conectar un sitio con otros como en un browser para extender el alcance de la aplicación. En resumen, según CFM resources: Dado que la gramática de ColdFusion lleva a menos líneas de código más entendibles, los tiempos de revisión y ajuste se acortan en un tercio o más. Esto gracias a aspectos del lenguaje que esconden la complejidad mientras proveen poder.

2.3.4 ActionScript

El ActionScript es el lenguaje de programación que ha utilizado Macromedia Flash desde sus comienzos, y que por supuesto, emplea Flash MX 2004. A grandes rasgos, podemos decir que el ActionScript nos permitirá realizar con Flash MX 2004 todo lo que nos propongamos, ya que nos da el control absoluto de todo lo que rodea a una película Flash.

2.3.4.1 Características generales del ActionScript

Es el lenguaje de programación propio de Flash, tal y como el Lingo lo es de Macromedia Director, por ejemplo. El ActionScript está basado en la especificación ECMA-262, al igual que otros lenguajes como Javascript.

Es un lenguaje de script, esto quiere decir que no hará falta crear un programa completo para conseguir resultados, normalmente la aplicación de fragmentos de código ActionScript a los objetos existentes en nuestras películas nos permiten alcanzar nuestros objetivos. Es un lenguaje de programación orientado a objetos, tiene similitudes, por tanto, con lenguajes tales como los usados en el Microsoft Visual Basic, en el Borland Delphi etc... y aunque, evidentemente, no tiene la potencia de un lenguaje puramente orientado a objetos derivado del C o del Pascal como los anteriores, cada versión se acerca más a un lenguaje de este tipo. Así, la versión 2.0 recientemente estrenada en el Flash MX 2004 es mucho más

potente y mucho más "orientado a objetos" que su anterior versión 1.0 presenta muchísimos parecidos con el Javascript.

2.3.5 JavaScript

JavaScript es un lenguaje interpretado, al igual que VisualBasic, Perl, TCL... (lenguajes de script) sin embargo, posee una característica que lo hace especialmente idóneo para trabajar en Web, ya que son los navegadores que utilizamos para viajar por ella los que interpretan (y por tanto ejecutan) los programas escritos en JavaScript. De esta forma, podemos enviar documentos a través de la Web que llevan incorporados el código fuente de programas, convirtiéndose de esta forma en documentos dinámicos, y dejando de ser simples fuentes de información estáticas.

Los programas en JavaScript no son la primera forma que conoce la Web para transformar información, dado que el uso de CGIs está ampliamente difundido. La diferencia básica que existe entre un programa CGI y uno escrito en JavaScript es que el CGI se ejecuta en el servidor de páginas Web mientras que el programa en Javascript se ejecuta en el cliente (es decir, en el navegador). Por regla general, el CGI necesita unos datos de entrada (que normalmente se proporcionan mediante un formulario), los procesa y emite un resultado en forma de documento HTML. Esto implica tres transacciones en la red:

- Cargar la página del formulario
- Enviar los datos al servidor
- Recibir la nueva página que ha generado el CGI.

Por el contrario, los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados. Sin embargo, esto no significa que los CGI vayan a ser substituidos por JavaScript. Las dos principales características de

JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones), y por otro JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas X-Windows). Esto implica que gran parte de la programación en JavaScript se centra en describir objetos (con sus variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos. Por último, es necesario resaltar que hay dos tipos de JavaScript. Por un lado está el que se ejecuta en el cliente. Este es el JavaScript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un JavaScript que se ejecuta en el servidor. Este, más reciente, se denomina LiveWire JavaScript. Nos centraremos solo en el primero, el Navigator JavaScript.

2.3.5.1 Que no es Javascript

Ante todo no es un lenguaje de propósito general. No permite un control absoluto sobre los recursos del ordenador, tal y como estamos acostumbrados al crear un programa (aunque las últimas innovaciones de los navegadores en cuestiones de seguridad le están dando un mayor operabilidad) . Cada programa en JavaScript solo tiene acceso al documento HTML en el que va inmerso y, si acaso, a las ventanas en las que se ejecuta el navegador dentro del cual se está ejecutando el programa en JavaScript. Como ya se mencionó, tampoco es un sustituto de los CGIs. Ciertas cosas no se pueden hacer en JavaScript, especialmente las relacionadas con acceso a ficheros en todos sus sentidos. Sin embargo es muy útil para depurar errores en los datos antes de pasárselos al CGI que los trata, por lo que el uso combinado de JavaScript y CGIs redundará en un mejor tratamiento de datos y un menor número de transacciones en la Web cuando se usan los CGIs. Por último, no es un lenguaje orientado a objetos, ya que, por ejemplo, no existe el concepto de clase. Es

basado en objetos, de modo que se trabaja directamente con instancias de objetos.

2.3.5.2 Diferencias entre Javascript y Java

Aunque pudiera pensarse que son una misma cosa, son dos lenguajes totalmente diferentes. Ciertos detalles son similares para ambos (lenguajes muy relacionados con la Web, basados en un paradigma de programación orientada a objetos, permiten trabajar con objetos inmersos en un documento HTML...), pero en esencia son dos lenguajes independientes.

He aquí algunas de las diferencias más notables entre Java y Javascript:

- Java es un lenguaje de propósito general, y por tanto se puede utilizar para programar aunque no sea para la Web. Javascript es específico de la Web.
- JavaScript es un lenguaje interpretado y ejecutado por el cliente. Java es un lenguaje compilado por el servidor y que puede ser ejecutado en el cliente.
- JavaScript es basado en objetos. Java es orientado a objetos. Los objetos usados en JavaScript existen cuando empieza a ejecutarse el programa; no existen clases ni por tanto ninguna de las características que de ello se derivan (herencia o encapsulación).

El código del programa en JavaScript va inscrito en la página HTML en la que se ejecuta. Javascript es un lenguaje débilmente tipificado, no es necesario declarar el tipo de los datos y se realizan conversiones de unos a otros automáticamente cuando es necesario y si es viable. Java es, por contrario, fuertemente tipificado, como C o Pascal. El hecho de que algunas personas confundan estos dos lenguajes puede deberse a que el compilador de Java crea un código intermedio que debe ser nuevamente traducido por java virtual machine, dándose la circunstancia de que los navegadores pueden traducir dicho código intermedio y, por tanto, pueden ejecutar programas escritos en Java. JavaScript se ejecuta tanto los navegadores de Netscape como en los Internet Explorer de Microsoft (NS e IE respectivamente). En lo básico son

compatibles, pero desde la última versión del lenguaje, la 1.2, han surgido ciertas diferencias que deben ser tenidas en cuenta por aquellos que diseñen páginas html (dichas diferencias se centran especialmente en la gestión del nuevo modelo de eventos).

2.3.6 Oracle

Oracle es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation.

Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.

2.3.6.1 Historia

Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre SGBD (Sistemas Gestores de Base de Datos) de George Koch. Computer World definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

En la actualidad, Oracle (Nasdaq: ORCL) todavía encabeza la lista. La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo y en las oficinas de 98 de las 100 empresas Fortune 100. Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software independiente más grande del mundo.

2.3.6.2 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 comparten 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.²⁵

2.3.6.3 Estructuras de Proceso

El servidor se vale de una serie de procesos que son el enlace entre las estructuras físicas y de memoria. A continuación se describen cada proceso y el papel que juega en la gestión de laBD. Todo esto se puede ver en la siguiente figura.

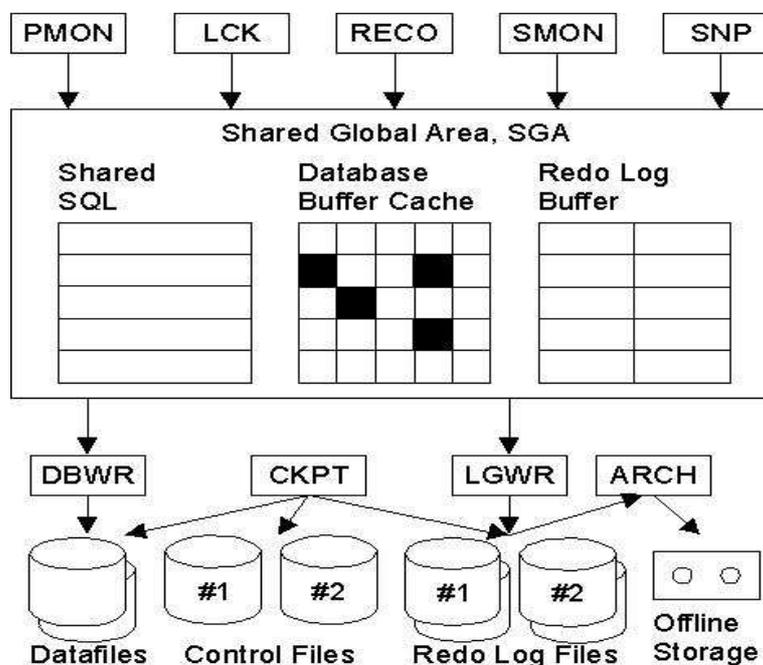


Fig. 2.5 Estructuras de procesos físicos y de memoria

Fuente: <http://www.infor.uva.es/~jvegas/cursos/bd/orarq/orarq.html#1.5>

2.3.6.3.1 System Monitor, SMON

El SMON es el supervisor del sistema y se encarga de todas las recuperaciones que sean necesarias durante el arranque. Esto puede ser necesario si la BD se paró inesperadamente por fallo físico, lógico u otras causas. Este proceso realiza la recuperación de la instancia de BD a partir de los ficheros redo log. Además limpia los segmentos temporales no utilizados y compacta los huecos libres contiguos en los ficheros de datos. Este proceso se despierta regularmente para comprobar si debe intervenir.

2.3.6.3.2 Process Monitor, PMON

Este proceso restaura las transacciones no validadas de los procesos de usuario que abortan, liberando los bloqueos y los recursos de la SGA. Asume la identidad del usuario que ha fallado, liberando todos los recursos de la BD que estuviera utilizando, y anula la transacción cancelada. Este proceso se despierta regularmente para comprobar si su intervención es necesaria.

2.3.6.3.3 Database Writer, DBWR

El proceso DBWR es el responsable de gestionar el contenido de los buffers de datos y del caché del diccionario. Él lee los bloques de los ficheros de datos y los almacena en la SGA. Luego escribe en los ficheros de datos los bloques cuyo contenido ha variado. La escritura de los bloques a disco es diferida buscando mejorar la eficiencia de la E/S.

Es el único proceso que puede escribir en la BD. Esto asegura la integridad. Se encarga de escribir los bloques de datos modificados por las transacciones, tomando la información del buffer de la BD cuando se valida una transacción. Cada validación no se lleva a la BD física de manera inmediata sino que los bloques de la BD modificados se vuelcan a los ficheros

de datos periódicamente o cuando sucede algún checkpoint o punto de sincronización: grabación diferida:

- Los bloques del buffer de la BD (bloques del segmento de rollback y bloques de datos) menos recientemente utilizados son volcados en el disco continuamente para dejar sitio a los nuevos bloques.
- El bloque del segmento de rollback se escribe SIEMPRE antes que el correspondiente bloque de datos.
- Múltiples transacciones pueden solapar los cambios en un sólo bloque antes de escribirlo en el disco.

Mientras, para que se mantenga la integridad y coherencia de la BD, todas las operaciones se guardan en los ficheros de redo log. El proceso de escritura es asíncrono y puede realizar grabaciones multibloque para aumentar la velocidad.

2.3.6.3.4 Log Writer, LGWR

El proceso LGWR es el encargado de escribir los registros redo log en los ficheros redo log. Los registros redo log siempre contienen el estado más reciente de la BD, ya que puede que el DBWR deba esperar para escribir los bloques modificados desde el buffer de datos a los ficheros de datos. Conviene tener en cuenta que el LGWR es el único proceso que escribe en los ficheros de redo log y el único que lee directamente los buffers de redo log durante el funcionamiento normal de la BD.

Coloca la información de los redo log buffers en los ficheros de redo log. Los redo log buffers almacenan una copia de las transacciones que se llevan a cabo en la BD. Esto se produce:

- a cada validación de transacción, y antes de que se comunique al proceso

- que todo ha ido bien,
- cuando se llena el grupo de buffers de redo log
- cuando el DBWR escribe buffers de datos modificados en disco.

Así, aunque los ficheros de DB no se actualicen en ese instante con los buffers de BD, la operación queda guardada y se puede reproducir. Oracle no tiene que consumir sus recursos escribiendo el resultado de las modificaciones de los datos en los archivos de datos de manera inmediata. Esto se hace porque los registros de redo log casi siempre tendrán un tamaño menor que los bloques afectados por las modificaciones de una transacción, y por lo tanto el tiempo que emplea en guardarlos es menor que el que emplearía en almacenar los bloques sucios resultado de una transacción; que ya serán trasladados a los ficheros por el DBWR. El LGWR es un proceso único, para asegurar la integridad. Es asíncrono. Además permite las grabaciones multibloque.

2.3.6.3.5 *Checkpoint, CKPT*

Este proceso escribe en los ficheros de control los checkpoints. Estos puntos de sincronización son referencias al estado coherente de todos los ficheros de la BD en un instante determinado, en un punto de sincronización. Esto significa que los bloques sucios de la BD se vuelcan a los ficheros de BD, asegurándose de que todos los bloques de datos modificados desde el último checkpoint se escriben realmente en los ficheros de datos y no sólo en los ficheros redo log; y que los ficheros de redo log también almacenan los registros de redo log hasta este instante. La secuencia de puntos de control se almacena en los ficheros de datos, redo log y control. Los checkpoints se producen cuando:

- un espacio de tabla se pone inactivo, offline,

- se llena el fichero de redo log activo,
- se para la BD,
- el número de bloques escritos en el redo log desde el último checkpoint alcanza el límite definido en el parámetro LOG_CHECKPOINT_INTERVAL,
- cuando transcurra el número de segundos indicado por el parámetro LOG_CHECKPOINT_TIMEOUT desde el último checkpoint.

Está activo si el parámetro CHECKPOINT_PROCESS tiene un valor verdadero.

2.3.6.3.6 Archiver, ARCH

El proceso archiver tiene que ver con los ficheros redo log. Por defecto, estos ficheros se reutilizan de manera cíclica de modo que se van perdiendo los registros redo log que tienen una cierta antigüedad. Cuando la BD se ejecuta en modo ARCHIVELOG, antes de reutilizar un fichero redo log realiza una copia del mismo. De esta manera se mantiene una copia de todos los registros redo log por si fueran necesarios para una recuperación. Este es el trabajo del proceso archiver.

2.3.6.3.7 Recovered, RECO

El proceso de recuperación está asociado al servidor distribuido. En un servidor distribuido los datos se encuentran repartidos en varias localizaciones físicas, y estas se han de mantener sincronizadas. Cuando una transacción distribuida se lleva a cabo puede que problemas en la red de comunicación haga que una de las localizaciones no aplique las modificaciones debidas. Esta transacción dudosa debe ser resuelta de algún modo, y esa es la

tarea del proceso recuperador. Está activo si el parámetro DISTRIBUTED_TRANSACTIONS tiene un valor distinto de 0.

2.3.6.3.8 Lock, LCK

El proceso de bloqueo está asociado al servidor en paralelo.

2.4 PARADIGMA ESPIRAL ORIENTADO A LA WEB

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.

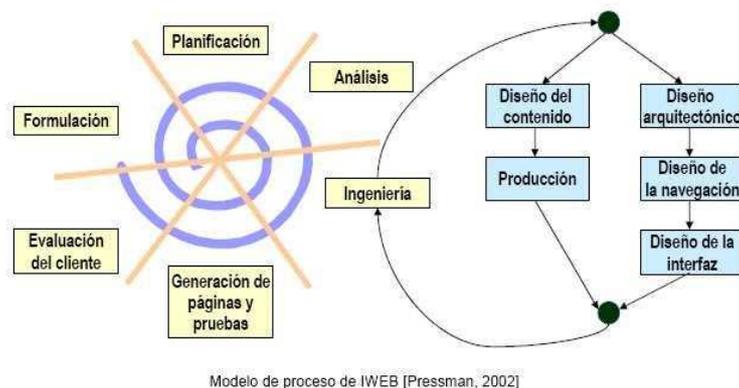


Fig. 2.6 Modelo de Pressman

2.4.1 Etapas

A continuación se indican las etapas que presenta este Modelo:

2.4.1.1 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 planificación más toscamente granulada para los incrementos subsiguientes.

2.4.1.2 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.

2.4.1.3 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.

2.4.1.4 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.

2.4.1.5 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: CORBA, DCaM 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 entornos (por ejemplo, con diferentes navegadores).

2.4.1.6 Evaluación Cliente

En este punto es donde se solicitan cambios, tiene lugar la ampliación del ámbito. En esta actividad se revisan los incrementos producidos como parte del proceso.²

2.5 MODELO OOHDM (OBJECT-ORIENTED HYPERMEDIA DESIGN MODEL) O MÉTODO DE DISEÑO DE HIPERMEDIA ORIENTADO A OBJETOS

Las metodologías tradicionales de ingeniería de software, o las metodologías para sistemas de desarrollo de información, no contienen una buena abstracción capaz de facilitar la tarea de especificar aplicaciones hipermedia. El tamaño, la complejidad y el número de aplicaciones crecen en forma acelerada en la actualidad, por lo cual una metodología de diseño sistemática es necesaria para disminuir la complejidad y admitir evolución y reusabilidad. Producir aplicaciones en las cuales el usuario pueda aprovechar el potencial del paradigma de la navegación de sitios web, mientras ejecuta transacciones 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 hipertexto 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 navegacional de la aplicación, de aquellas relacionadas con el modelo del dominio.

OOHDM propone el desarrollo de aplicaciones hipertexto a través de un proceso compuesto por cuatro etapas: diseño conceptual, diseño navegacional, diseño de interfaces abstractas e implementación.

2.5.1 Diseño Conceptual

Durante esta actividad se construye un esquema conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. En las aplicaciones hipertexto convencionales, cuyos componentes de hipertexto no son modificados durante la ejecución, se podría usar un modelo de datos semántico estructural (como el modelo de entidades y relaciones). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intenten ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos.

En OOHDM, el esquema conceptual está construido por clases, relaciones y subsistemas. Las clases son descritas como en los modelos orientados a objetos tradicionales. Sin embargo, los atributos pueden ser de múltiples tipos para representar perspectivas diferentes de las mismas entidades del mundo real. Se usa notación similar a UML (Lenguaje de Modelado Unificado) y tarjetas de clases y relaciones similares a las tarjetas CRC (Clase Responsabilidad Colaboración). El esquema de las clases consiste en un conjunto de clases

conectadas por relaciones. Los objetos son instancias de las clases. Las clases son usadas durante el diseño navegacional para derivar nodos, y las relaciones que son usadas para construir enlaces.

2.5.2 Diseño Navegacional

La primera generación de aplicaciones web fue pensada para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermedia. En OOHDM, la navegación es considerada un paso crítico en el diseño aplicaciones. Un modelo navegacional es construido como una vista sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual. El diseño de navegación es expresado en dos esquemas: el esquema de clases navegacionales y el esquema de contextos navegacionales. En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos. La principal estructura primitiva del espacio navegacional es la noción de contexto navegacional. Un contexto navegacional es un conjunto de nodos, enlaces, clases de contextos, y otros contextos navegacionales (contextos anidados). Pueden ser definidos por comprensión o extensión, o por enumeración de sus miembros.

Los contextos navegacionales juegan un rol similar a las colecciones y fueron inspirados sobre el concepto de contextos anidados. Organizan el espacio navegacional en conjuntos convenientes que pueden ser 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 contexto particular.

2.5.3 Diseño de Interfaz Abstracta

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

Una clara separación entre diseño navegacional y diseño de interfaz abstracta permite construir diferentes interfaces para el mismo modelo navegacional, dejando un alto grado de independencia de la tecnología de interfaz de usuario.

El aspecto de la interfaz de usuario de aplicaciones interactivas (en particular las aplicaciones *web*) es un punto crítico en el desarrollo que las modernas metodologías tienden a descuidar. En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz del usuario de la aplicación de hipermedia.

El modelo de interfaz ADVs (Vista de Datos Abstracta⁵) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las ADVs en la pantalla real son hechas en la fase de implementación.

2.5.4 Implementación

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

Al llegar a esta fase, el primer paso que debe realizar el diseñador es

definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada.³

2.6 UML

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares.

En 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de un metamodelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de 1997, los seis

promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML, llamado UML versión 1.1. Este documento fue aprobado por el OMG en Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el

1 de abril de 1999.

Los inconvenientes de UML son los siguientes:

- UML no es una metodología.
- No cubre todas las necesidades de especificación de un proyecto software. Por ejemplo, No define los documentos textuales.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

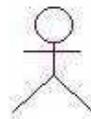
- Diagramas de Casos de Uso para modelar los procesos.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.

- Diagramas de Implementación para modelar la distribución del sistema.

2.6.1 Diagramas de Casos de Uso

Estos permiten modelar el comportamiento de un sistema, el cual sirve para visualizar gráficamente cada uno de los componentes del sistema, la relación que existe entre ellos, quien hará uso de dicho comportamiento sin importar como estén implementados, esto nos ayuda a describir las funciones del sistema desde el punto de vista de los actores (usuarios, operadores, sistemas externos). En estos diagramas se especifica cada función que realizará el sistema, obteniendo así el comportamiento total del sistema en desarrollo.

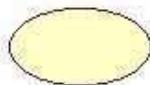
2.6.1.1 Actor



Un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

Como ejemplo a la definición anterior, tenemos el caso de un sistema de ventas en que el rol de Vendedor con respecto al sistema puede ser realizado por un Vendedor o bien por el Jefe de Local.

2.6.1.2 Casos de Uso



Es una operación/tarea específica que se realiza tras una orden de algún agente

externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

2.6.1.3 Relaciones

2.6.1.3.1 Asociación



Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

2.6.1.3.2 Dependencia o Instanciación



Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

2.6.1.3.3 Generalización



Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia**(<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

extends: Se recomienda utilizar cuando un caso de uso es similar a otro (características).

uses: Se recomienda utilizar cuando se tiene un conjunto de características

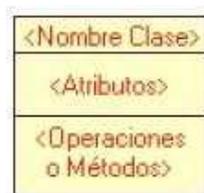
que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

2.6.2 Diagrama de Clases

El Diagrama de Clases es el principal para el análisis y diseño del sistema. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso debería aportar información para establecer las clases, objetos, atributos y operaciones.

2.6.2.1 Clase

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio (una Casa, un Auto, una Cuenta Corriente, etc.).



Tanto los atributos como los métodos pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno que pueden ser:

- Public: es accesible desde todos lados, dentro y fuera de la clase.
- Private: solo será accesible dentro de la clase.
- Protected: no será accesible desde fuera de la clase, pero si de las que se deriven de esta.

2.6.2.1.1 Relaciones entre clases

Previo daremos el concepto de cardinalidad de relaciones, la cardinalidad

indica el grado y nivel de dependencia que se anotan en los extremos de cada relación, y son los siguientes:

Uno o muchos 1..* (1..n)

0 muchos 0..* (0..n)

Número fijo m (m denota el número)

2.6.2.1.2 (Herencia/Generalización)



Indica que una subclase hereda los métodos y atributos especificados por una Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public y protected).

2.6.2.1.3 Agregación



Para modelar objetos complejos, no bastan los tipos de datos básicos que proveen los lenguajes: enteros, reales y secuencias de caracteres. Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos las siguientes posibilidades:

Por Valor: Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada Composición (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").

Por Referencia: Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (el objeto base utiliza al incluido para su funcionamiento).

2.6.2.1.4 Asociación



La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre sí. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

2.6.2.1.5 Dependencia o Instancia



Representa un tipo de relación muy particular, en la que una clase es instanciada (su instanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

El uso más particular de este tipo de relación es para denotar la dependencia que tiene una clase de otra.⁷

2.6.3 Diagrama de Objetos

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: nombre_del_objeto: nombre_de_la_clase. Puede representarse un objeto sin un nombre específico, entonces solo aparece el nombre de la clase.

Todas las relaciones son las mismas que en el diagrama de Clases por lo que no se las volverá a mencionar, la única diferencia que posee es que los atributos están instanciados como se muestra a continuación en la figura:

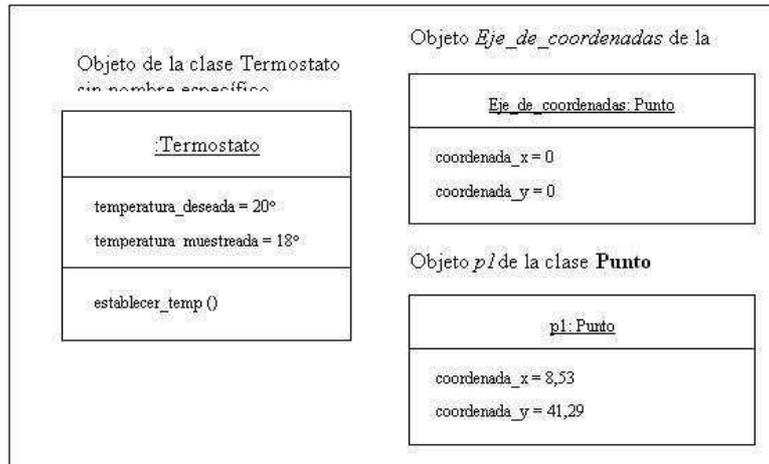


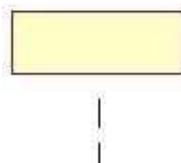
Fig. 2.7 Ej. Diagrama de objetos

Una clase describe un conjunto de objetos con características y comportamiento idéntico.

2.6.4 Diagrama de Secuencia

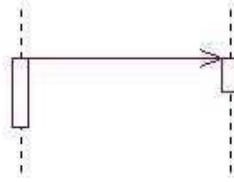
Muestra la secuencia de mensajes entre objetos durante un escenario concreto. Cada objeto viene dado por una barra Vertical. El tiempo transcurre de arriba abajo. Cuando existe demora entre el envío y la atención se puede indicar usando una línea oblicua.

2.6.4.1 Objeto/Actor



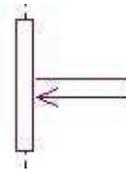
El rectángulo representa una instancia de un Objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

2.6.4.2 Mensaje a otro Objeto



Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

2.6.4.3 Mensaje al mismo Objeto



No solo llamadas a métodos de objetos externos pueden realizarse, también es posible visualizar llamadas a métodos desde el mismo objeto en estudio.9

2.6.5 Diagrama de Colaboración

Son útiles en la fase exploratoria para identificar objetos. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás. La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.

2.6.5.1 Objeto

Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase.

2.6.5.2 Enlaces

Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Esta acompañada por un número que indica el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (association y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global.

2.6.5.3 Flujo de Mensajes

Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace.

2.6.5.4 Marcadores de Creación y Destrucción de Objetos

Puede mostrarse en la gráfica cuáles objetos son creados y destruidos, agregando una restricción con la palabra new o delete, respectivamente, cercana al rectángulo del objeto.

2.6.5.5 Objeto Compuesto

Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. 10

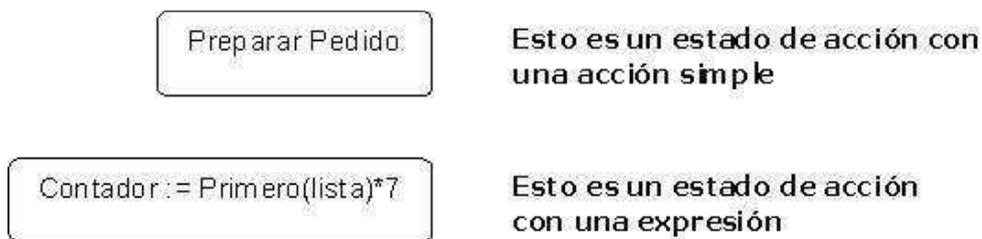
2.6.6 Diagrama de Actividades

Un diagrama de actividades es un caso especial de un diagrama de estados en el cual casi todos los estados son estados de acción (identifican que acción

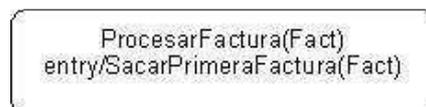
se ejecuta al estar en él) y casi todas las transiciones son enviadas al terminar la acción ejecutada en el estado anterior. Puede dar detalle a un caso de uso, un objeto o un mensaje en un objeto. Sirven para representar transiciones internas, sin hacer mucho énfasis en transiciones o eventos externos.

2.6.6.1 Estados de Actividad y Estados de Acción

La representación de ambos es un rectángulo con las puntas redondeadas, en cuyo interior se representa bien una actividad o bien una acción. La forma de expresar tanto una actividad como una acción, no queda impuesta por UML, se podría utilizar lenguaje natural, una especificación formal de expresiones, un metalenguaje, etc. La idea central es la siguiente: "Un estado que represente una acción es atómico, lo que significa que su ejecución se puede considerar instantánea y no puede ser interrumpida" En la Figura, podemos ver ejemplos de estados de acción



En cambio un estado de actividad, sí puede descomponerse en más sub-actividades representadas a través de otros diagramas de actividades. Además estos estados sí pueden ser interrumpidos y tardan un cierto tiempo en completarse. En los estados de actividad podemos encontrar otros elementos adicionales como son: acciones de entrada (entry) y de salida (exit) del estado en cuestión.



2.6.6.2 Transiciones

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



2.6.6.3 Bifurcaciones

Un flujo de control no tiene porqué ser siempre secuencial, puede presentar caminos alternativos. Para poder representar dichos caminos alternativos o bifurcación se utilizará como símbolo el rombo. Dicha bifurcación tendrá una transición de entrada y dos o más de salida. En cada transición de salida se colocará una expresión booleana que será evaluada una vez al llegar a la bifurcación, las guardas de la bifurcación han de ser excluyentes y contemplar todos los casos ya que de otro modo la ejecución del flujo de control quedaría interrumpida. Para poder cubrir todas las posibilidades se puede utilizar la palabra ELSE, para indicar una transición obligada a un determinado estado cuando el resto de guardas han fallado.

2.6.6.4 División y Unión

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

2.6.6.5 Calles

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

CAPITULO III

ASPECTOS METODOLOGICOS

3.1 CONCLUSIONES

ColdFusion es un servidor de aplicaciones Web moderno que permite desarrollar páginas complejas en un tiempo realmente corto, ya que tiene un administrador de base de datos propio, que facilita la conexión a las mismas, de esta forma se las puede invocar desde cualquier página con extensión .cfm o .cfc y realizar consultas directamente a las bases de datos.

El servidor de Colfusion es una excelente opción debido a que además de trabajar con formatos html para la interfaz de usuario, permite desarrollar páginas con formato Flash mediante la tecnología Flash Remoting, que permite utilizar lenguajes JavaScript y ActionScript, evitándonos realizar los eventos de Submit en un botón y a la vez recargar la página.

La Metodología OOHDM permitió una adecuada documentación del proceso, además de facilitar el desarrollo del proyecto y llevar un control en las pruebas de funcionamiento, lo cual incide en la calidad del producto de Software positivamente.

El manejo de perfiles de usuario y claves personales para la seguridad del sitio y el acceso a la información, fue de gran ayuda porque permitió dar un seguimiento a la información ingresada y modificada en la base de datos, y a la vez brinda eficaz seguridad a la información.

3.2 RECOMENDACIONES

Es necesario realizar pruebas de funcionamiento con los usuarios que posteriormente utilizarán la aplicación, para descubrir las falencias del aplicativo y

de esta manera construir un software amigable con el usuario, a medida que se avanza en el desarrollo.

Se recomienda para el desarrollo del proyecto configurar dos sitios en el que se alojará un ambiente de desarrollo y uno de producción, para en el ambiente de desarrollo hacer todas las pruebas de funcionalidad que luego se subirán al ambiente de producción en la cual se trabajará ya con datos reales.

Se necesita una excelente capacitación al personal que manejará el aplicativo para su buen uso ya que ellos serán los responsables del buen funcionamiento del mismo.

3.3 BIBLIOGRAFÍA

3.3.1 Referencias Web

3.3.1.1 Paradigmas

<http://lsi.ugr.es/~ig1/docis/espinal.pdf>

<http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernosetaaprevia/CORCOS-ESPIRAL.pdf>

<http://www.ra-ma.es/cf/html/catalogo/libros/down/adaigGS.pdf>

<http://kybele.escet.urjc.es/documentos/IS2/IS2-1v4.pdf>

<http://www.inf.udec.cl/~mvaras/gpis/apunteGPDS.pdf>

<http://trevinca.ei.uvigo.es/~ebalonso/asignaturas/esx/guiones/esxClase4.pdf>

<http://www.biblioteca.co.cr/pdf/unidad12-4.pdf>

<http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20dise%F1o%20orientado%20a%20objetos/rumbaugh.pdf>

<http://ccc.inaoep.mx/~labvision/doo/proy/T32.pdf>

http://exa.unne.edu.ar/depar/areas/informatica/anasistem1/public_html/Temas/Temas_08.pdf

<http://148.202.148.5/cursos/cc321/fundamentos/unidad1/espiral.htm>

3.3.1.2 Metodología

<http://www.itlalaguna.edu.mx/academico/carreras/sistemas/Analisis%20y%20di-se%F1o%20orientado%20a%20objetos/rumbaugh.pdf>

http://www.emagister.com/tutorial/frame.cfm?id_centro=35596070040551554868665657484551&id_curso=32700070040551665352555170674570&url_frame=http://www.ilustrados.com/publicaciones/EpZVVyAkyuqpxfFpAs.php

<http://www.monografias.com/trabajos13/metomt/metomt.shtml>

http://exa.unne.edu.ar/depar/areas/informatica/anasistem1/public_html/Temas/Temas_08.pdf

http://exa.unne.edu.ar/depar/areas/informatica/anasistem1/public_html/Temas/Temas_08.pdf

3.3.1.3 UML

<http://www.clikear.com/manuales/uml/introduccion.aspx>

http://www.cs.ualberta.ca/~pfiguero/soo/uml/estr_estatica01.html

http://www.cs.ualberta.ca/~pfiguero/soo/uml/casos_uso01.html

<http://www.cs.ualberta.ca/~pfiguero/soo/uml/secuencia01.html>

<http://www.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>

<http://www.cs.ualberta.ca/~pfiguero/soo/uml/estados01.html>

<http://www.cs.ualberta.ca/~pfiguero/soo/uml/actividades01.html>

http://www.cs.ualberta.ca/~pfiguero/soo/uml/estr_estatica02.html

3.3.1.4 Java

<http://www.sc.ehu.es/sbweb/fisica/cursoJava/Intro.htm>

<http://www.google.com.ec/search?hl=es&ei=zKeySfrWENCctwflzKy8Bw&sa=X&oi=spell&resnum=0&ct=result&cd=1&q=tutorial+POO+con+java&spell=1>

<http://java.sun.com/javase/6/docs/api/>

MANUAL TÉCNICO

ANÁLISIS

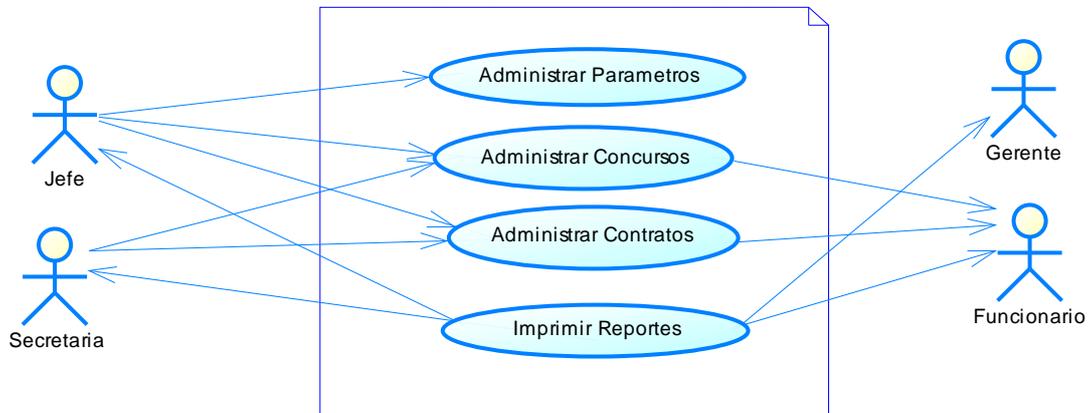
DISEÑO CONCEPTUAL

SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 DESCRIPCIÓN DE ACTORES

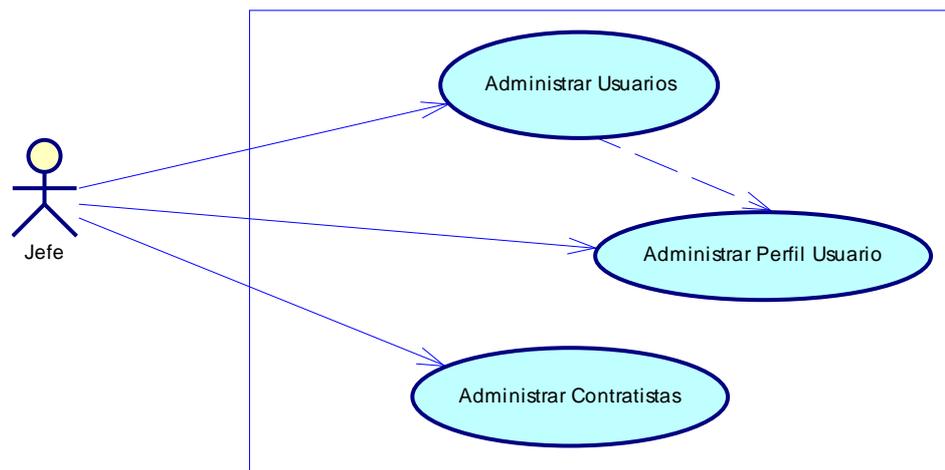
ACTOR	DESCRIPCIÓN
 Jefe	<p><u>Jefe</u></p> <p>Es la entidad que representa al gerente de cada gerencia de la empresa PETROECUADOR que interactúa con el Sistema Gestión de Contratos, actualizando y generando reportes de la información. Esta entidad asume los siguientes grupos de trabajo:</p> <ul style="list-style-type: none"> ➤ Usuario Administrador: tiene control total en el sistema para su Gerencia.
 Funcionario	<p><u>Funcionario</u></p> <p>Es la entidad que representa al contratante de servicios, a la que se le entrega los documentos de avance físico y económico de cada obra, el historial de sus controles periódicos, una gráfica evolutiva; registrados en el Sistema de Gestión de Contratos a través de archivos impresos o digitales. Esta entidad asume los siguientes grupos de trabajo:</p> <ul style="list-style-type: none"> ➤ Usuario Sofisticado: tiene control parcial del sistema, e interactúa actualizando y realizando seguimientos físicos y económicos de las obras.
 Secretaria	<p><u>Secretaria</u></p> <p>Es la entidad que representa al digitador encargado de ingresar los contratos respectivos a la gerencia que pertenece, a las que se les entrega informes de avances de las obras, consultas realizadas dentro de un periodo determinado; a través de archivos impresos o digitales. Esta entidad asume los siguientes grupos de trabajo:</p> <ul style="list-style-type: none"> ➤ Usuario Normal: Interactúa con el sistema ingresando información de contratos y generando reportes para su Gerencia.
 Gerente	<p><u>Gerente</u></p> <p>Es la entidad representante legal de la empresa PETROECUADOR, a la que se le entrega reportes de la información de todos los servicios y avances respectivos que tiene la empresa y vigencia de los contratos; a través de archivos impresos o digitales. Esta entidad asume los siguientes grupos de trabajo:</p> <ul style="list-style-type: none"> ➤ Usuario Especial: Interactúa con el sistema generando reportes de los servicios prestados de toda la empresa.

DIAGRAMAS DE CASOS DE USO

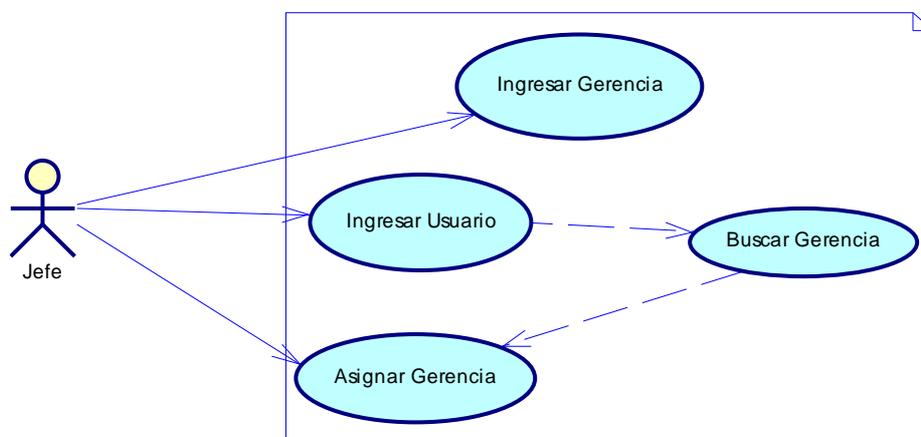
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
DIAGRAMA MODULAR DE CASOS DE USO**



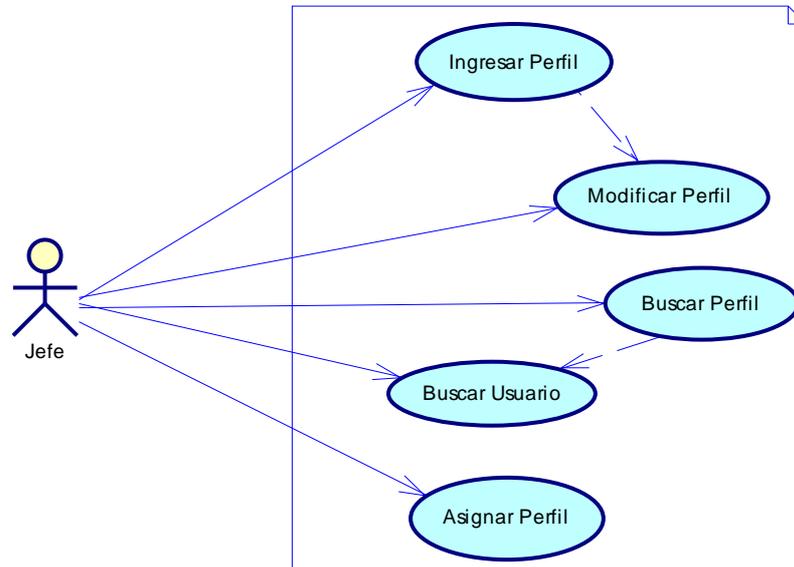
**CASOS DE USO GENERALES
ADMINISTRAR PARÁMETROS**



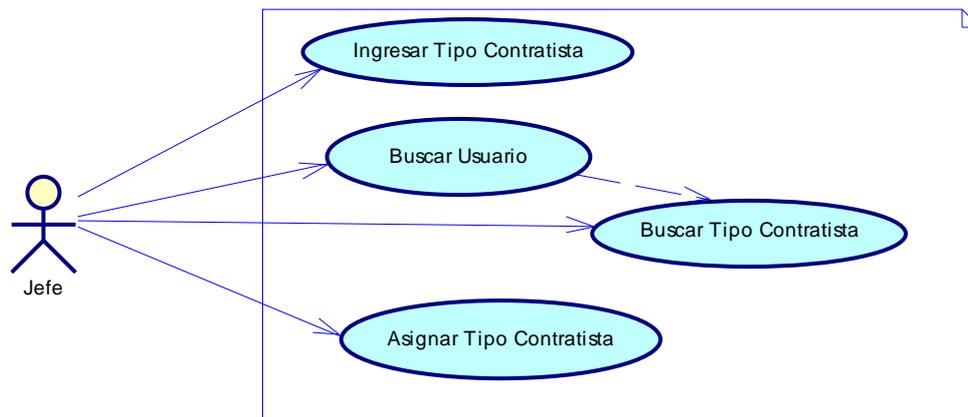
ADMINISTRAR PARÁMETROS - ADMINISTRAR USUARIOS



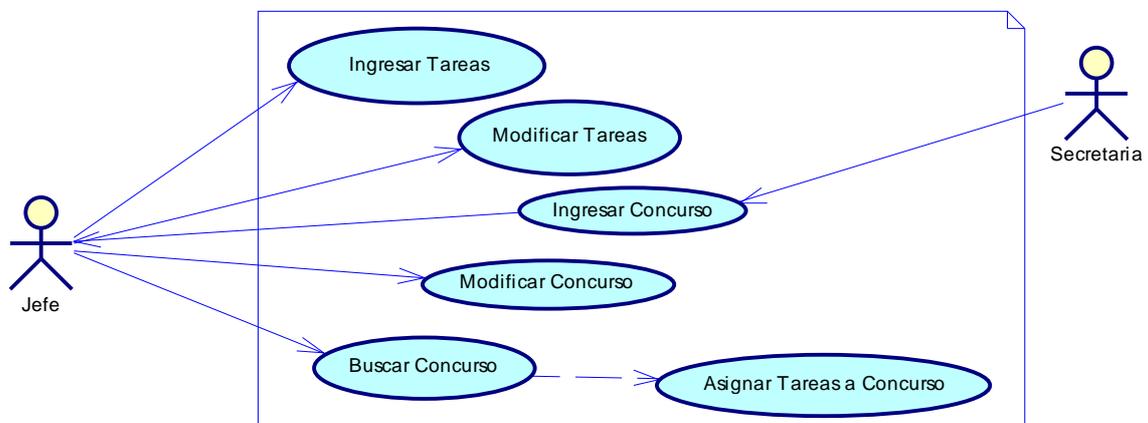
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR PARÁMETROS - ADMINISTRAR PERFIL USUARIO**



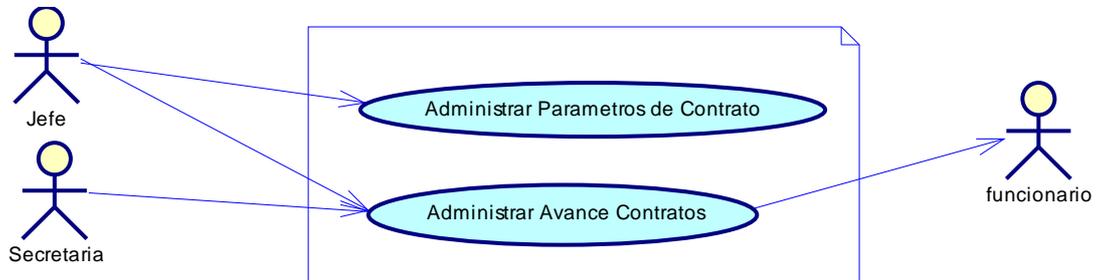
ADMINISTRAR PARÁMETROS - ADMINISTRAR CONTRATISTAS



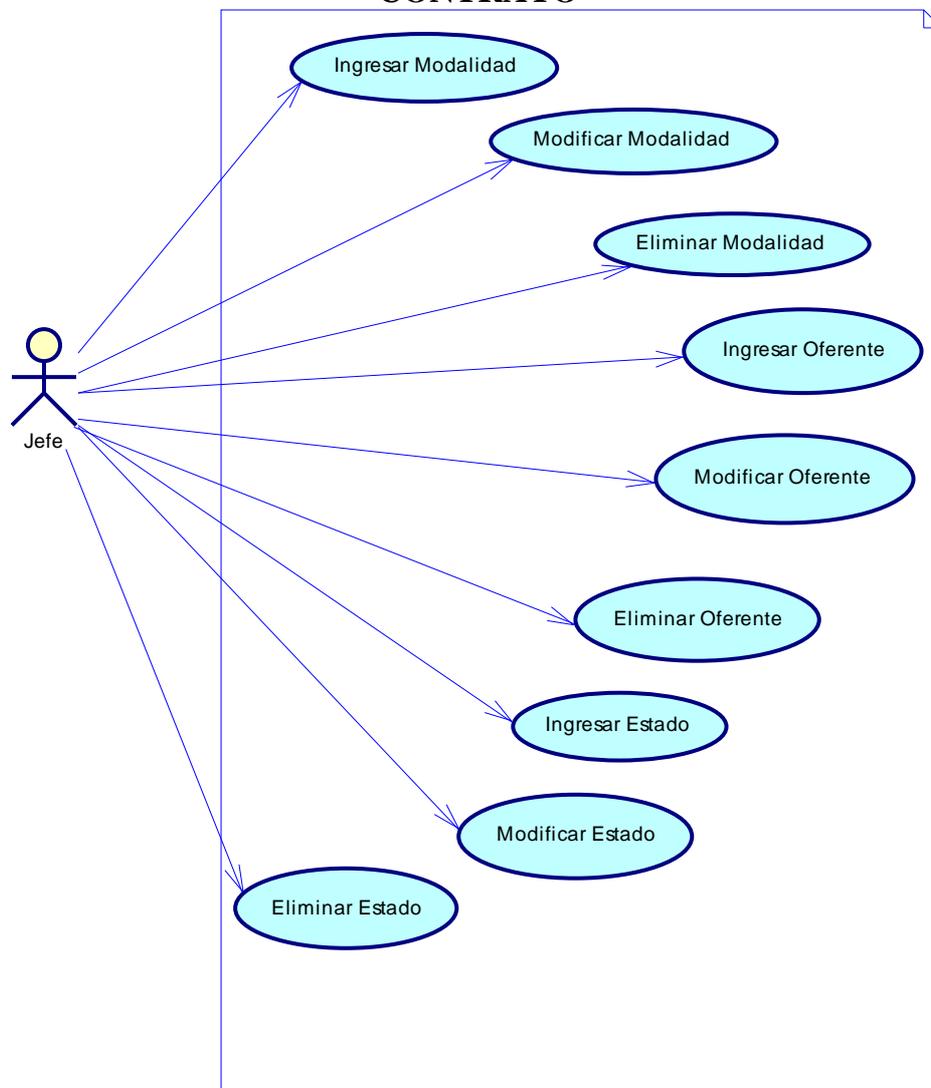
ADMINISTRAR CONCURSOS



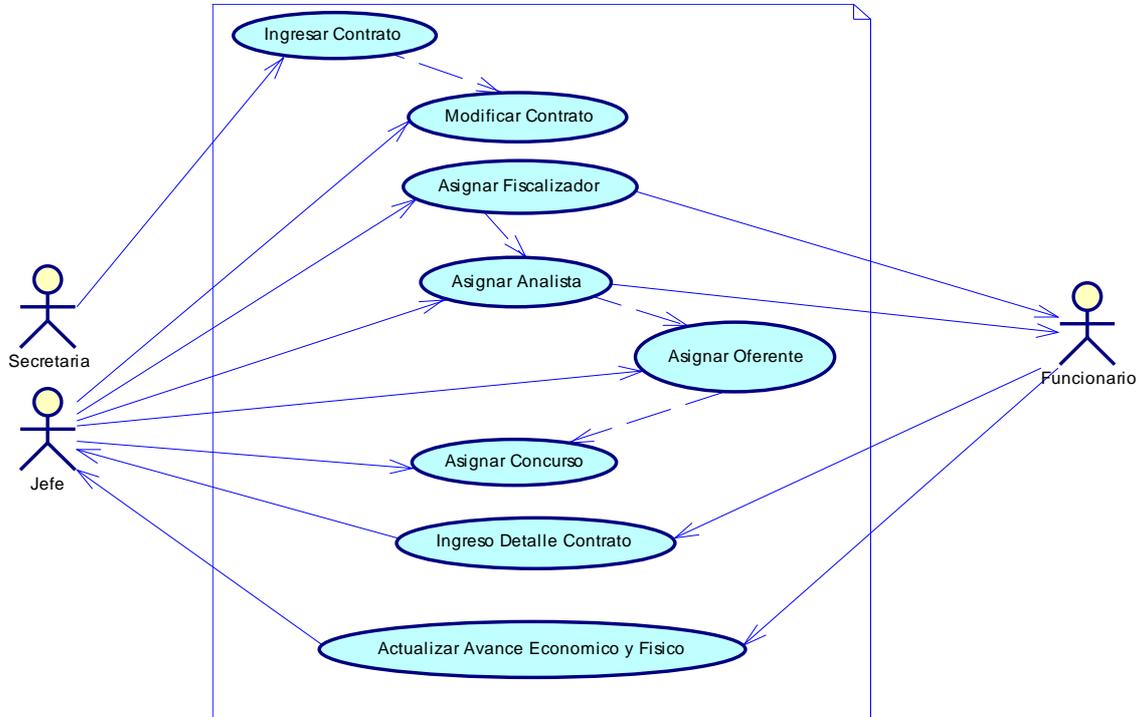
SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR CONTRATOS



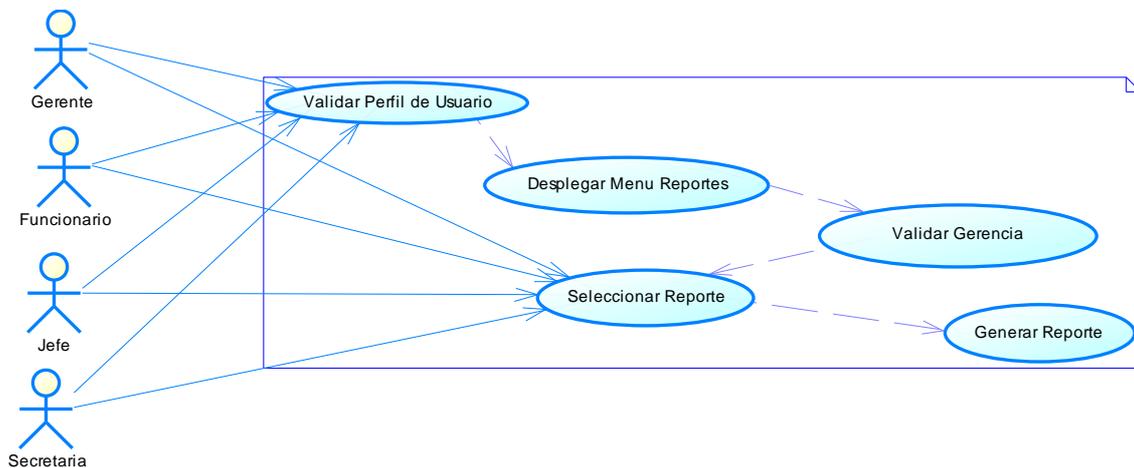
ADMINISTRAR CONTRATOS - ADMINISTRAR PARAMETROS DE CONTRATO



SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR CONTRATOS - ADMINISTRAR AVANCE CONTRATOS



IMPRIMIR REPORTES



DESCRIPCIÓN DE CASOS DE USO

**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
DESCRIPCIÓN DE CASOS DE USO**

CASOS DE USO MODULARES	
CASO DE USO	DESCRIPCIÓN
Administrar Parámetros	Permite administrar la información de contratistas, usuarios, perfiles de usuarios para acceso al menú.
Administrar Concursos	Permite administrar la información de concursos y seguimiento de concursos mediante el ingreso de tareas.
Administrar Contratos	Permite administrar la información de los contratos, modalidad del contrato, estado, avances económicos, avances físicos, contratistas y fiscalizadores.
Imprimir Reportes	Permite exportar información de la gestión realizada de los concursos y contratos.

CASOS DE USO GENERALES

CASOS DE USO DE ADMINISTRAR PARÁMETROS	
CASO DE USO	DESCRIPCIÓN
Administrar Usuarios	Permite administrar la información de los datos del usuario y gerencias.
Administrar Perfiles de Usuario	Permite administrar la información de perfiles de usuarios y control de acceso al menú del sistema. Incluye añadir, consultar, modificar, eliminar.
Administrar Contratistas	Permite administrar la información de contratistas.

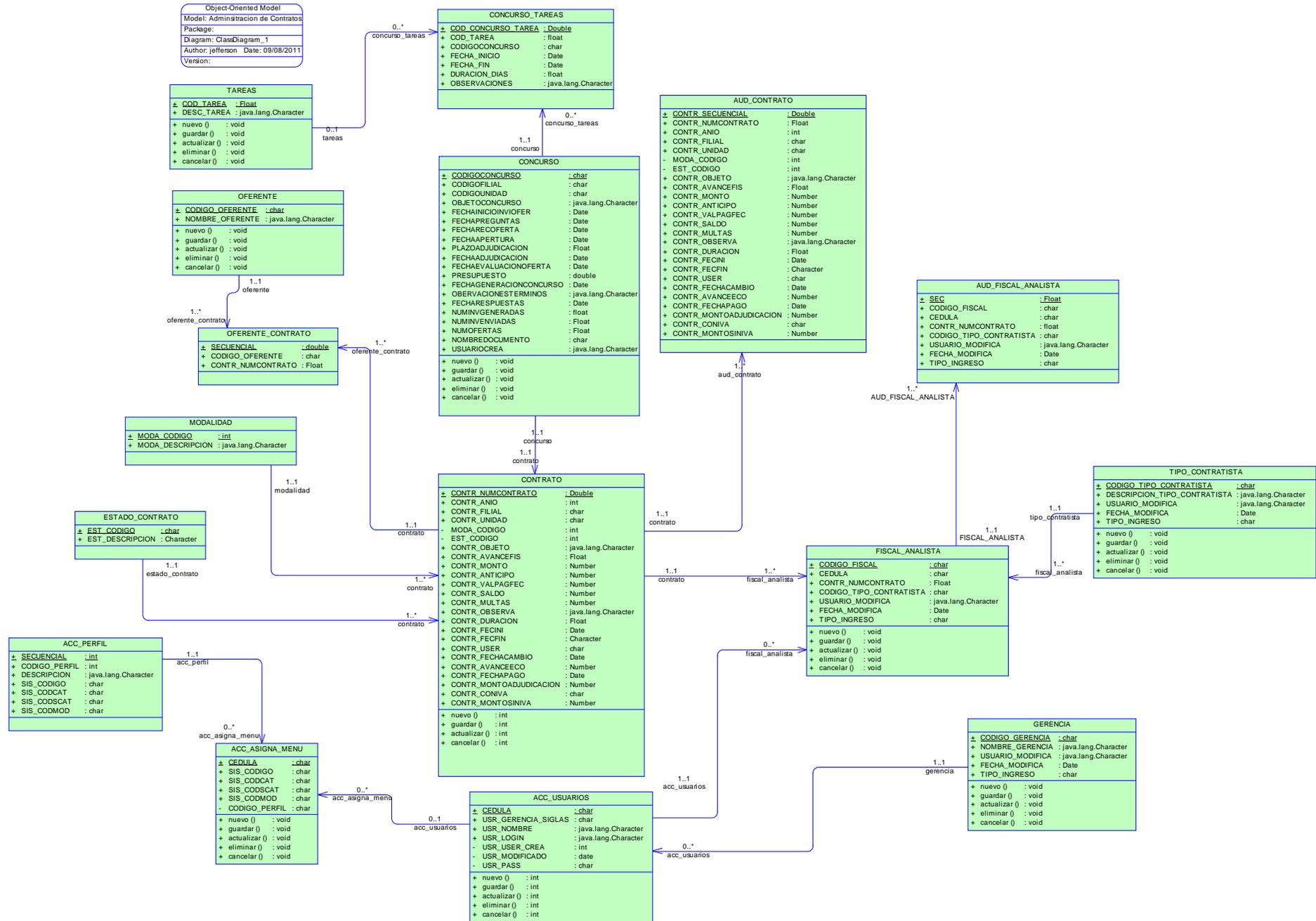
CASOS DE USO DE ADMINISTRAR CONCURSOS	
CASO DE USO	DESCRIPCIÓN
Administrar Concurso	Permite administrar la información de los concursos, y realizar seguimientos mediante el ingreso de tareas. Incluye consultar, modificar, eliminar.

CASOS DE USO DE ADMINISTRAR CONTRATOS	
CASO DE USO	DESCRIPCIÓN
Administrar Parámetros de Contrato	Permite administrar la información de la modalidad de contratación y estado del contrato. Además permite el registro de los oferentes adjudicados para cada contrato.
Administrar Avance Contratos	Permite administrar la información del contrato y dar un seguimiento de avance económico y físico del mismo.

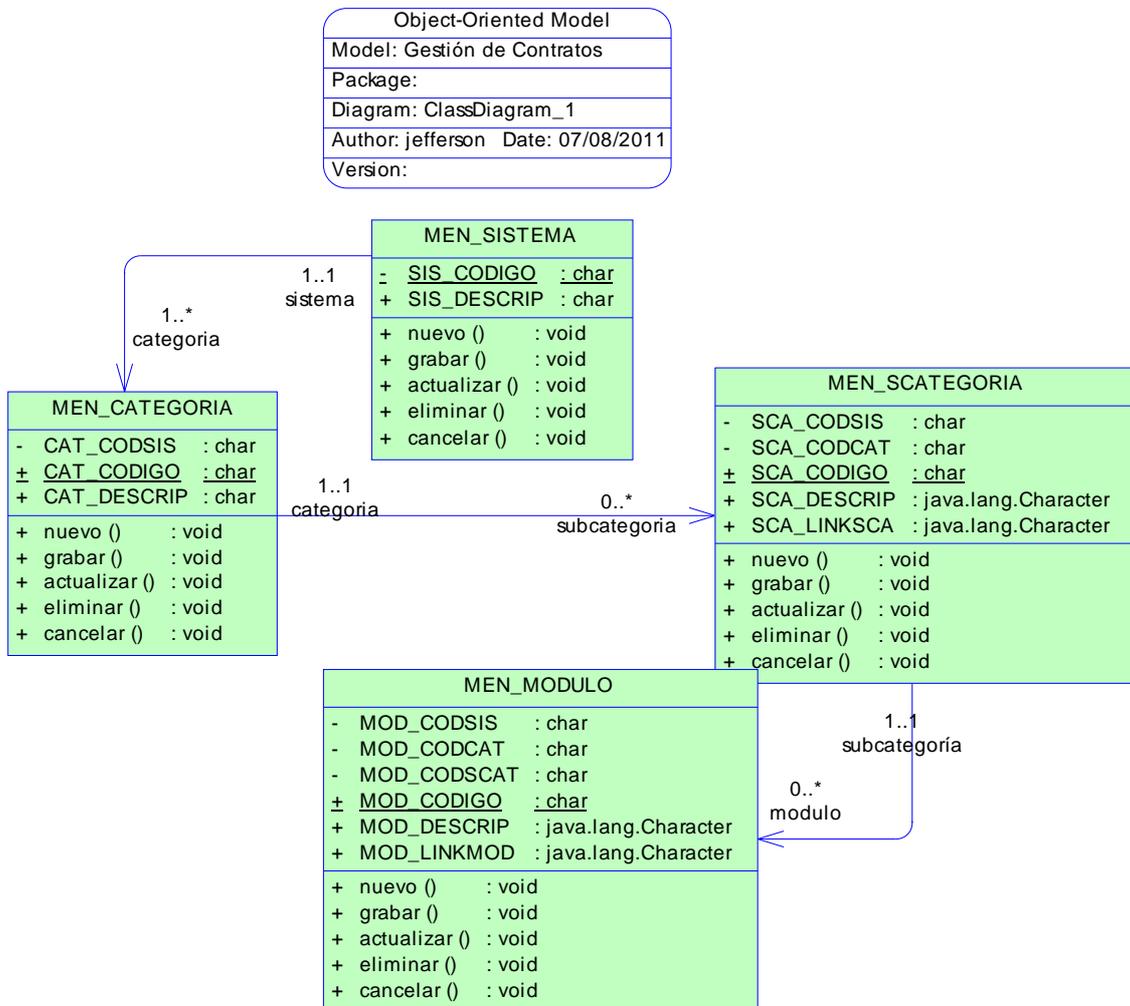
CASOS DE USO DE IMPRIMIR REPORTES	
CASO DE USO	DESCRIPCIÓN
Imprimir Reportes	Permite acceder a la información de usuarios, contratos, y concursos mediante criterios de búsqueda e imprimirlos en formato pdf.

DIAGRAMAS DE CLASES

DIAGRAMA DE CLASES DEL SISTEMA GESTIÓN DE CONTRATOS



SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 DIAGRAMA DE CLASES DEL SISTEMA ADMINISTRAR MODULOS



DICCIONARIO DE CLASES

Clase: TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
COD_TAREA	Float	Código de tarea	TAREAS
DESC_TAREA	varchar	Nombre de la tarea	TAREAS
Clase: CONCURSO_TAREAS			
COD_CONCURSO_TAREA	Double	secuencial	CONCURSO_TAREAS
COD_TAREA	float	Código tarea	CONCURSO_TAREAS
CODIGOCONCURSO	char	Código concurso	CONCURSO_TAREAS
FECHA_INICIO	Date	Fecha inicio de la tarea	CONCURSO_TAREAS
FECHA_FIN	Date	Fecha fin de la tarea	CONCURSO_TAREAS
DURACION_DIAS	float	Duración de la tarea en días	CONCURSO_TAREAS
OBSERVACIONES	varchar	Observaciones de la tarea	CONCURSO_TAREAS
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CODIGOCONCURSO	char	Código concurso	CONCURSO
CODIGOFILIAL	char	Código filial	CONCURSO
CODIGOUNIDAD	char	Código Gerencia	CONCURSO
OBJETOCONCURSO	varchar	Objeto del concurso	CONCURSO
FECHAINICIOINVIOFER	Date	Fecha inicio de invitaciones	CONCURSO
FECHAPREGUNTAS	Date	Fecha hasta de preguntas	CONCURSO
FECHARECOFERTA	Date	Fecha recolección de ofertas	CONCURSO
FECHAAPERTURA	Date	Fecha Apertura de sobres	CONCURSO
PLAZOADJUDICACION	Float	Plazo adjudicación de oferentes	CONCURSO
FECHAADJUDICACION	Date	Fecha de la adjudicación oferente	CONCURSO
FECHAEVALUACIONOFERTA	Date	Fecha evaluación al oferente	CONCURSO
PRESUPUESTO	double	Presupuesto del concurso	CONCURSO
FECHAGENERACIONCONCURSO	Date	Fecha de ingreso del concurso	CONCURSO
OBERVACIONESTERMINOS	varchar	Observaciones o términos de referencia	CONCURSO
FECHARESPUESTAS	Date	Fecha límite respuestas	CONCURSO
NUMINVGENERADAS	float	No. de invitaciones generadas	CONCURSO
NUMINVENVIADAS	Float	No. de invitaciones enviadas	CONCURSO
NUMOFERTAS	Float	Ofertas recibidas	CONCURSO

NOMBREDOCUMENTO	char	Nombre del documento concurso	CONCURSO
USUARIOCREA	varchar	Usuario crea el concurso	CONCURSO
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CONTR_NUMCONTRATO	Double	Número de contrato	CONTRATO
CONTR_ANIO	int	Año ingreso del contrato	CONTRATO
CONTR_FILIAL	char	Filial	CONTRATO
CONTR_UNIDAD	char	Gerencia	CONTRATO
MODA_CODIGO	int	Modalidad del contrato	CONTRATO
EST_CODIGO	int	Estado del contrato	CONTRATO
CONTR_OBJETO	varchar	Objeto del contrato	CONTRATO
CONTR_AVANCEFIS	Float	Avance físico	CONTRATO
CONTR_MONTO	Number	Monto	CONTRATO
CONTR_ANTICIPO	Number	Anticipo	CONTRATO
CONTR_VALPAGFEC	Number	Valor pagado a la fecha	CONTRATO
CONTR_SALDO	Number	Saldo	CONTRATO
CONTR_MULTAS	Number	Multas	CONTRATO
CONTR_OBSERVA	varchar	Observaciones	CONTRATO
CONTR_DURACION	Float	Duración en días	CONTRATO
CONTR_FECINI	Date	Fecha inicio contrato	CONTRATO
CONTR_FECFIN	Date	Fecha fin contrato	CONTRATO
CONTR_USER	char	Usuario crea contrato	CONTRATO
CONTR_FECHACAMBIO	Date	Fecha modificaciones	CONTRATO
CONTR_AVANCEECO	Number	Avance económico	CONTRATO
CONTR_FECHAPAGO	Date	Fecha pago	CONTRATO
CONTR_MONTOADJUDICACION	Number	Monto adjudicado	CONTRATO
CONTR_CONIVA	char	Monto con IVA	CONTRATO
CONTR_MONTOSINIVA	Number	Monto sin IVA	CONTRATO
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CODIGO_FISCAL	char	Código fiscal	FISCAL_ANALISTA
CEDULA	char	Cedula fiscal	FISCAL_ANALISTA
CONTR_NUMCONTRATO	Float	Número contrato	FISCAL_ANALISTA
CODIGO_TIPO_CONTRATISTA	char	Tipo contratista	FISCAL_ANALISTA
USUARIO_MODIFICA	varchar	Usuario modifica	FISCAL_ANALISTA

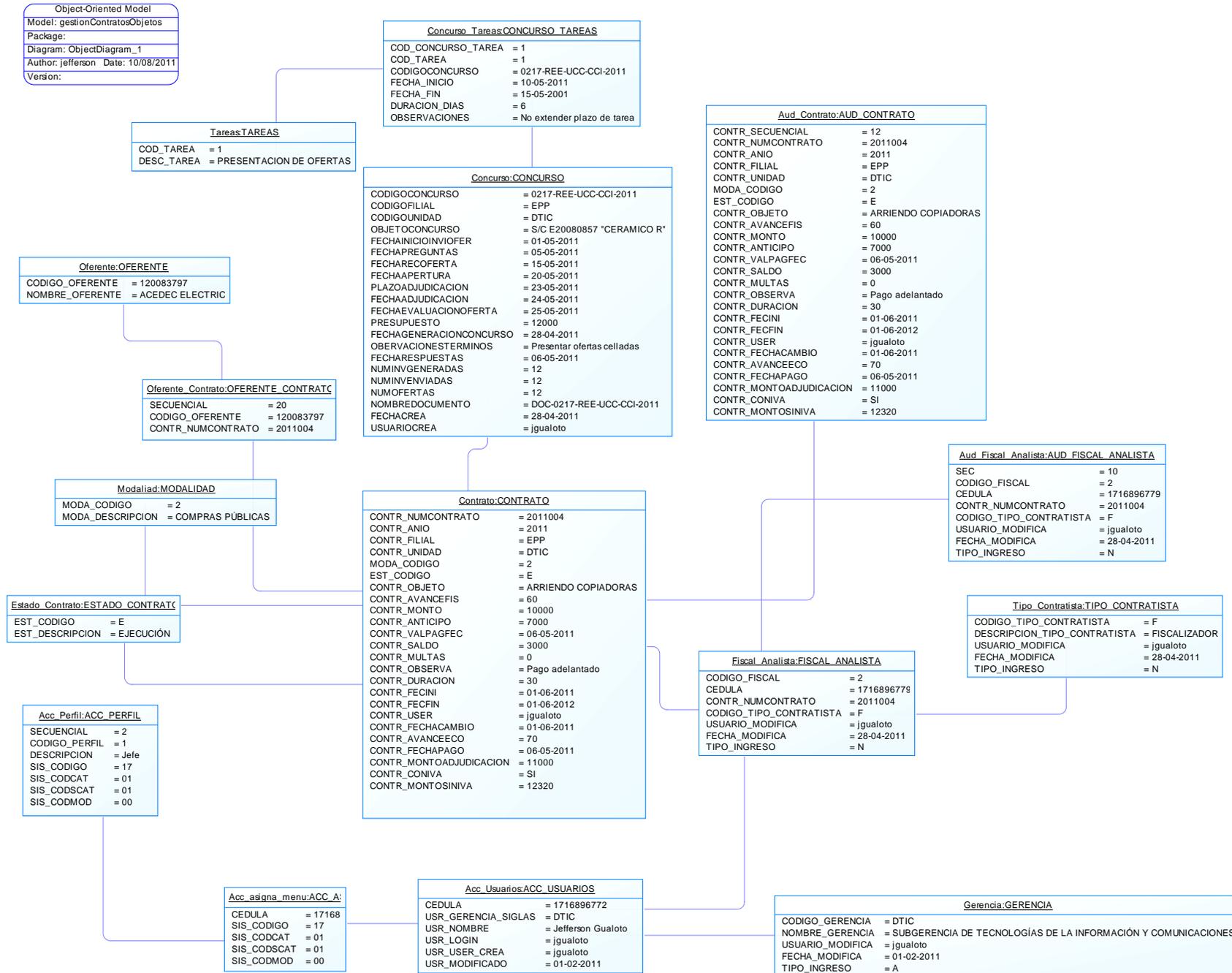
FECHA_MODIFICA	Date	Fecha modificación	FISCAL_ANALISTA
TIPO_INGRESO	char	Tipo de actualización	FISCAL_ANALISTA
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CODIGO_OFERENTE	char	Código del oferente	OFERENTE
NOMBRE_OFERENTE	varchar	Nombre del oferente	OFERENTE
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CODIGO_TIPO_CONTRATISTA	char	Código del Contratista	TIPO_CONTRATISTA
DESCRIPCION_TIPO_CONTRATISTA	varchar	Descripción tipo contratista	TIPO_CONTRATISTA
USUARIO_MODIFICA	varchar	Usuario modifica	TIPO_CONTRATISTA
FECHA_MODIFICA	Date	Fecha modificación	TIPO_CONTRATISTA
TIPO_INGRESO	char	Tipo de ingreso (Nuevo, actualiza, elimina)	TIPO_CONTRATISTA
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CODIGO_GERENCIA	char	Código de gerencia	GERENCIA
NOMBRE_GERENCIA	varchar	Nombre de gerencia	GERENCIA
USUARIO_MODIFICA	varchar	Usuario modifica gerencia	GERENCIA
FECHA_MODIFICA	Date	Fecha de modificación	GERENCIA
TIPO_INGRESO	char	Tipo de ingreso (Nuevo, actualiza, elimina)	GERENCIA
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CONTR_SECUENCIAL	Double	Código secuencial	AUD_CONTRATO
CONTR_NUMCONTRATO	Float	Número contrato	AUD_CONTRATO
CONTR_ANIO	int	Año del contrato	AUD_CONTRATO
CONTR_FILIAL	char	Filial	AUD_CONTRATO
CONTR_UNIDAD	char	Gerencia	AUD_CONTRATO
MODA_CODIGO	int	Modalidad del contrato	AUD_CONTRATO
EST_CODIGO	int	Estado del contrato	AUD_CONTRATO
CONTR_OBJETO	varchar	Objeto	AUD_CONTRATO
CONTR_AVANCEFIS	Float	Avance físico	AUD_CONTRATO
CONTR_MONTO	Number	Monto	AUD_CONTRATO
CONTR_ANTICIPO	Number	Anticipo	AUD_CONTRATO
CONTR_VALPAGFEC	Number	Valor pagado a la fecha	AUD_CONTRATO
CONTR_SALDO	Number	Saldo	AUD_CONTRATO

CONTR_MULTAS	Number	Multas	AUD_CONTRATO
CONTR_OBSERVA	varchar	Observaciones	AUD_CONTRATO
CONTR_DURACION	Float	Duración en días	AUD_CONTRATO
CONTR_FECINI	Date	Fecha inicio contrato	AUD_CONTRATO
CONTR_FECFIN	Character	Fecha fin contrato	AUD_CONTRATO
CONTR_USER	char	Usuario ingresa contrato	AUD_CONTRATO
CONTR_FECHACAMBIO	Date	Fecha ingreso contrato	AUD_CONTRATO
CONTR_AVANCEECO	Number	Avance económico	AUD_CONTRATO
CONTR_FECHAPAGO	Date	Fecha Pago	AUD_CONTRATO
CONTR_MONTOADJUDICACION	Number	Monto adjudicación	AUD_CONTRATO
CONTR_CONIVA	char	Almacena S si es con IVA N si no	AUD_CONTRATO
CONTR_MONTOSINIVA	Number	Monto sin IVA	AUD_CONTRATO
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
MODA_CODIGO	int	Código modalidad	MODALIDAD
MODA_DESCRIPCION	varchar	Descripción modalidad	MODALIDAD
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
EST_CODIGO	char	Código Estado	ESTADO_CONTRATO
EST_DESCRIPCION	Character	Descripción estado	ESTADO_CONTRATO
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
SEC	Float	Código secuencial	AUD_FISCAL_ANALISTA
CODIGO_FISCAL	char	Código del analista o fiscal	AUD_FISCAL_ANALISTA
CEDULA	char	Cédula del analista o fiscal	AUD_FISCAL_ANALISTA
CONTR_NUMCONTRATO	float	Número de contrato	AUD_FISCAL_ANALISTA
CODIGO_TIPO_CONTRATISTA	char	Código del tipo de contratista	AUD_FISCAL_ANALISTA
USUARIO_MODIFICA	varchar	Usuario modifica	AUD_FISCAL_ANALISTA
FECHA_MODIFICA	Date	Fecha de modificación	AUD_FISCAL_ANALISTA
TIPO_INGRESO	char	Tipo de ingreso (Nuevo, actualiza, elimina)	AUD_FISCAL_ANALISTA
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
SECUENCIAL	double	Código secuencial	OFERENTE_CONTRATO
CODIGO_OFERENTE	char	Código oferente	OFERENTE_CONTRATO
CONTR_NUMCONTRATO	Float	Número de contrato	OFERENTE_CONTRATO

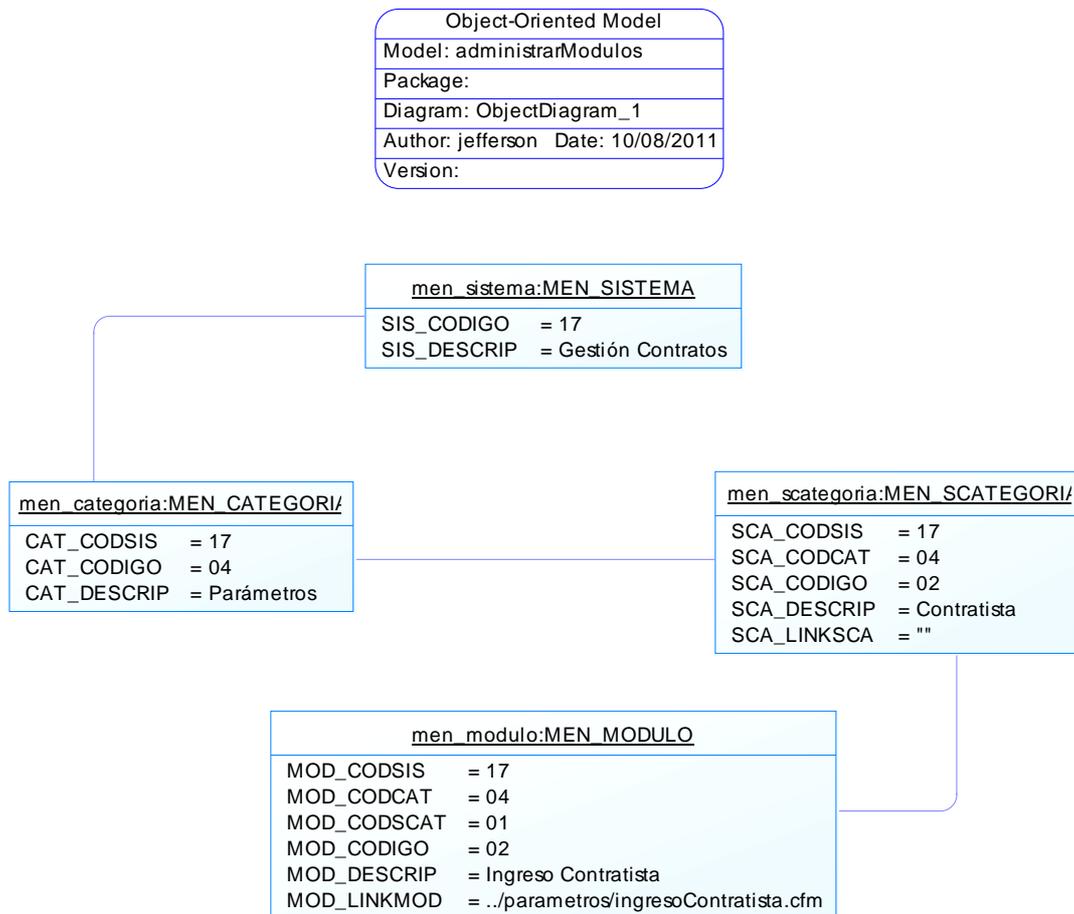
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CEDULA	char	Cédula del usuario	ACC_ASIGNA_MENU
SIS_CODIGO	char	Código sistema	ACC_ASIGNA_MENU
SIS_CODCAT	char	Código categoría	ACC_ASIGNA_MENU
SIS_CODSCAT	char	Código de subcategoría	ACC_ASIGNA_MENU
SIS_CODMOD	char	Código de módulo	ACC_ASIGNA_MENU
CODIGO_PERFIL	char	Código de perfil	ACC_ASIGNA_MENU
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
SECUENCIAL	int	Código secuencial	ACC_PERFIL
CODIGO_PERFIL	int	Código perfil	ACC_PERFIL
DESCRIPCION	varchar	Descripción de perfil	ACC_PERFIL
SIS_CODIGO	char	Código sistema	ACC_PERFIL
SIS_CODCAT	char	Código categoría	ACC_PERFIL
SIS_CODSCAT	char	Código de subcategoría	ACC_PERFIL
SIS_CODMOD	char	Código de módulo	ACC_PERFIL
Clase: CONCURSO_TAREAS			
ATRIBUTO	TIPO DE DATO	DESCRIPCIÓN	NOMBRE CLASE
CEDULA	char	Cedula	ACC_USUARIOS
USR_GERENCIA_SIGLAS	char	Código de Gerencia	ACC_USUARIOS
USR_NOMBRE	varchar	Nombre completo	ACC_USUARIOS
USR_LOGIN	varchar	Login	ACC_USUARIOS
USR_USER_CREA	int	Usuario crea	ACC_USUARIOS
USR_MODIFICADO	date	Fecha de creación de usuario	ACC_USUARIOS
USR_PASS	char	Clave del usuario	ACC_USUARIOS

DIAGRAMAS DE OBJETOS

DIAGRAMA DE OBJETOS DEL SISTEMA GESTIÓN DE CONTRATOS



SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 DIAGRAMA DE OBJETOS DEL SISTEMA GESTIÓN DE CONTRATOS ADMINISTRAR MODULOS



DISEÑO

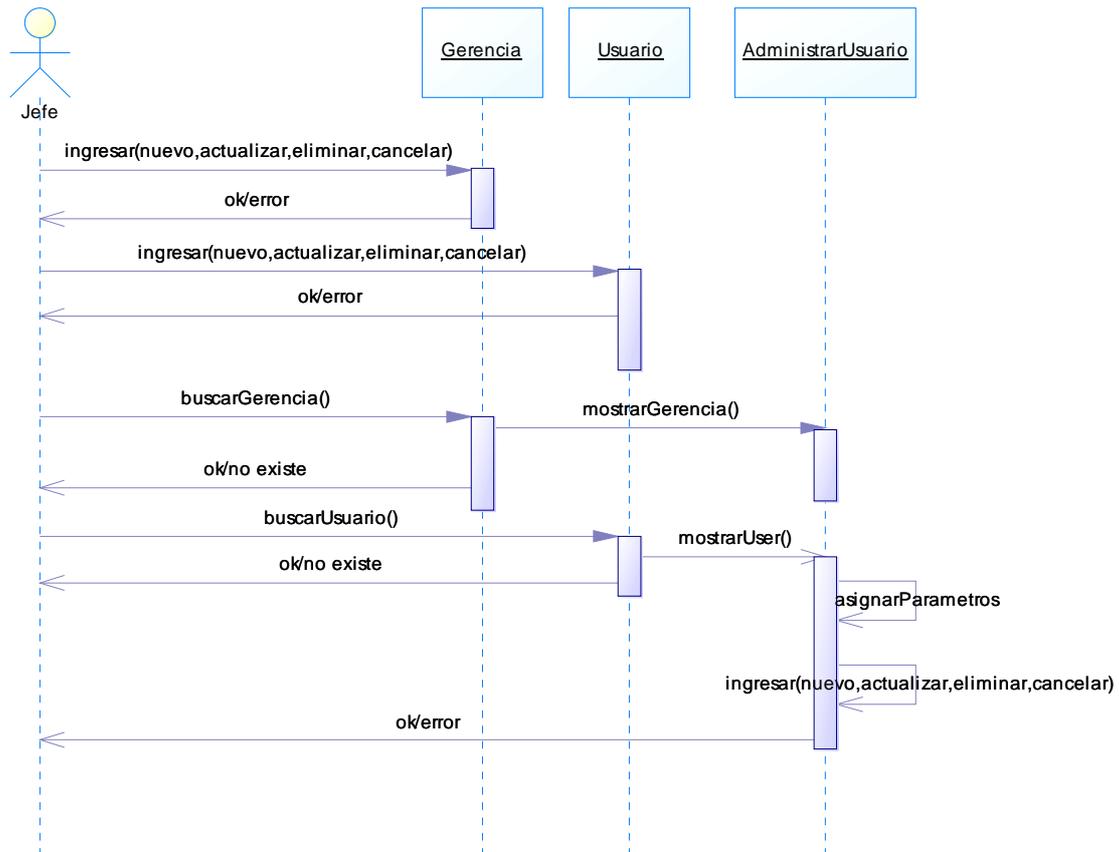
MODELO DINÁMICO

DIAGRAMAS DE INTERACCIÓN

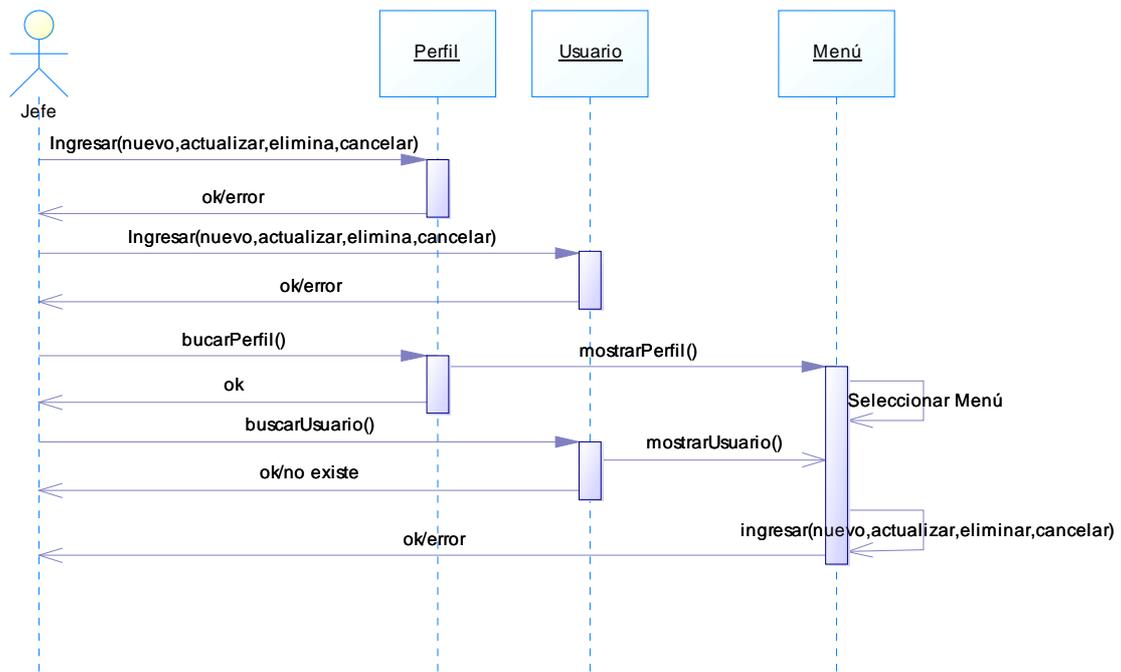
DIAGRAMAS DE SECUENCIA

SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR PARAMETROS

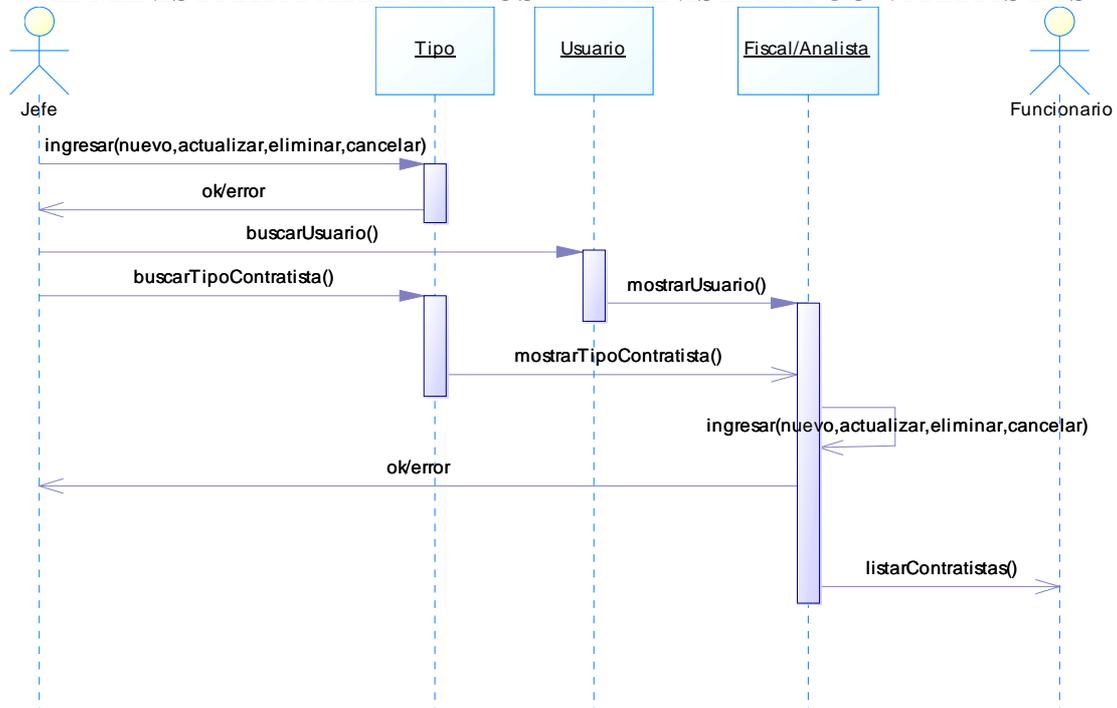
ADMINISTRAR PARÁMETROS - ADMINISTRAR USUARIOS



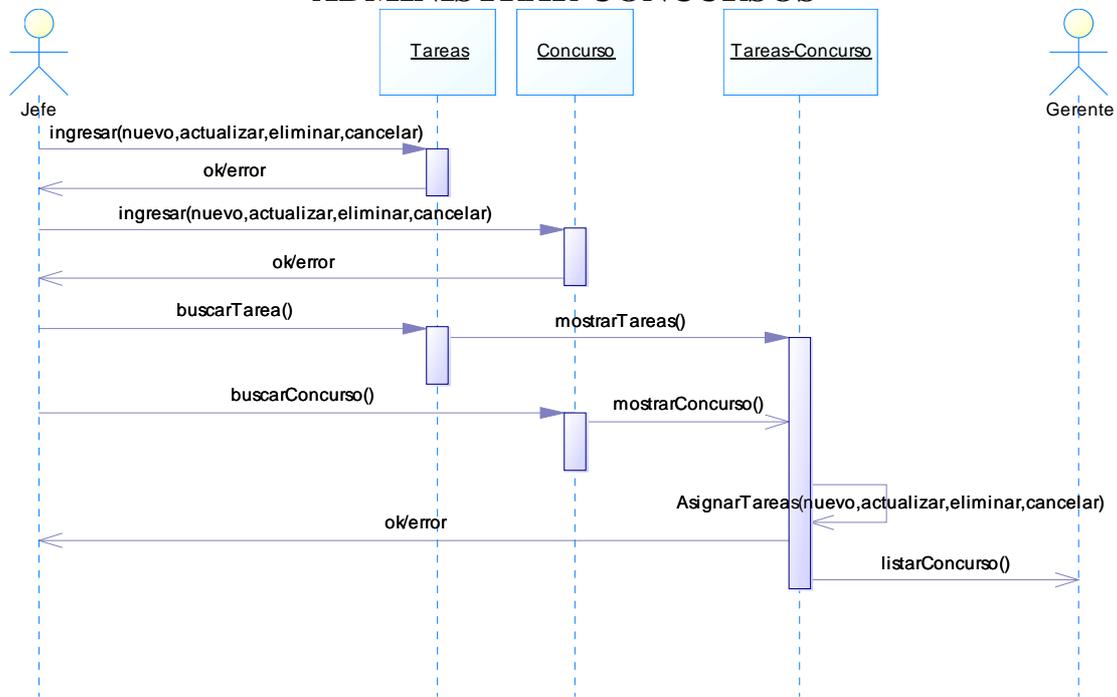
SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR PARAMETROS ADMINISTRAR PARÁMETROS - ADMINISTRAR PERFIL USUARIO



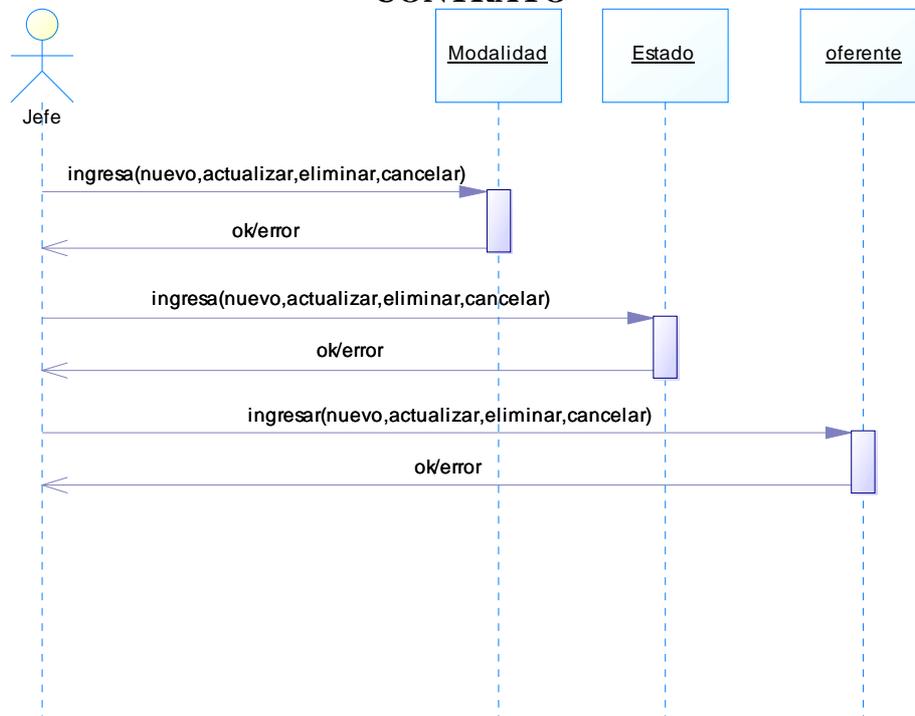
ADMINISTRAR PARÁMETROS - ADMINISTRAR CONTRATISTAS



SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR CONCURSOS

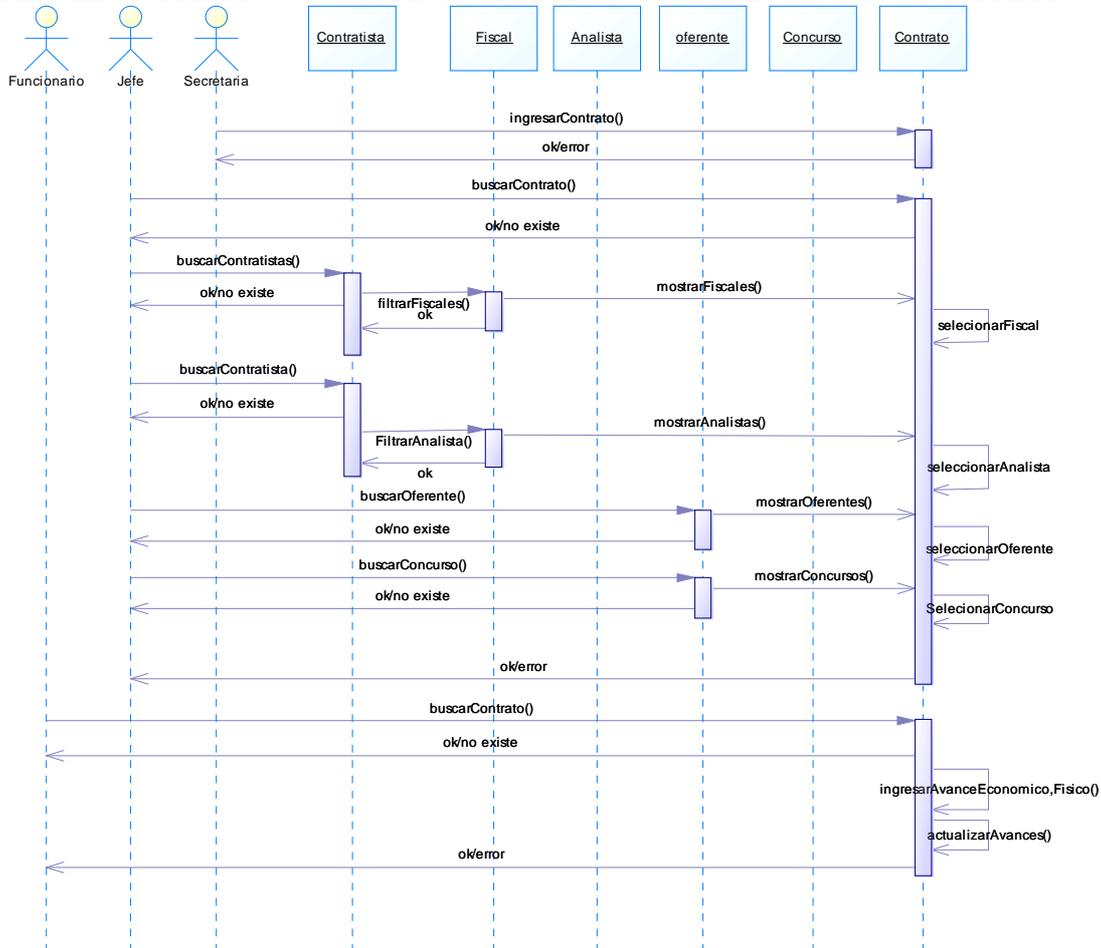


ADMINISTRAR CONTRATOS – ADMINISTRAR PARÁMETROS DE CONTRATO

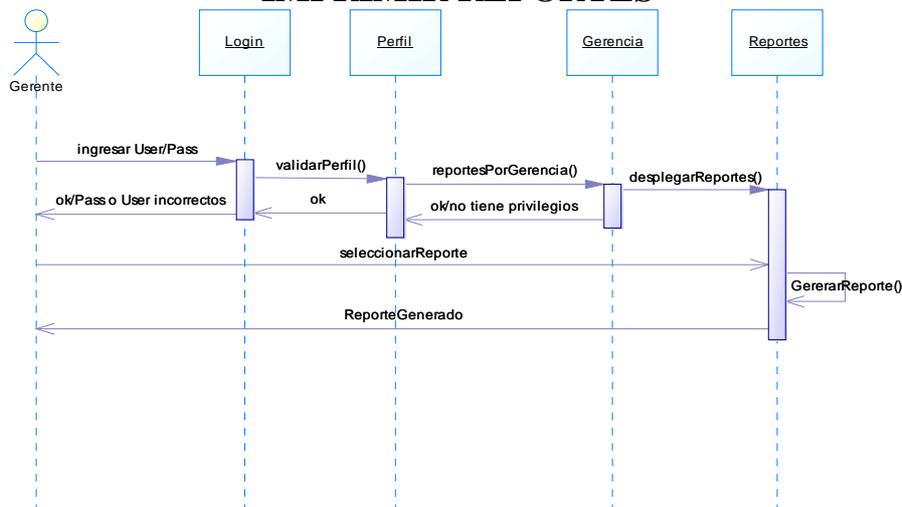


SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7

ADMINISTRAR CONTRATOS-ADMINISTRAR AVANCE CONTRATOS

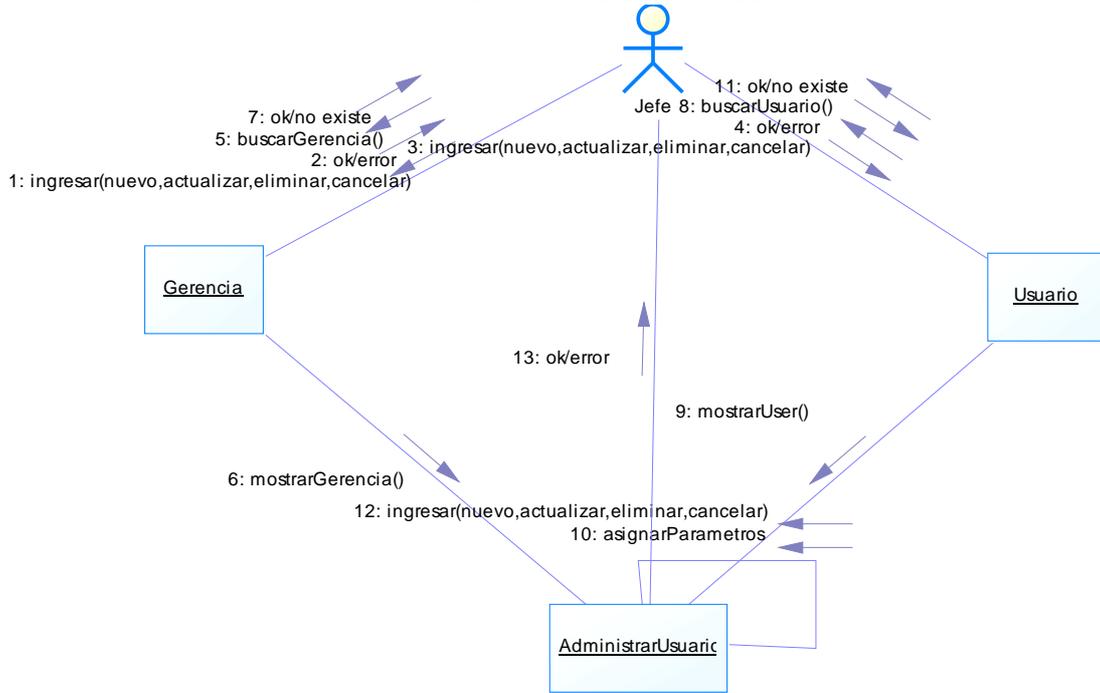


IMPRIMIR REPORTES

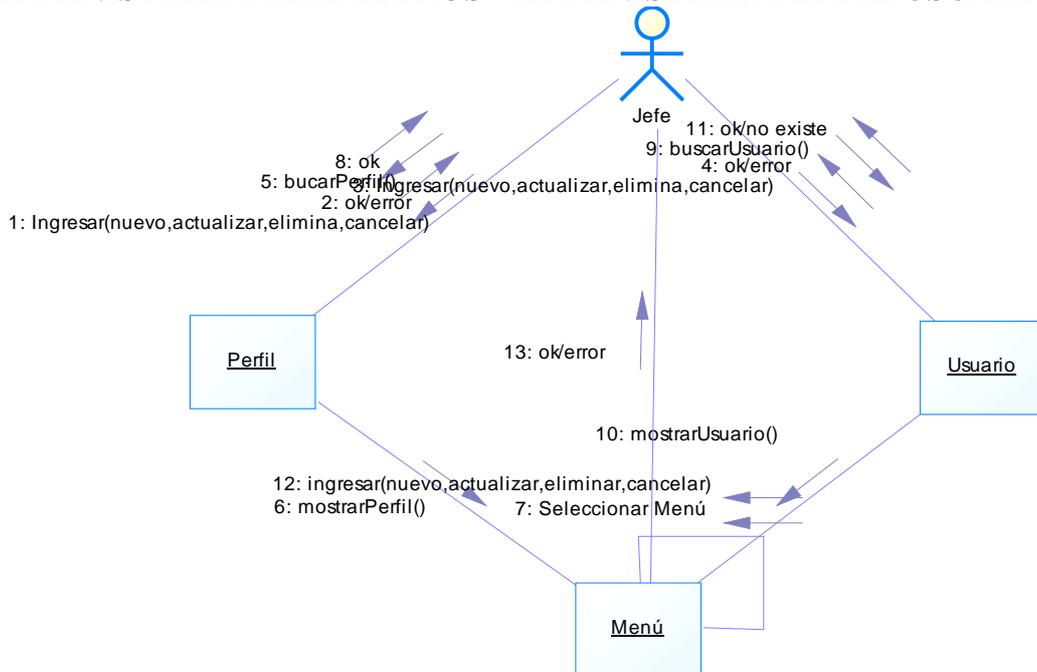


DIAGRAMAS DE COLABORACIÓN

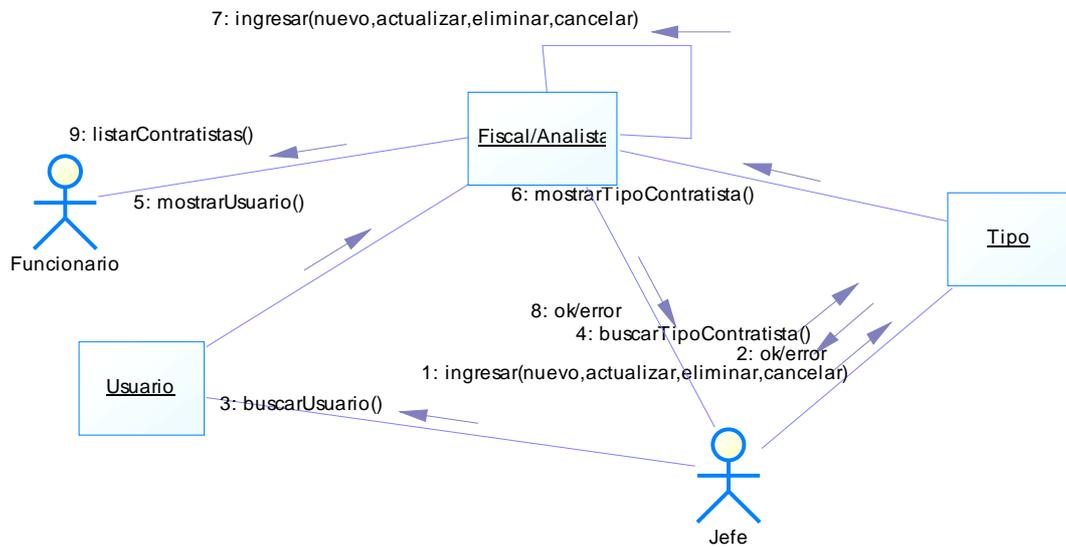
SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR PARÁMETROS ADMINISTRAR USUARIOS



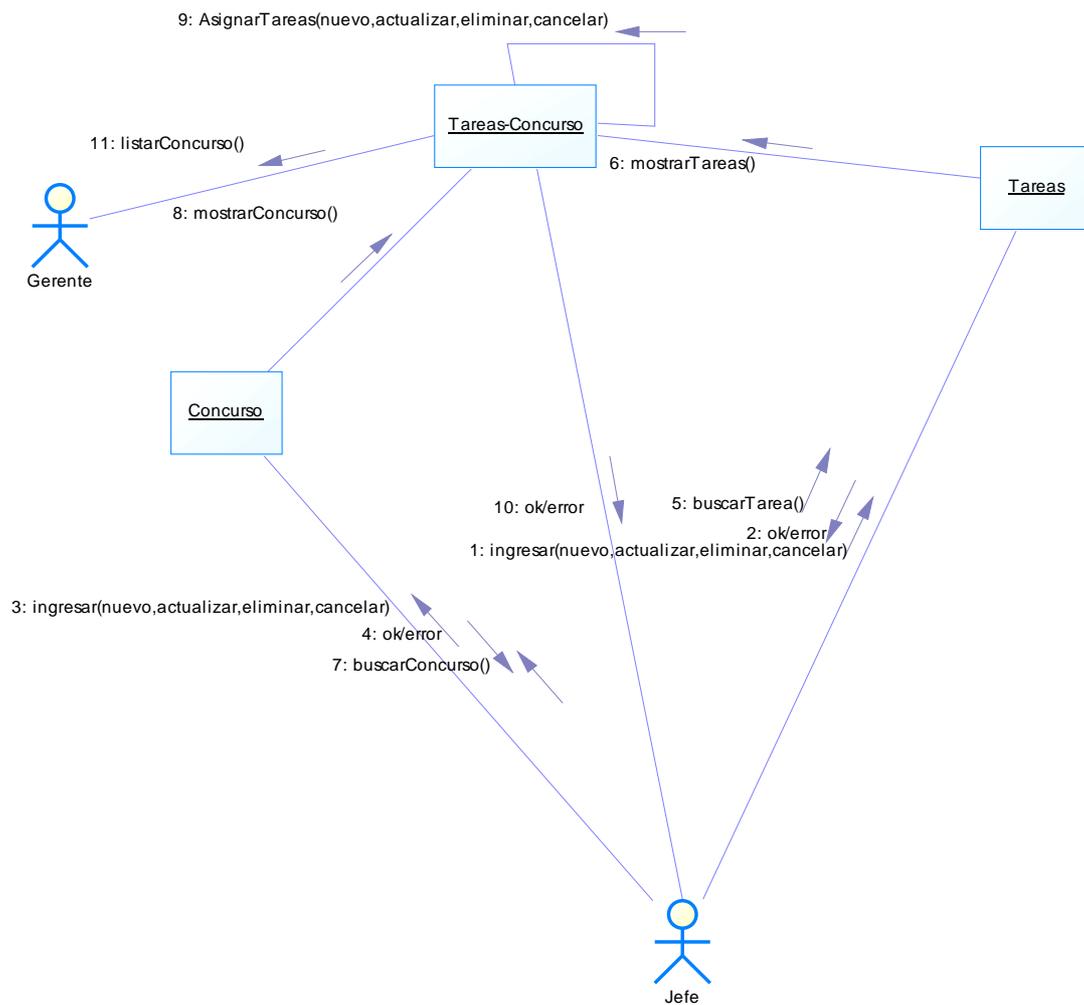
ADMINISTRAR PARÁMETROS - ADMINISTRAR PERFIL USUARIO



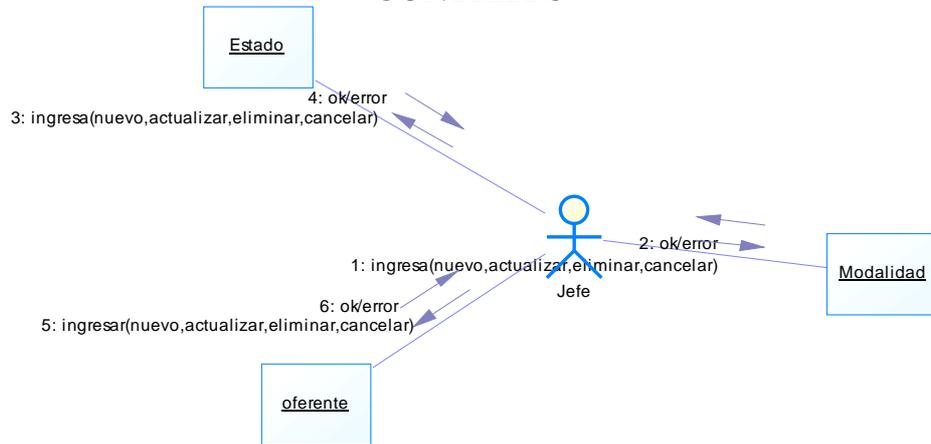
ADMINISTRAR PARÁMETROS - ADMINISTRAR CONTRATISTAS



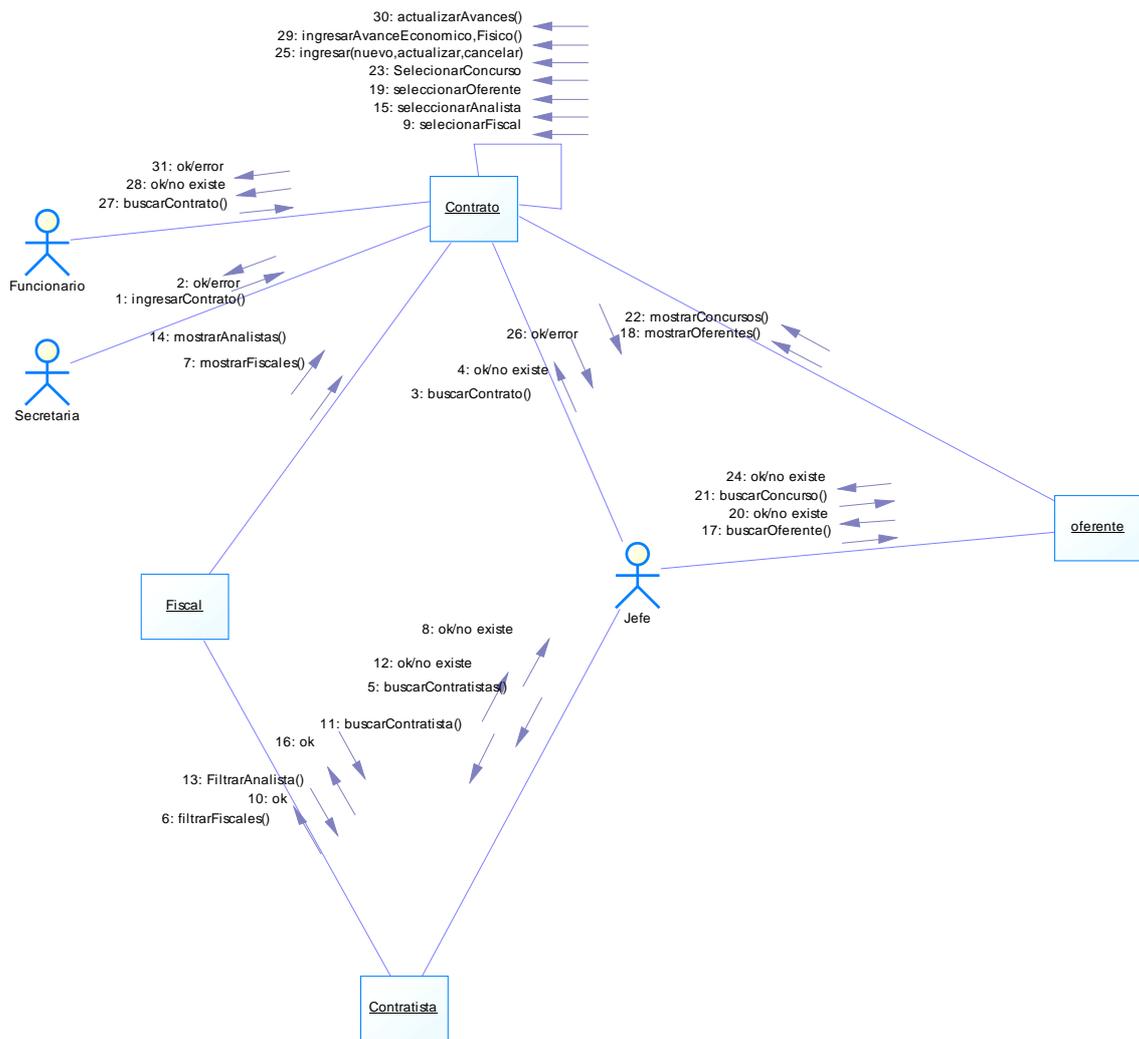
ADMINISTRAR CONCURSOS



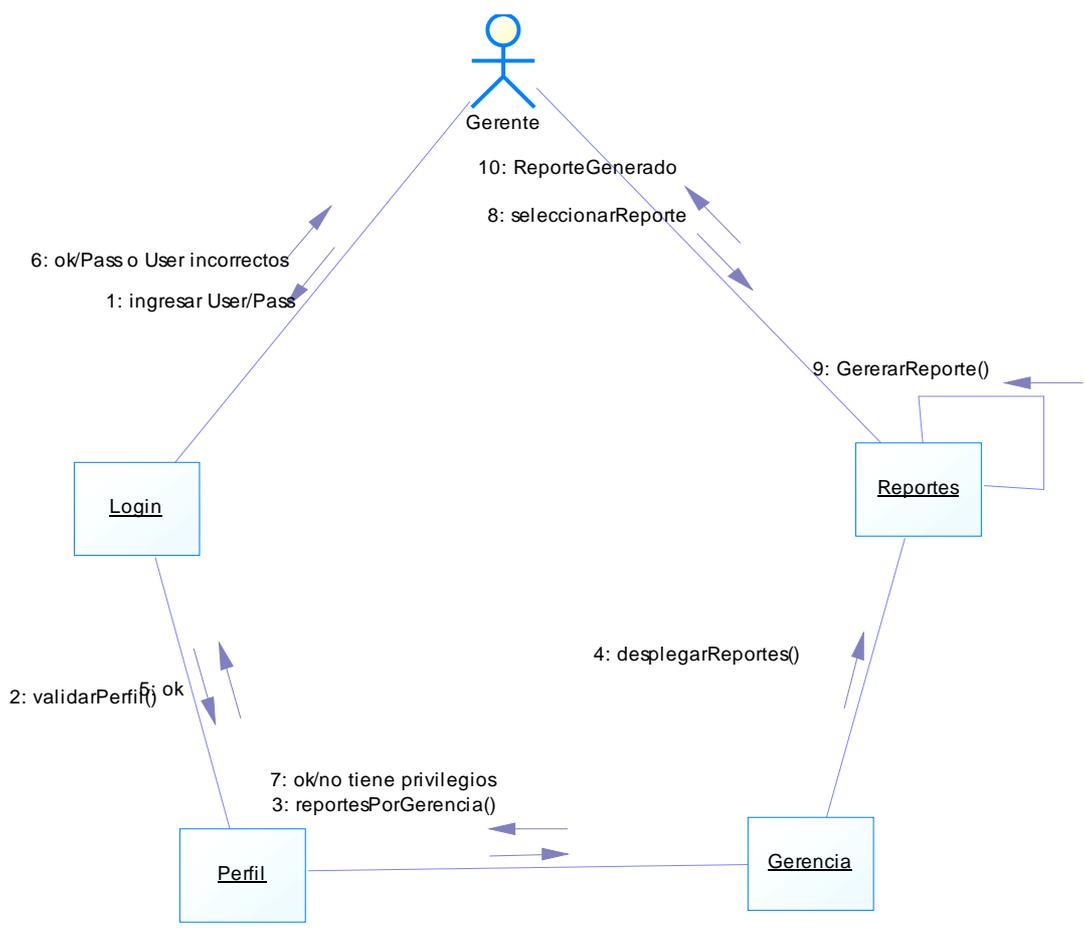
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR CONTRATOS-ADMINISTRAR PARÁMETROS DE
CONTRATO**



ADMINISTRAR CONTRATOS-ADMINISTRAR AVANCE CONTRATOS



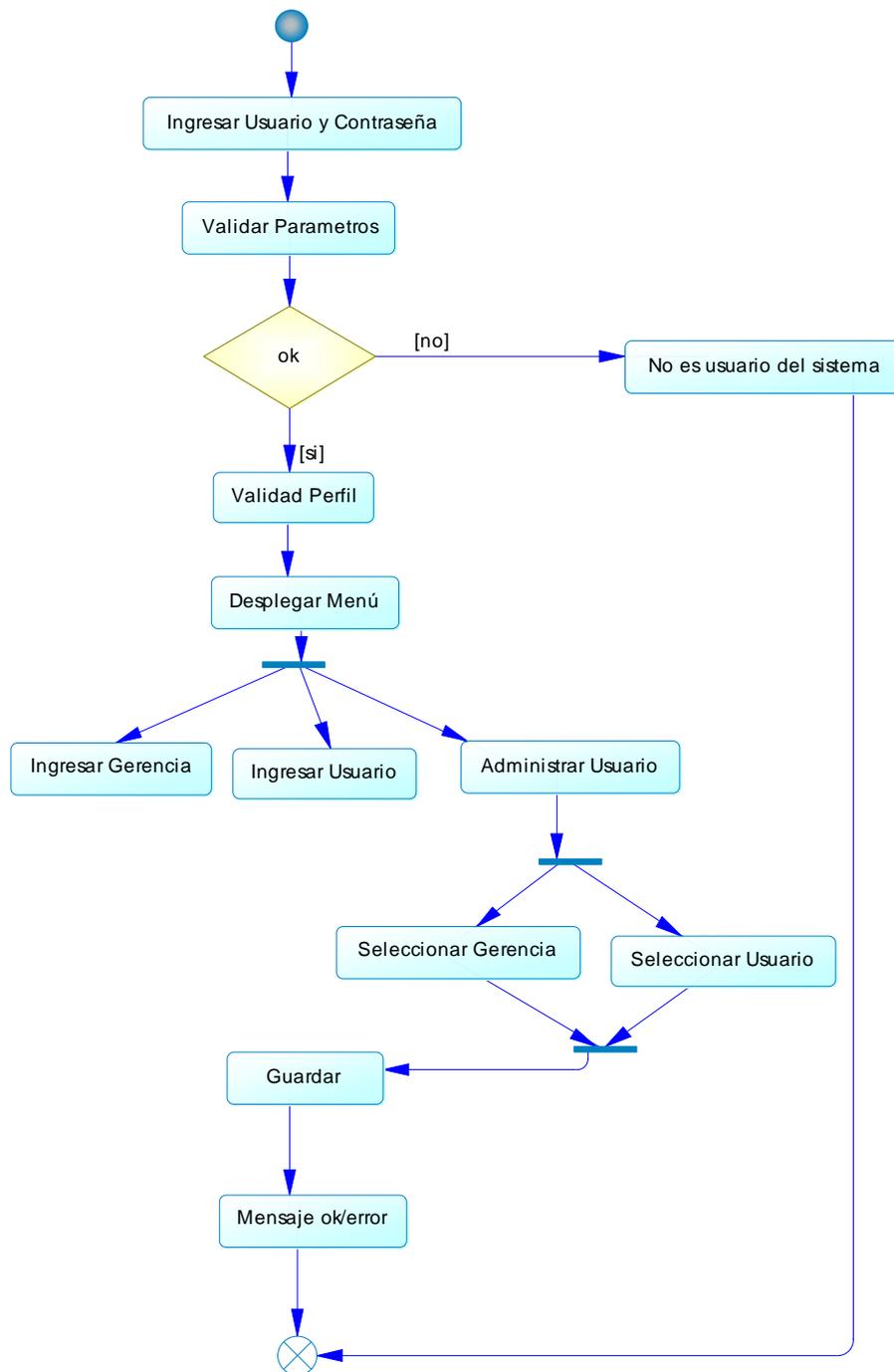
**SISTEMA GESTIÓN DE CONTRATOS PARA
 PETROECUADOR DESARRROLLADO EN LA
 HERRAMIENTA WEB COLDFUSION MX7
 IMPRIMIR REPORTES**



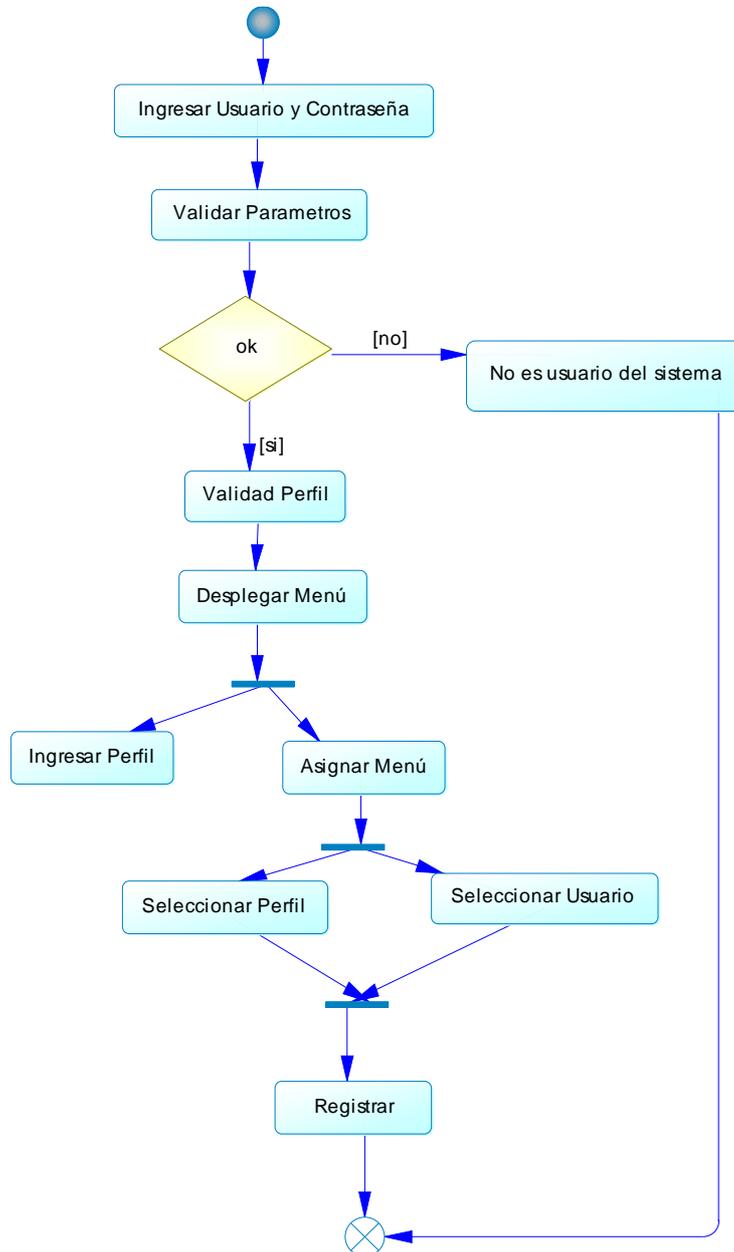
MODELO FUNCIONAL

DIAGRAMAS DE ACTIVIDADES

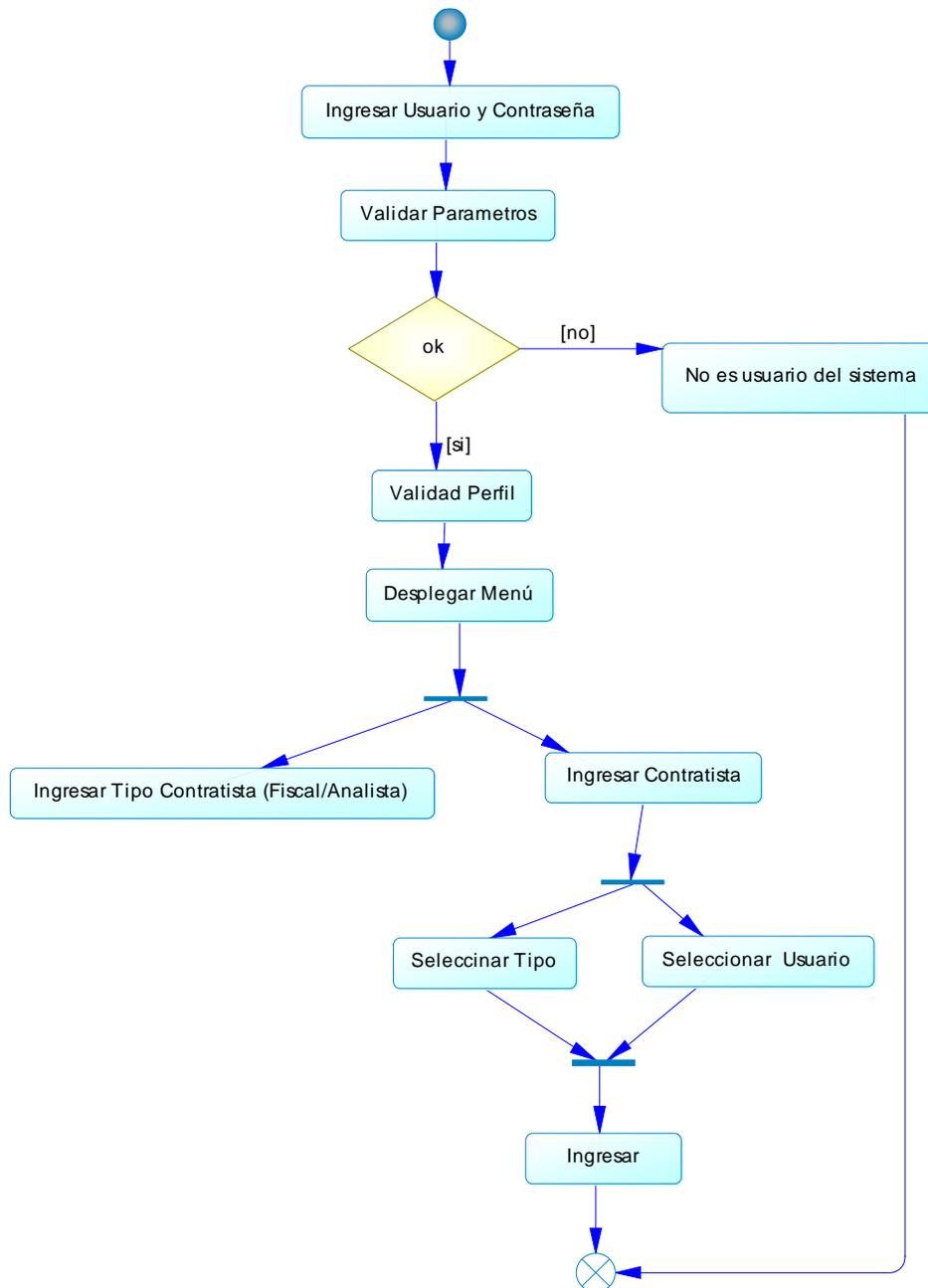
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR PARÁMETROS
ADMINISTRAR USUARIOS**



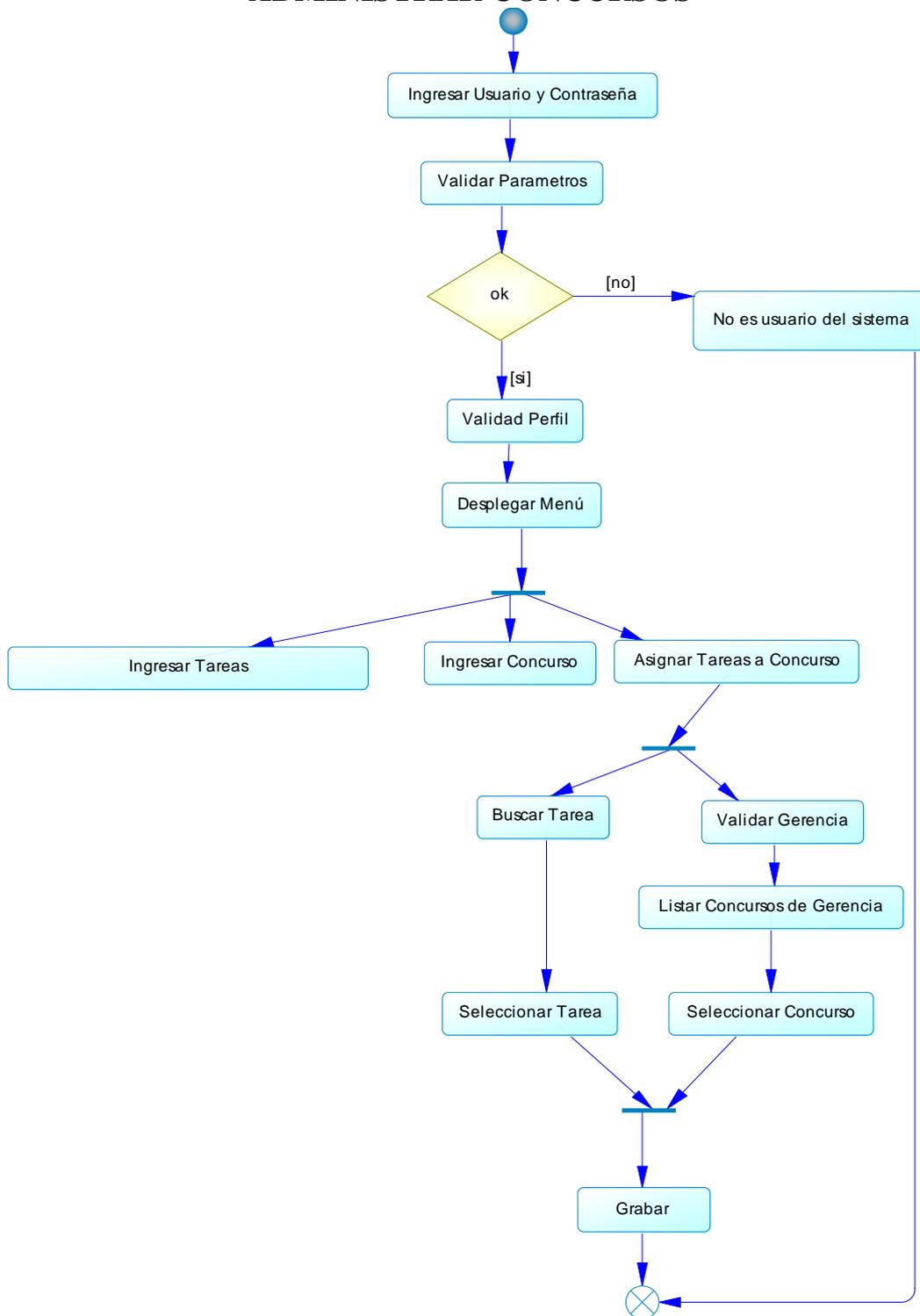
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR PARÁMETROS - ADMINISTRAR PERFIL USUARIO**



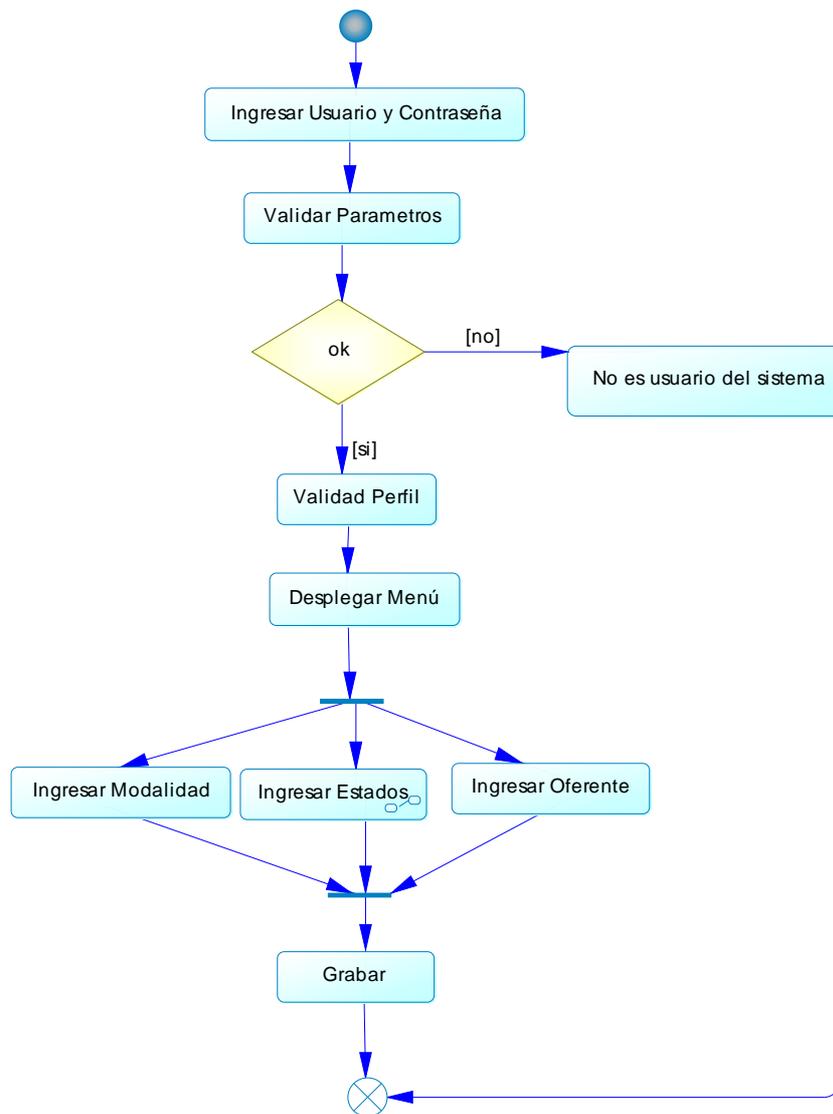
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR PARÁMETROS - ADMINISTRAR CONTRATISTAS**



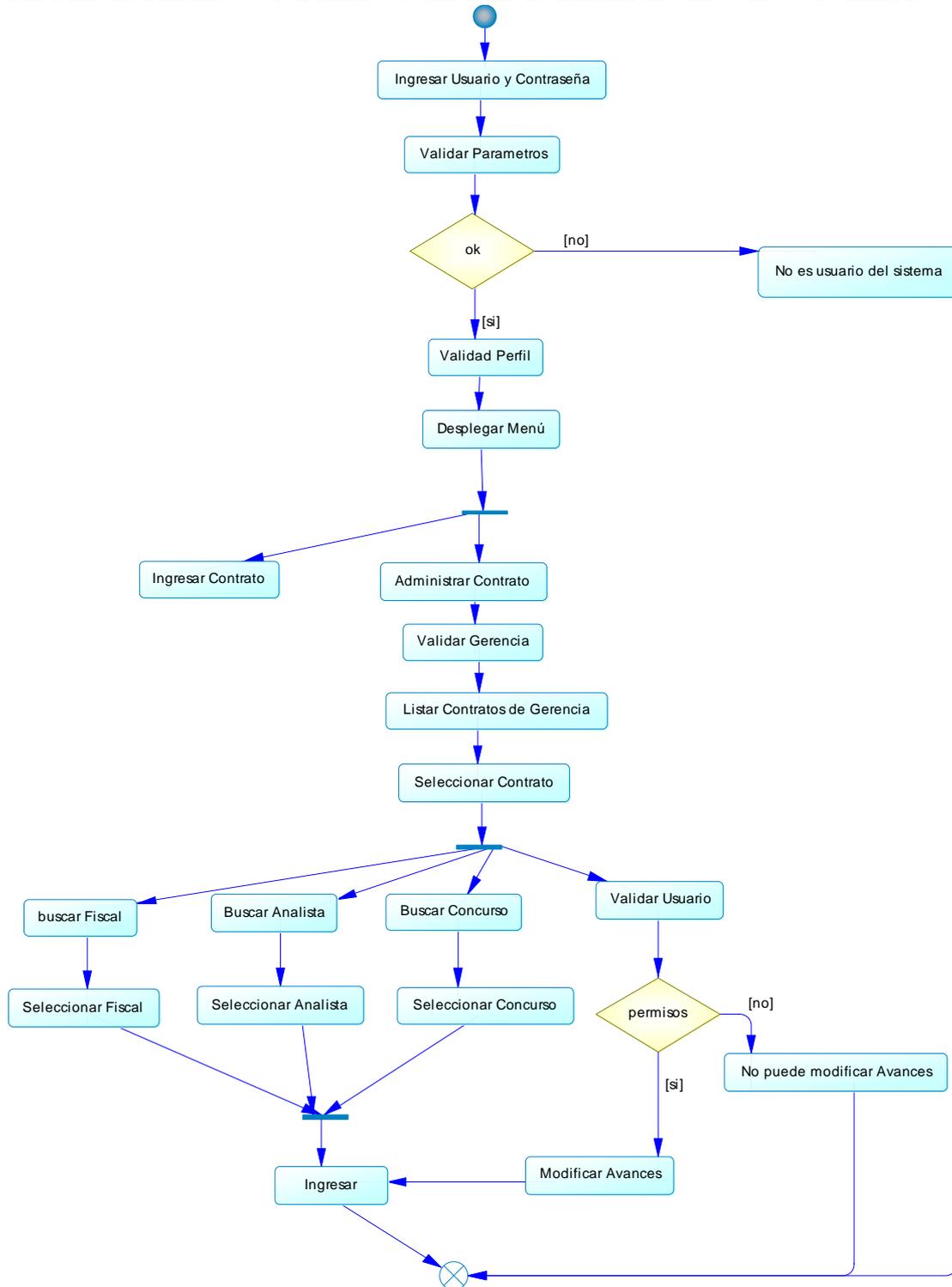
SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR CONCURSOS



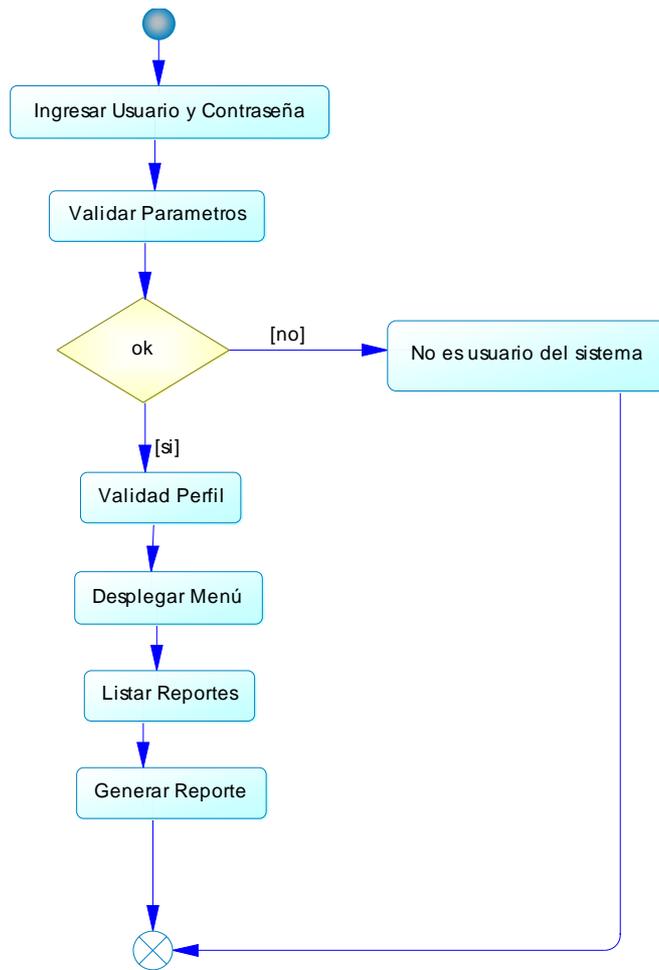
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
ADMINISTRAR CONTRATOS – ADMINISTRAR PARÁMETROS DE
CONTRATO**



SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 ADMINISTRAR CONTRATOS-ADMINISTRAR AVANCE CONTRATOS



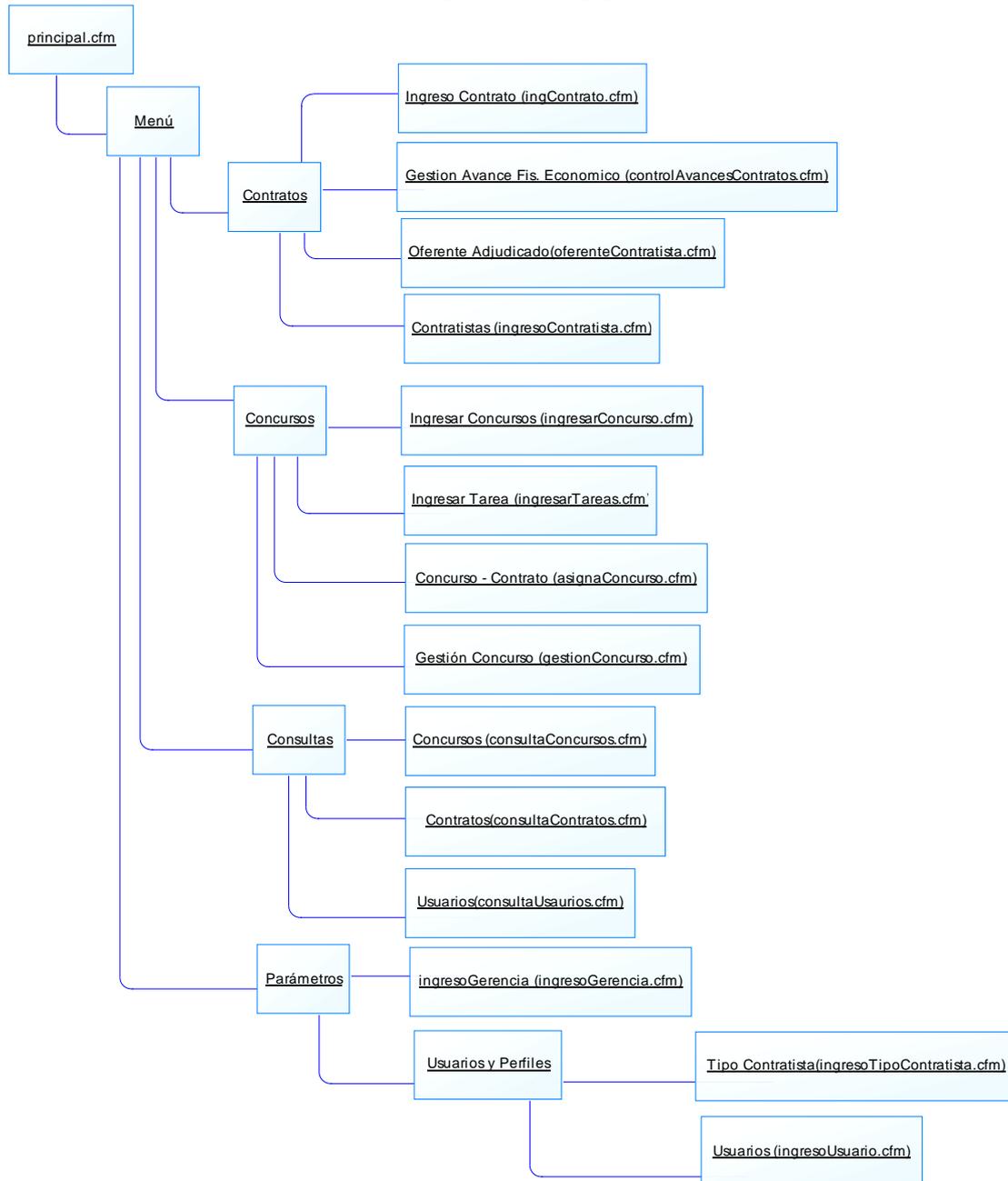
**SISTEMA GESTIÓN DE CONTRATOS PARA
PETROECUADOR DESARROLLADO EN LA
HERRAMIENTA WEB COLDFUSION MX7
IMPRIMIR REPORTES**



DISEÑO NAVEGACIONAL

DIAGRAMA NAVEGACIONAL

SISTEMA GESTIÓN DE CONTRATOS PARA PETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 DIAGRAMA NAVEGACIONAL DEL SISTEMA GESTION DE CONTRATOS



MANUAL DE INSTALACIÓN

SISTEMA GESTIÓN DE CONTRATOS PARA EPPETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 MANUAL DE INSTALACIÓN

INSTALACIÓN DEL SERVIDOR ORACLE ENTERPRISE 11g

Se explica paso a paso la instalación del servidor que se ha utilizado para la implementación de la base de datos:

Requisitos mínimos de Software:

- Una dirección IP estática en el equipo en el que va a instalar Oracle 11g. Si intenta instalar Oracle 11g sin disponer de ella, se le indicará que instale un adaptador de bucle cerrado.
- Un mínimo de 125 MB de espacio libre en el disco para archivos temporales de instalación.
- 1 GB como mínimo de memoria RAM física para Oracle 11g Standard Edition (1,3 GB de RAM para Oracle 11g Enterprise Edition) y 2 GB de memoria virtual.
- 4,55 GB como mínimo de espacio libre en el disco para la instalación básica y 4,92 GB de espacio libre en el disco para la instalación avanzada.

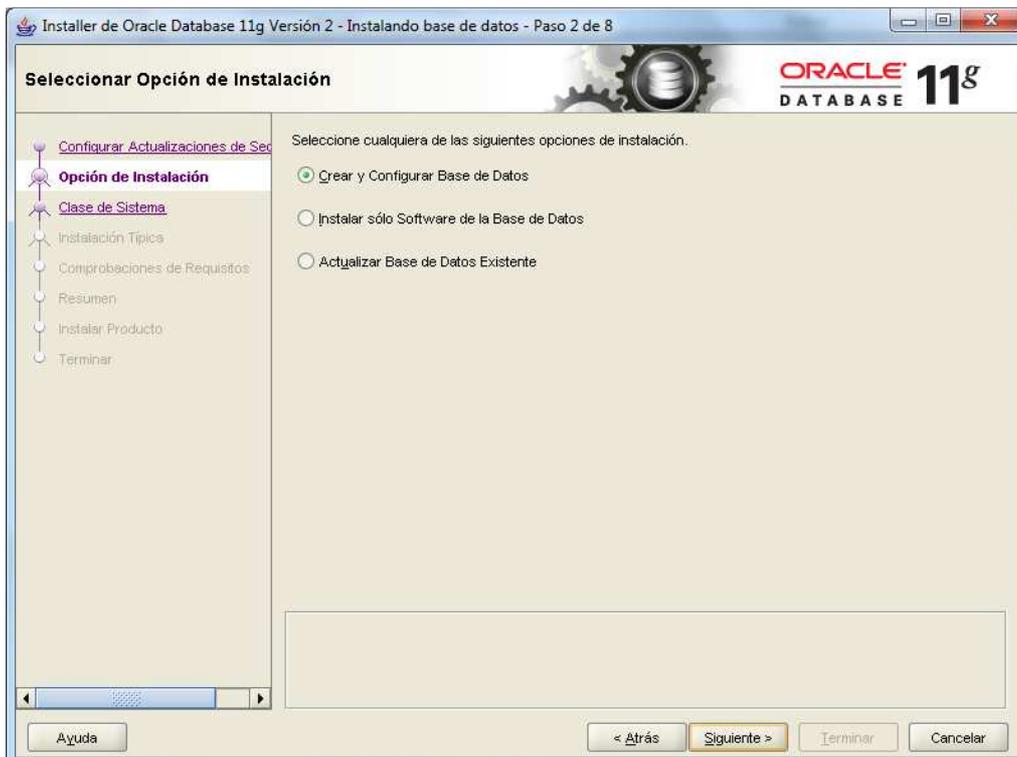
Proceso de Instalación

Tomando en cuenta estos requerimientos se procede a la instalación y configuración del Servidor Oracle 11g a continuación: doble clic sobre el icono del programa.

1. Ingresar una dirección de correo electrónica (opcional)



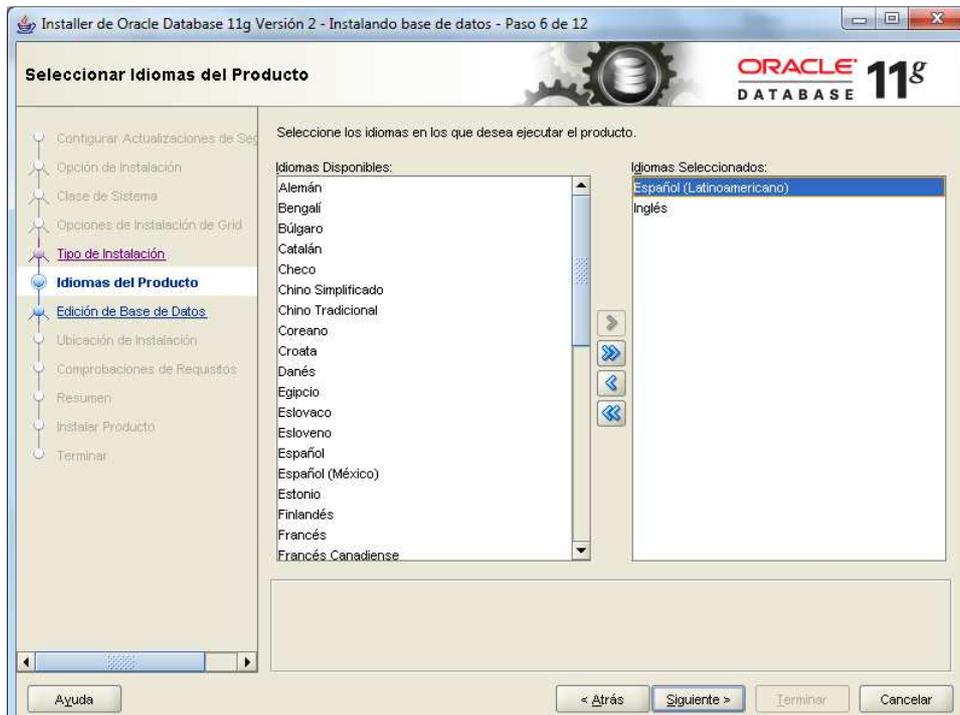
2. Seleccionar la primera opción para crear una base de datos Global



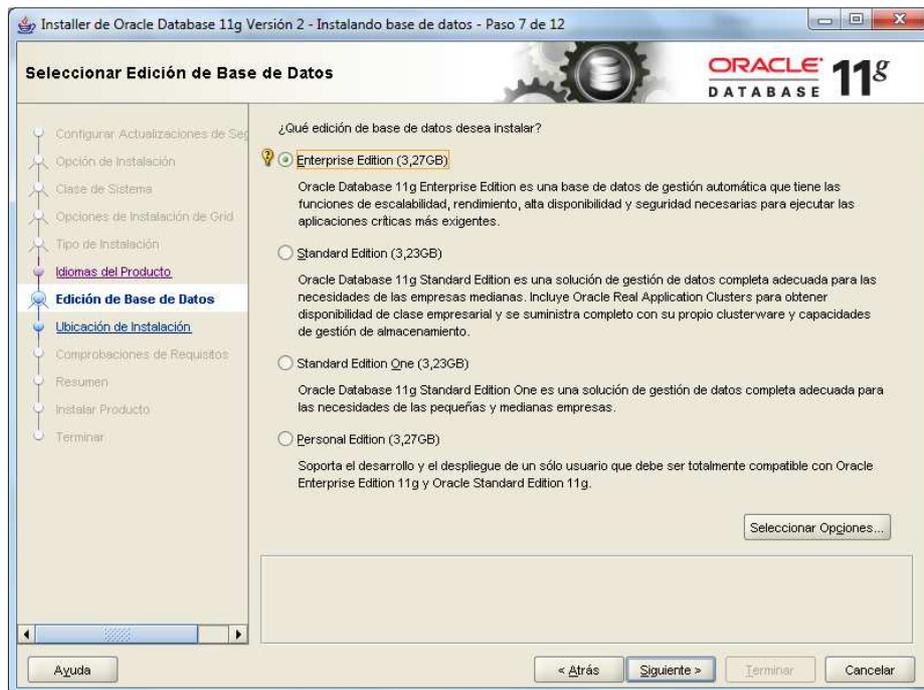
3. Seleccionar la primera opción para instalar la base de datos como una instancia única.



4. Seleccionar el idioma para la instalación del paquete oracle.



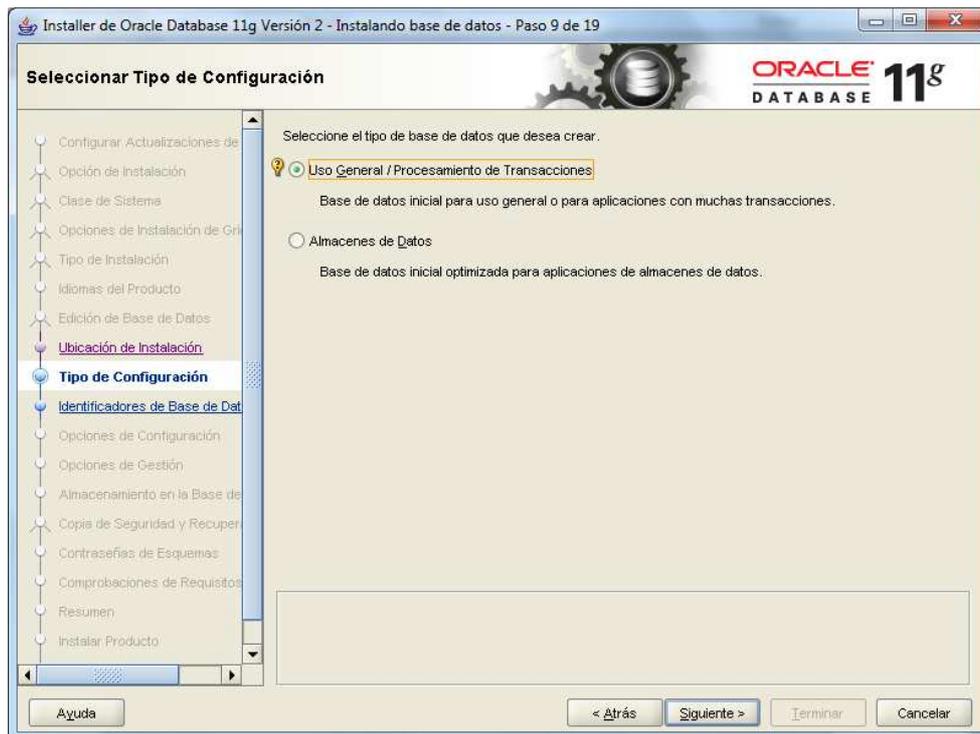
5. Seleccionar la edición de base datos, Enterprise Edition, que gestiona automáticamente las funciones de rendimiento



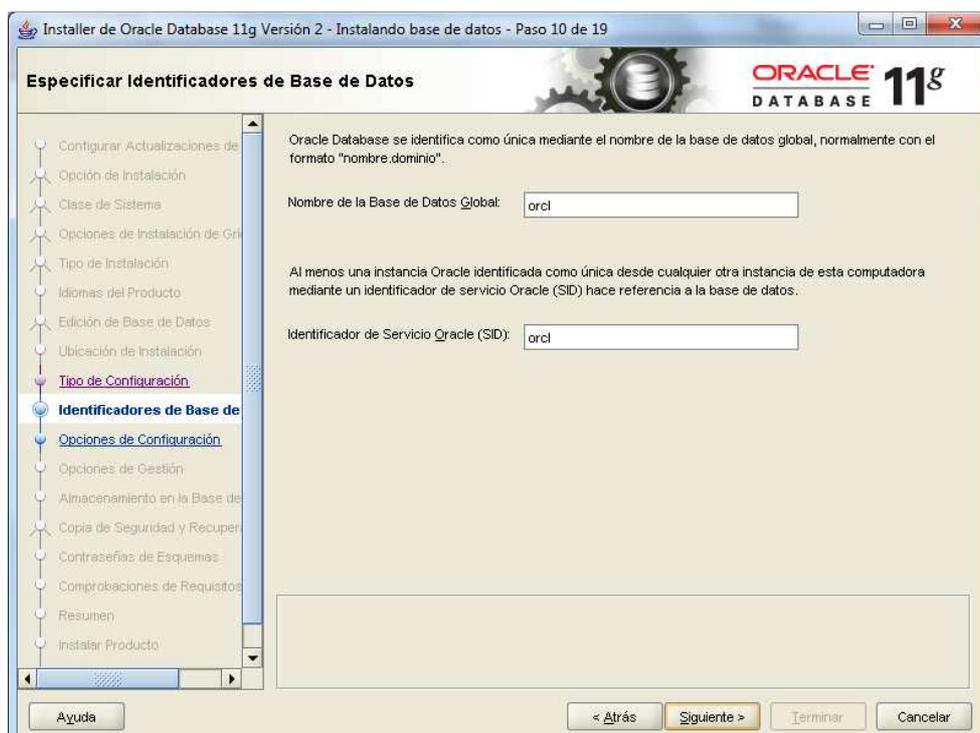
6. Seleccionar la ruta para la instación de los archivos relacionados con la configuración de Oracle 11g



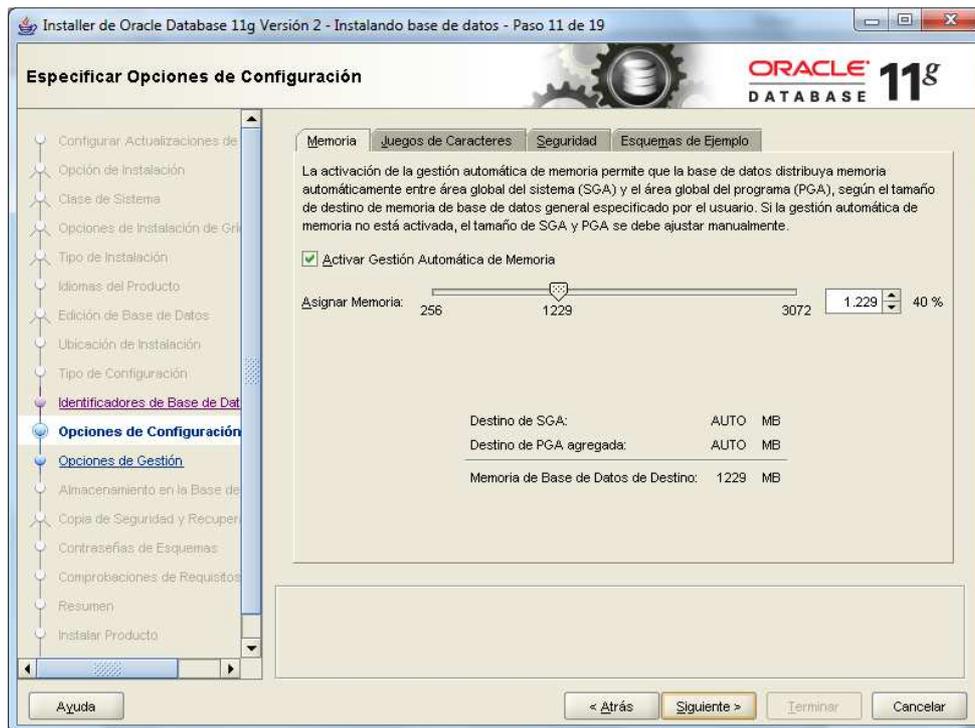
7. Seleccionar el tipo de base de datos a crear, en nuestro caso seleccionamos Uso General.



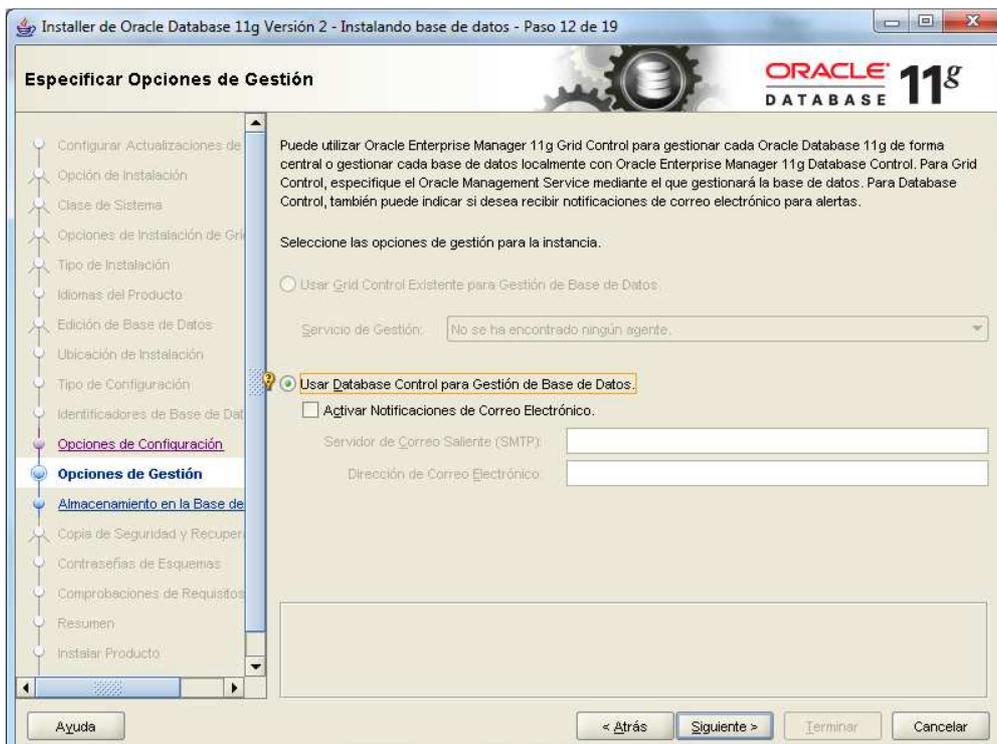
8. Ingresar el nombre de la base de datos Global



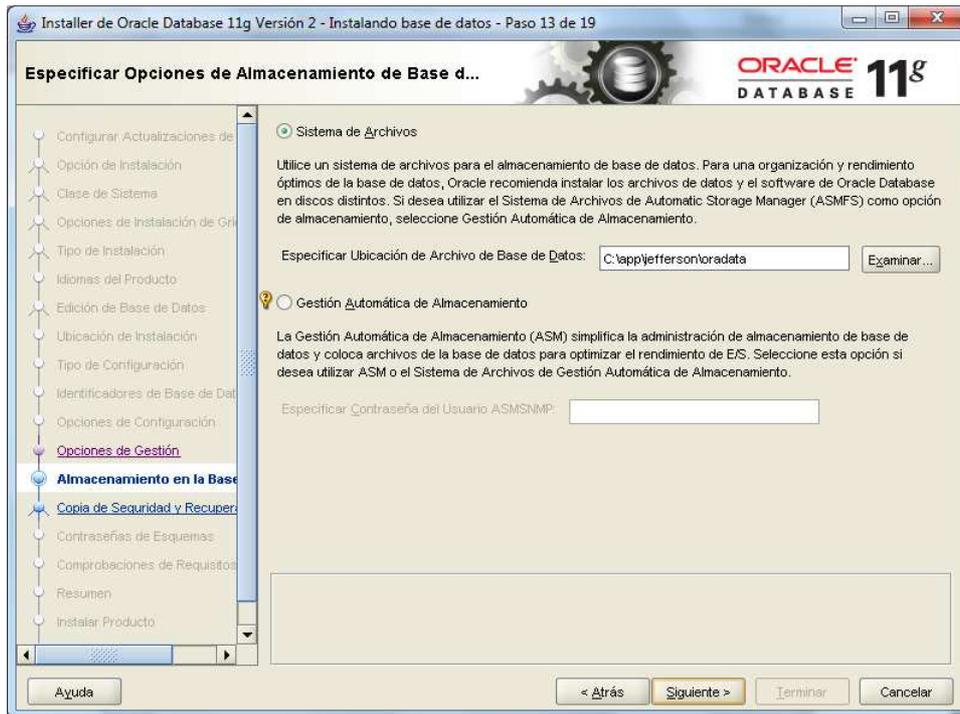
9. Activar la casilla Gestión automática de Memoria, para que el servidor gestione la memoria de manera conveniente



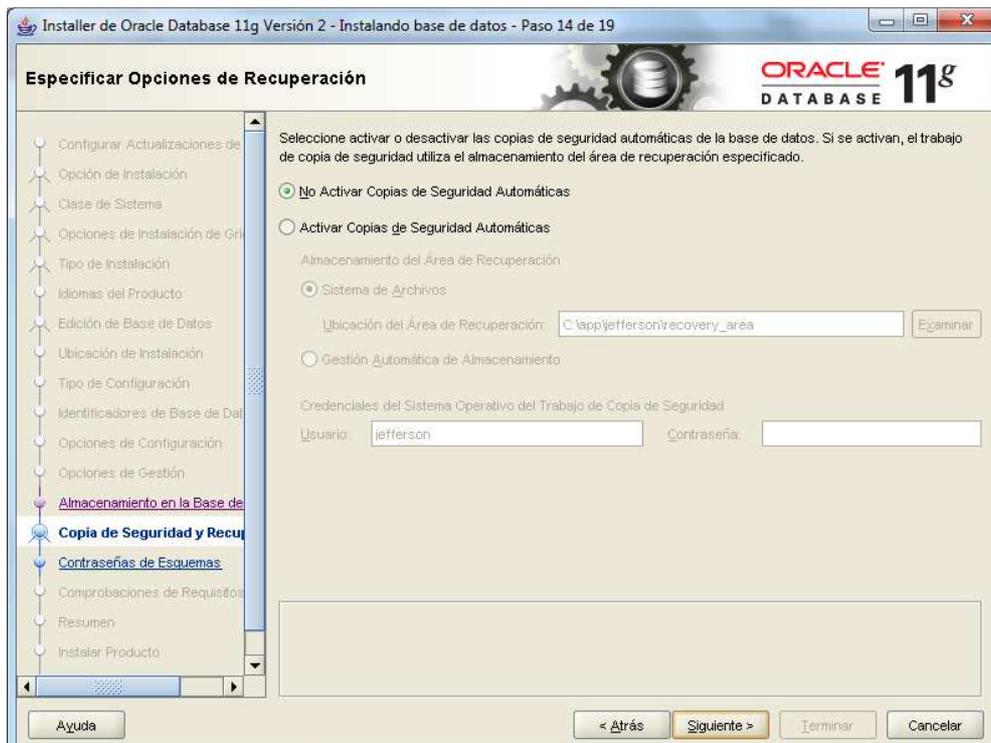
10. Seleccionar User Database Control para gestional las bases por usuarios



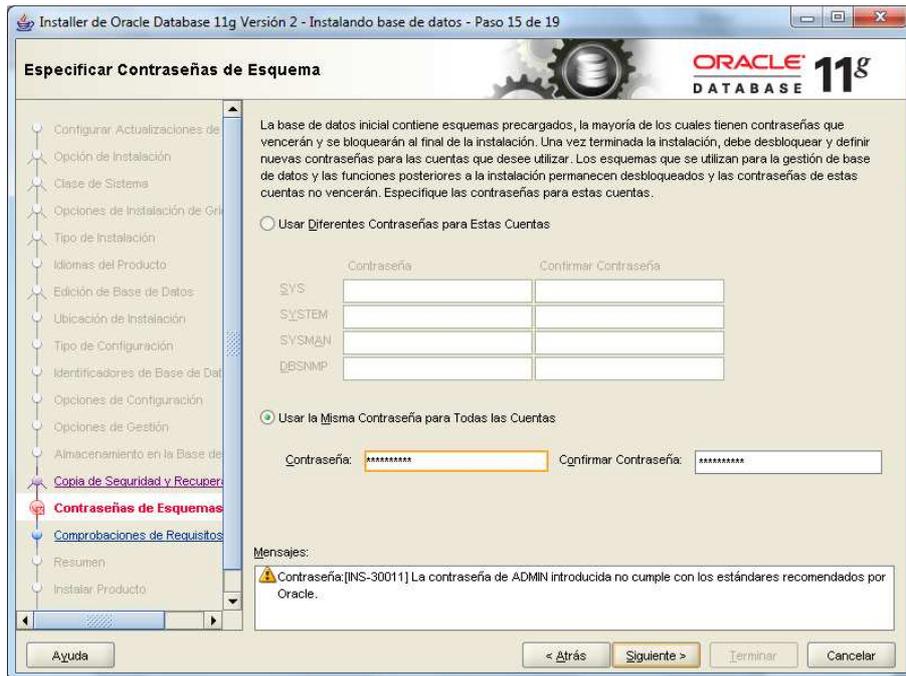
11. Seleccionar Sistema de Archivos para el almacenamiento de base de datos.
Esto optimiza el rendimiento del servidor



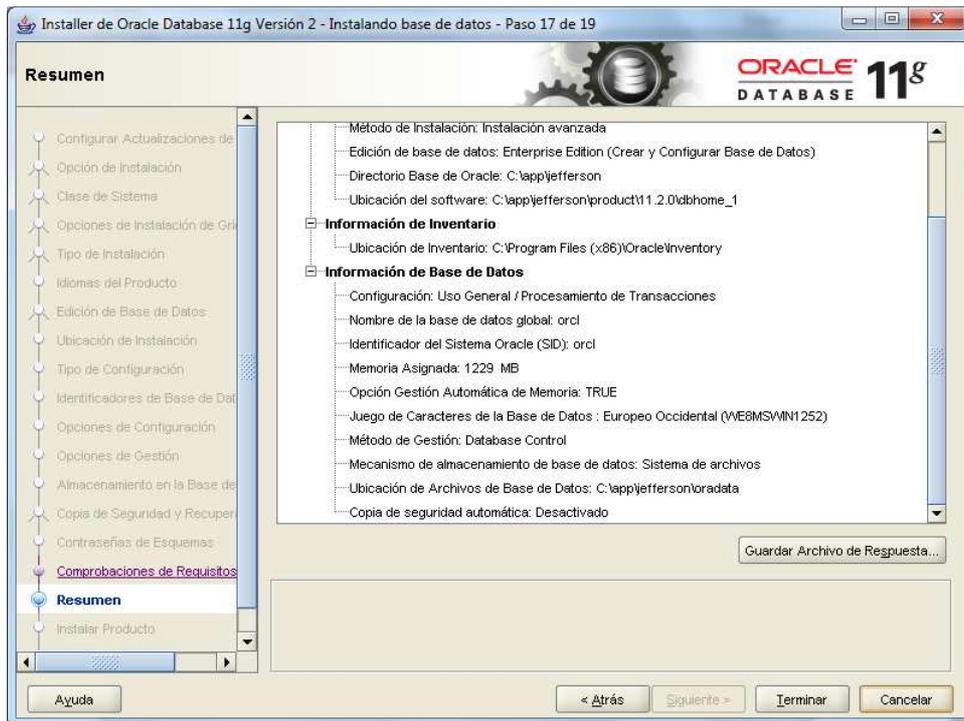
12. Seleccionar No activar Copias de seguridad Automáticas



13. Ingresar una clave para los usuarios default de Oracle.



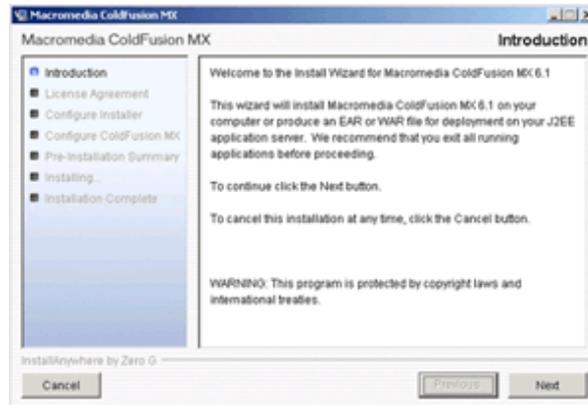
14. Reporte de la configuración de nuestro servidor Oracle



INSTALACIÓN DEL SERVIDOR COLD FUSION

Se detalla paso a paso la instalación del servidor que se ha utilizado para el desarrollo del aplicativo web:

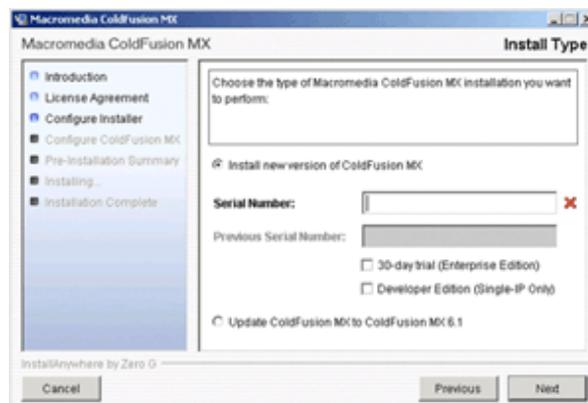
1. Introducción



2. Aceptar Licencia



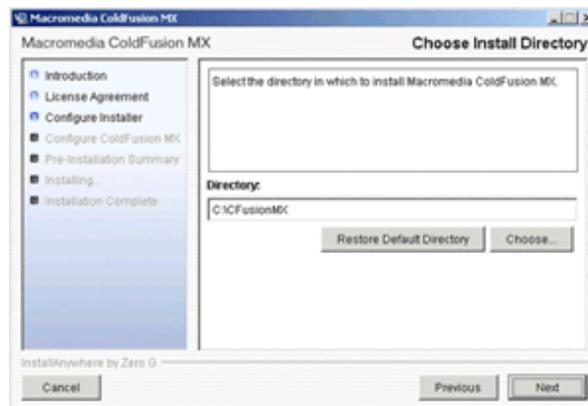
3. Seleccionar el tipo de instalación *Developer*



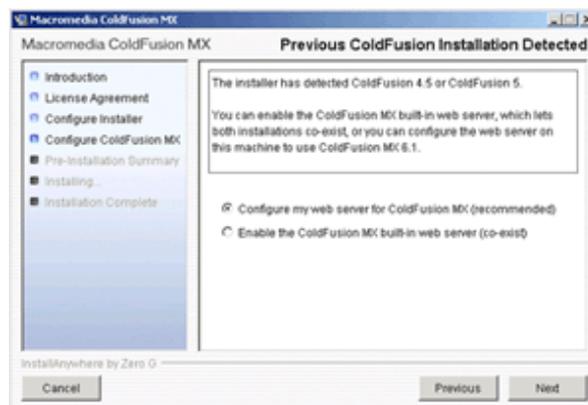
4. Seleccionar **Server configuration**



5. Escoger directorio de Instalación



6. Configurar directorio para los sitios web



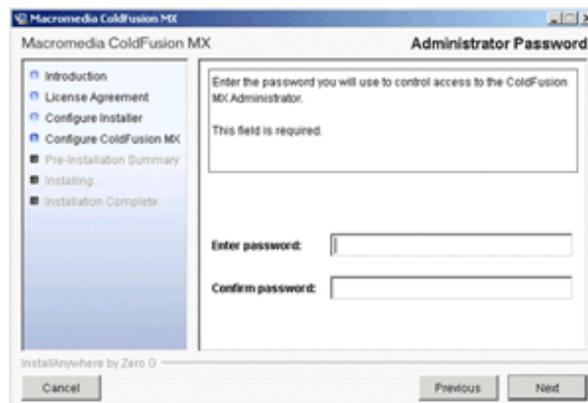
7. Seleccionamos el servidor web Coldfusion



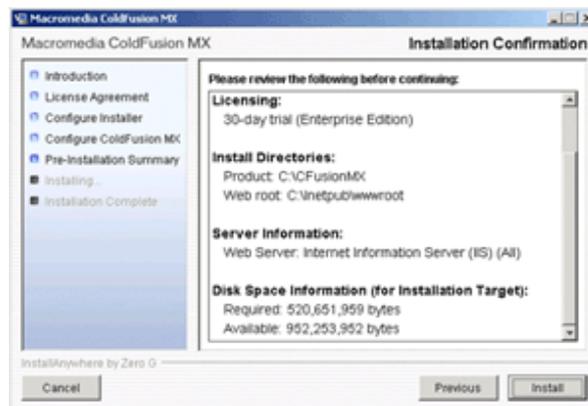
8. Escoger la localidad donde se encuentra el administrador de Coldfusion



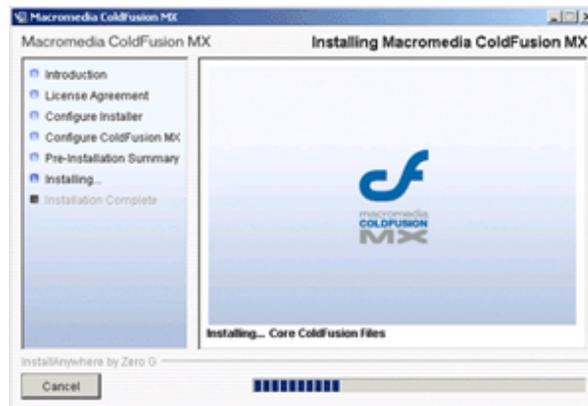
9. Ingresar password para el administrador de Coldfusion



10. Confirmar instalación



11. Instalar

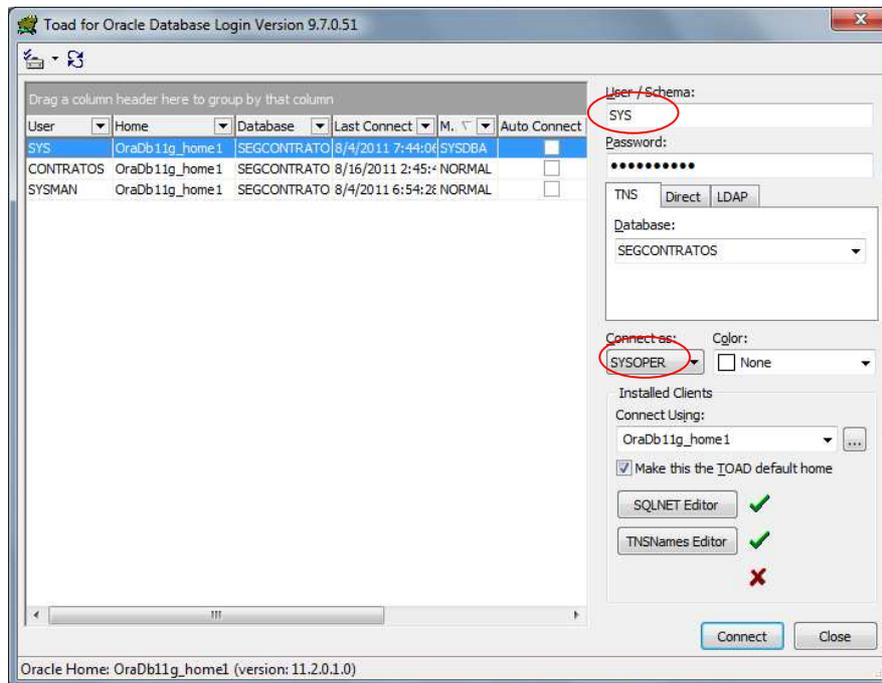


12. Reporte de instalación

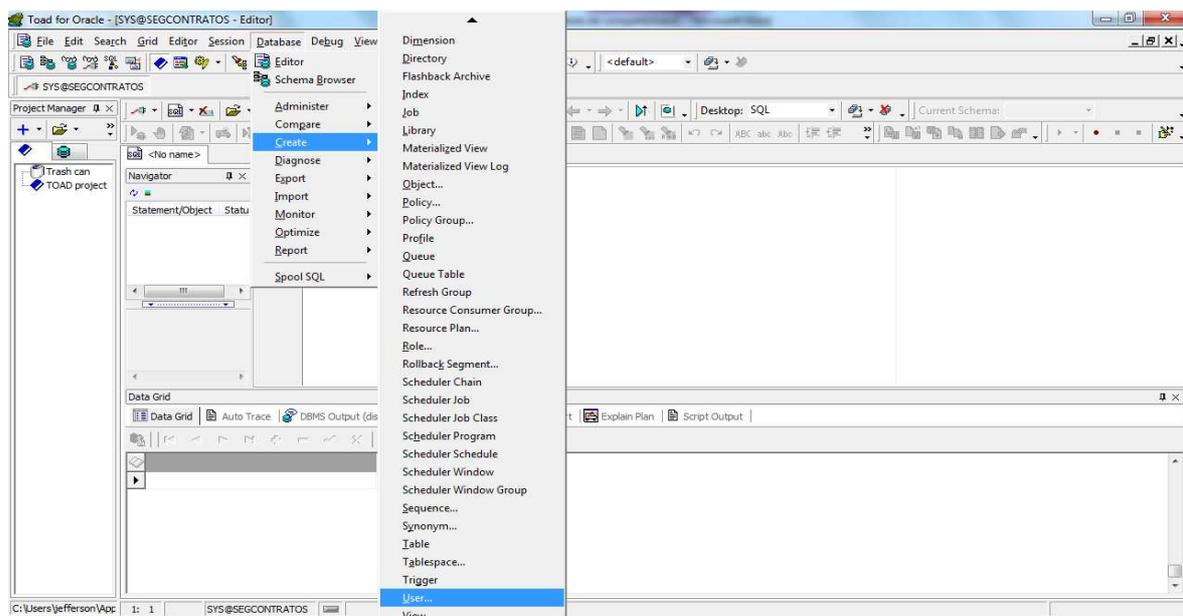


CONFIGURACIÓN DE ORACLE Y COLDFUSION

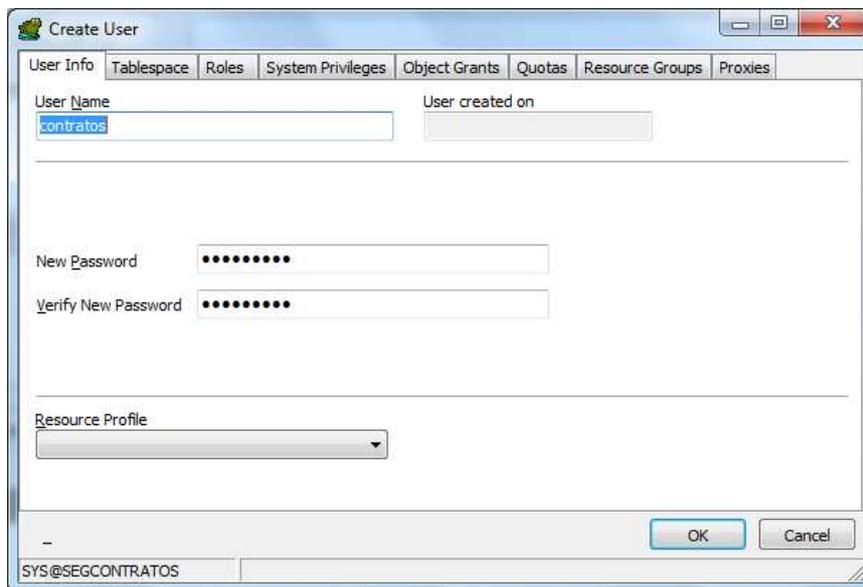
Con la herramienta Toad for Oracle ingresamos a la base de datos SEGCONTRATOS con el usuario SYS, y como Sysoper para crear un Schema y sobre ella crear nuestras tablas para el aplicativo Gestión de Contratos.



En la opción menú Database opción Create User, ingresar un nuevo usuario para crear las tablas del aplicativo.



Ingresar el nombre del usuario o esquema y password



Asignamos al esquema a un *TABLESPACE*. Ejecutamos en el editor de SQL el siguiente script

```
ALTER USER contratos DEFAULT TABLESPACE users.
```

Asignar privilegios al esquema Contratos. En el sql editor ejecutamos el siguiente script.

```
GRANT ALL PRIVILEGES to contratos
```

Salir de la sesión SYS e ingresar con el usuario Contratos para crear las tablas del aplicativo.

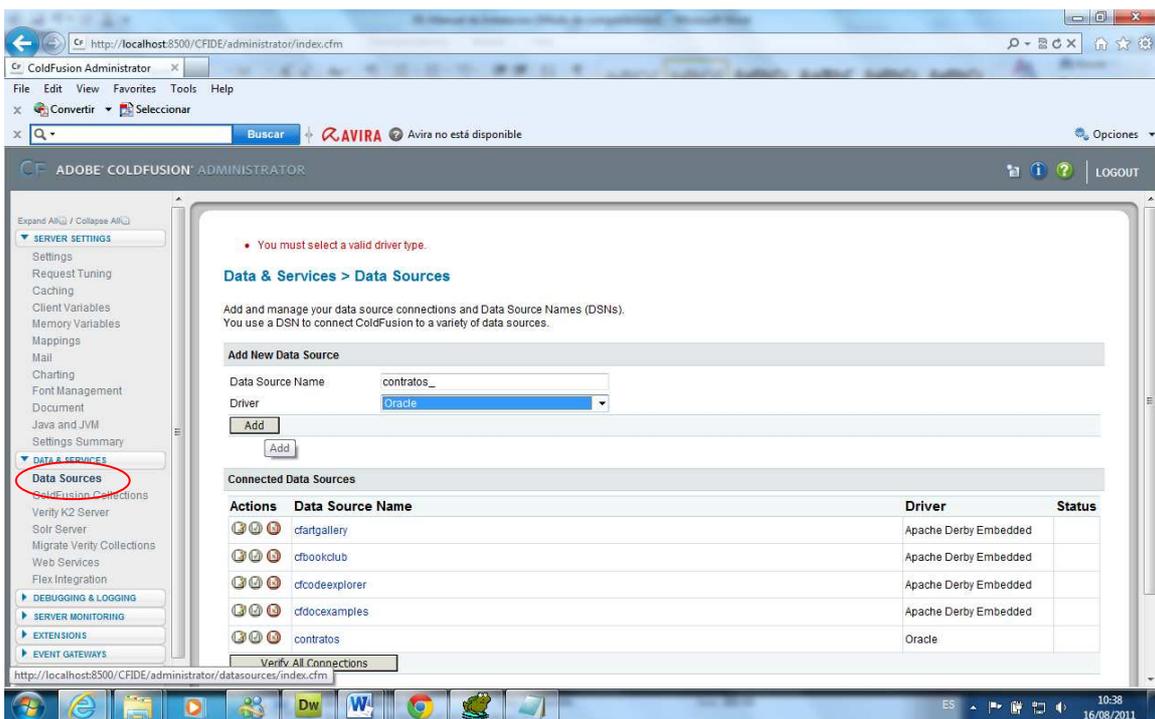
CONFIGURACIÓN DE COLDFUSION

Para configurar la conexión de coldfusión a la base de datos Oracle ingresamos al administrador de coldfusión en la siguiente dirección:

<http://localhost:8500/CFIDE/administrator/> , en la pantalla ingresamos el password que se asignó para el administrador al momento de la instalación.

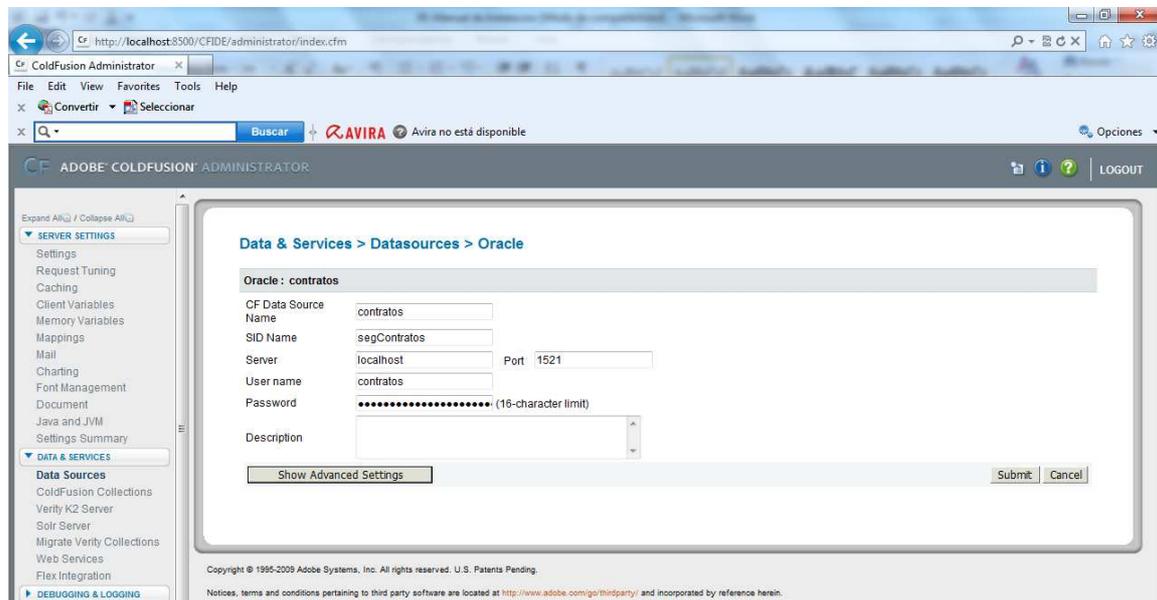


Seleccionar la opcion Data Source



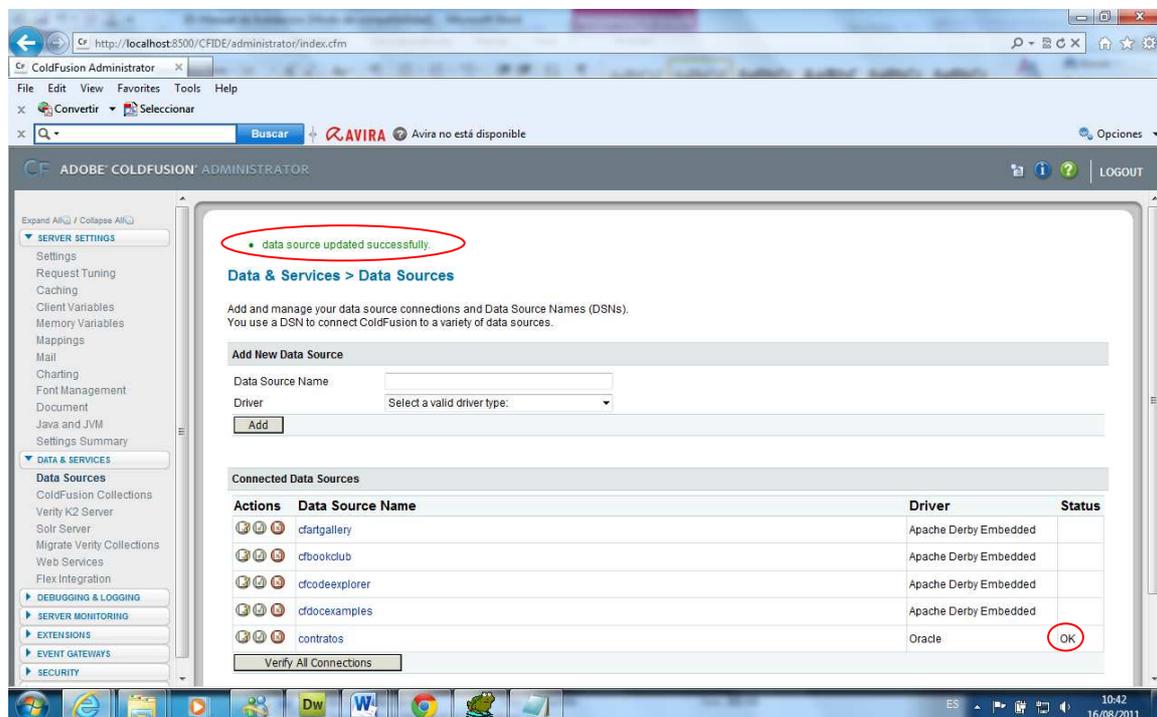
En la pantalla Data Source ingresamos un alias para la base de datos Contratos y el driver con el que se va a conectar coldfusion (oracle). En este caso se la asigno el mismo nombre y presionar el botón ADD.

En esta pantalla ingresamos el SID Name de la base: segContratos, server: localhost, user name: contratos, y password.

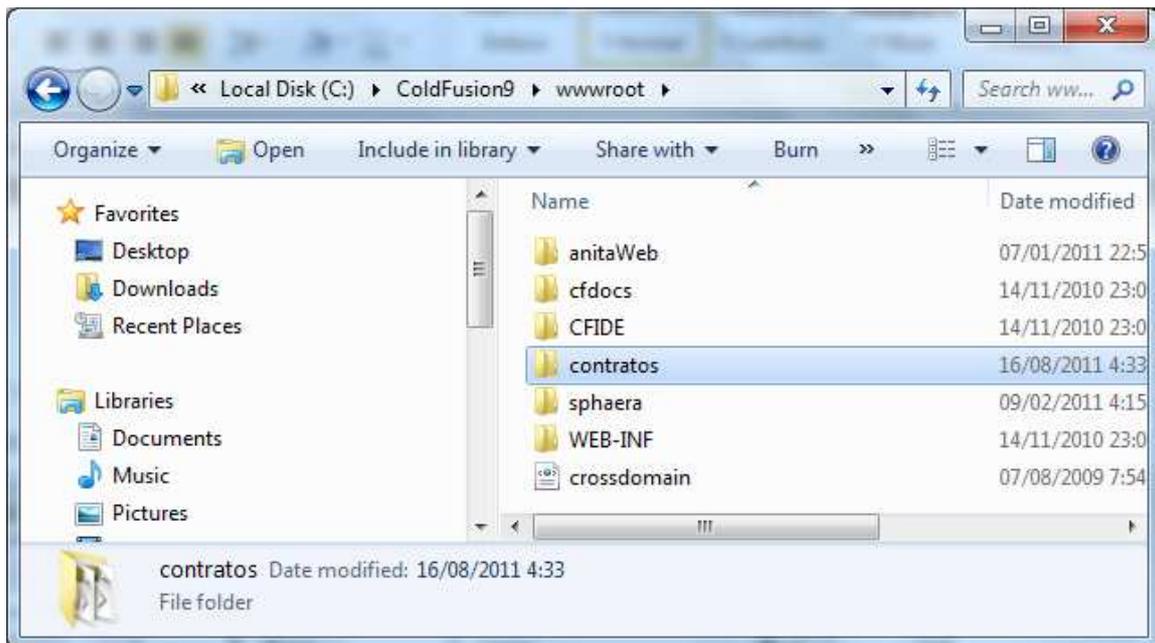


Luego de ingresar los datos para la conexión presionar *Submit*.

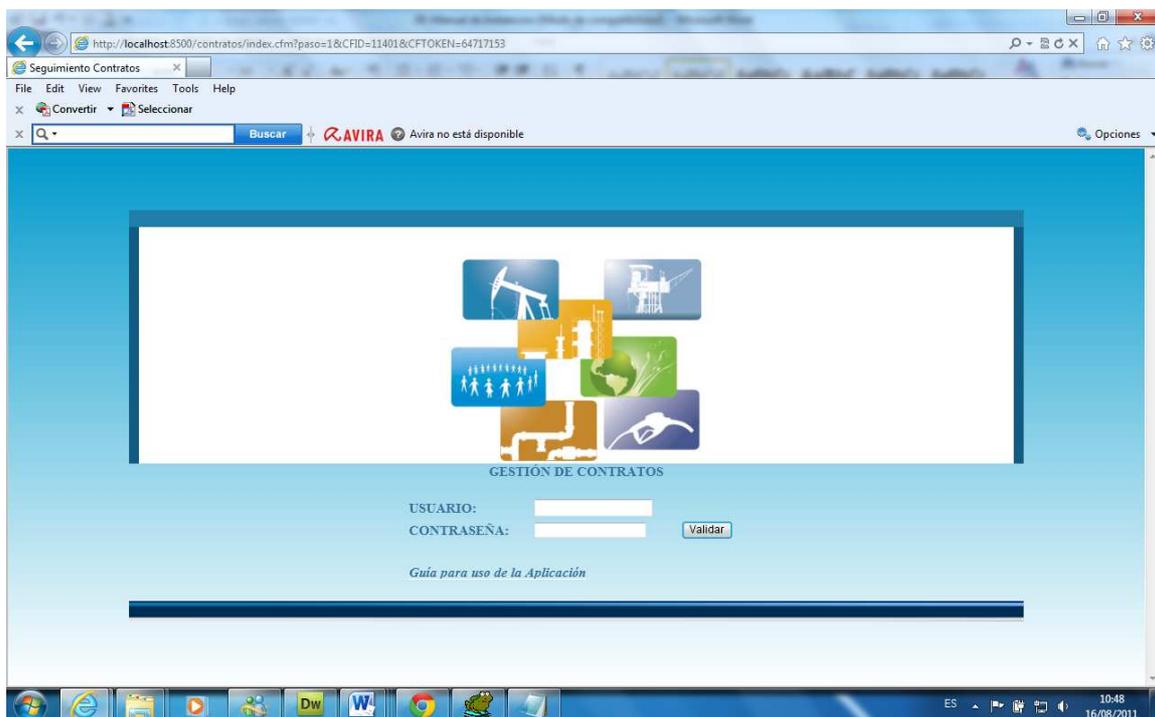
Si la conexión fue exitosa se mostrará un mensaje en color verde en la parte superior de página.



Una vez configurado oracle y coldfusion ubicar los fuentes del aplicativo en el directorio del administrador de coldfusion. C:\ColdFusion9\wwwroot



Para ingresar al aplicativo Gestión de Contratos, ingresar a la siguiente dirección:
<http://localhost:8500/contratos>



MANUAL DE USUARIO

GUIA PARA USO DE LA APLICACIÓN SEGUIMIENTO DE CONTRATOS

Ingreso a la Aplicación

1. Ingresar a la siguiente dirección:

<http://www2.petroecuador.com.ec:8500/contratos> o a través de la intranet de PETROECUADOR en la opción

2. Ingresar Usuario y Password, el mismo que es el mismo que el acceso a intranet.



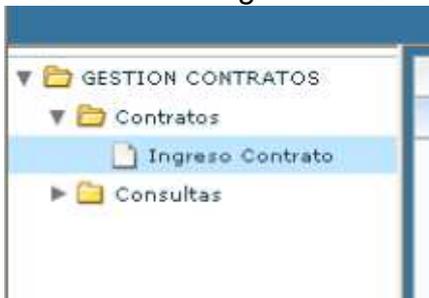
3. En la parte superior izquierda se encuentra una guía para uso de la aplicación.

Ingreso de Contratos:

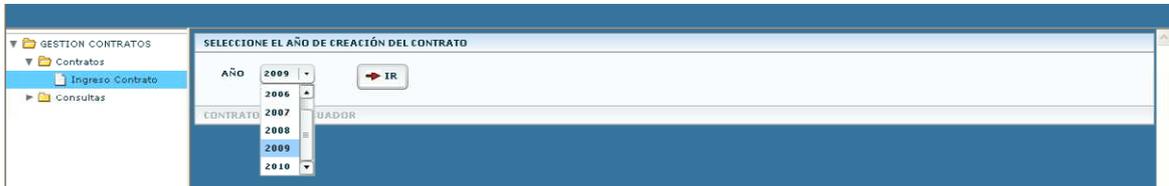
1. Ingresar en la opción *Contratos* del menú.



4. Seleccionar *Ingreso de Contrato*.



5. El ingreso de los contratos es por año, por es necesario seleccionar el Año del Contrato y presionar el botón **IR**



6. A continuación se carga la pantalla para el ingreso de Contratos

SELECCIONE EL AÑO DE CREACIÓN DEL CONTRATO
CONTRATOS PETROECUADOR

Unidad: SISTEMAS

No. Contrato:

Modalidad: (Seleccione)

Estado: (Seleccione)

Fecha Inicio:

Fecha Fin:

Duración (Días) Meses Días

Empresa Contratista:

Fiscalizador/Responsable:

Objeto del Contrato:

Año del Contrato: 2009

Monto: Monto con Iva: (%)Avance Físico:

(%)Anticipo: Anticipo USD: (%)Avance Económico:

Multas: (%)Multas:

Valor Pagado a la Fecha:

Saldo:

Observaciones:

Nº Contrato	Objeto	Duración	Empresa Contratista	% Av.Físico	%Av.Económ	Monto	Saldo	Nom. Fiscal	Inicio	Fin	Fec. Cambio
0001	VBNVBNVN	88	NVB	25	22.32	1120	870	BVNVBN	21/08/2009	21/08/2009	24/08/2009
1-001	SUMINISTROS DE OFIC	1528	COGECOMSA	25.36	35.89	57120	36620.75	EDWIN MO:	10/03/2009	04/11/2009	24/08/2009
10-090	300 canecas de grasa lu		LUBRISA			18746.56	0	WILMER LE:	16/06/2009	07/07/2009	
11-095	SERVICIO ANUAL DE A	30	EDICIONES LEGALES	100	100	6540.8	0	WILMER LE:	19/06/2009	10/07/2009	24/08/2009
1200090018	GHFHFHFH	123	FHFGHRYTRYR	52.36	14.88	13440	11440	RTRYRYTY	24/08/2009	31/08/2009	24/08/2009
1200090000	JNKJN	20	454544	20	0.82	2003.38	2618.23	MNKJN	18/08/2009	31/08/2009	21/08/2009
120090010	FGHFHFHFH	30	GHJGHJ	50	44.64	1120	620	GFGFHHF	21/08/2009	31/08/2009	21/08/2009

Intranet.local 100%

Si no existen Contratos se mostrará el mensaje correspondiente:



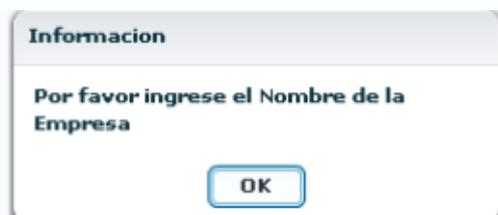
5. De lo contrario en la parte inferior se despliegan los contratos ingresados .

Nº Contrato	Objeto	Duración	Empresa Contratista	% Av.Físico	%Av.Econón	Monto	Saldo	Nom. Fiscal	Inicio	Fin	Fec. Cambio
0001	VBNVBNVN	88	NVB	25	22.32	1120	870	BVNVBN	21/08/2009	21/08/2009	24/08/2009
1-001	SUMINISTROS DE OFIC	1528	COGECOMSA	25.36	35.89	57120	36620.75	EDWIN MO	10/03/2009	04/11/2009	24/08/2009
10-090	300 canecas de grasa lu		LUBRISA			18746.56	0	WILMER LE	16/06/2009	07/07/2009	
11-095	SERVICIO ANUAL DE A	30	EDICIONES LEGALES	100	100	6540.8	0	WILMER LE	19/06/2009	10/07/2009	24/08/2009
120090018	GHFHFHFH	123	FHFGHRYTRYR	52.36	14.88	13440	11440	RTYRYRTY	24/08/2009	31/08/2009	24/08/2009
120090009	JNKNJ	20	454544	20	9.82	2903.38	2618.23	MNKJN	18/08/2009	31/08/2009	21/08/2009
120090010	FGHFHFHFH	30	GHJGHJ	50	44.64	1120	620	GFGFHFH	21/08/2009	31/08/2009	21/08/2009

7. Para ingresar un nuevo contrato presionar **NUEVO**



8. Ingresar todos los campos que se indican, si un campo no es ingresado aparecerá un mensaje como en siguiente.



9. Cuando haya ingresado todos los campos presionar el botón **Grabar**



En forma mensual se debe revisar y actualizar los contratos en lo que corresponde a cambios en avance físico y valor pagado a la fecha si existe un cambio.

Actualizar un Contrato

1. Seleccionar un Contrato del Grid.

Nº Contrato	Objeto	Duración	Empresa Contratista	% Av.Físico	%Av.Econón	Monto	Saldo	Nom. Fiscal	Inicio	Fin	Fec. Cambio
0001	VBNVBNVN	88	NVB	25	22.32	1120	870	BVNVBN	21/08/2009	21/08/2009	24/08/2009
1-001	SUMINISTROS DE OFIC	1528	COGECOMSA	25.36	35.89	57120	36620.75	EDWIN MO	10/03/2009	04/11/2009	24/08/2009
10-090	300 canecas de grasa lu		LUBRISA			18746.56	0	WILMER LE	16/06/2009	07/07/2009	
11-095	SERVICIO ANUAL DE A	30	EDICIONES LEGALES	100	100	6540.8	0	WILMER LE	19/06/2009	10/07/2009	24/08/2009
120090018	GHFHFHFH	123	FHFGHRYTRYR	52.36	14.88	13440	11440	RTYRYRTY	24/08/2009	31/08/2009	24/08/2009
120090009	JNKNJ	20	454544	20	9.82	2903.38	2618.23	MNKJN	18/08/2009	31/08/2009	21/08/2009
120090010	FGHFHFHFH	30	GHJGHJ	50	44.64	1120	620	GFGFHFH	21/08/2009	31/08/2009	21/08/2009

Al seleccionar un contrato los datos del grid se cargarán en los textos para ser modificados cuando presione el botón **Actualizar**.



Unidad: SISTEMAS

No. Contrato:	1-001	Empresa Contratista:	COGECOMSA		
Modalidad:	SUBASTA INVERSA E1	Fiscalizador/Responsable:	EDWIN MOSQUERA		
Estado:	EJECUCIÓN	Objeto del Contrato:	SUMINISTROS DE OFICINA		
Fecha Inicio:	10/03/2009	Año del Contrato:	2009		
Fecha Fin:	04/11/2009				
Duración (Días):	1528	Meses:	50	Días:	28

Monto:	51000.00	Monto con Iva:	57120	(%)Avance Físico:	25.36
(%)Anticipo:	0	Anticipo USD:	0.00	(%)Avance Económico:	35.89
Multas:	0	(%)Multas:	0.00		
Valor Pagado a la Fecha:	20496.25				
Saldo:	36620.75				

Observaciones: EWRWERWER

2. Realizar el cambio sobre los textos que desee.
3. Presionar **Grabar**.

Nota: Para cancelar las opciones Nuevo o Actualizar, presionamos el botón **Cancelar**.

Imprimir Anexos

1. Click sobre el botón Reporte.
2. Elegir el tipo de anexo: Anexo 1 o Anexo 2. Recuerde que en esta pantalla se desplegará solo la información del año seleccionado. Si desea un reporte de todos los años generar el reporte de la opción del Menú Consulta por Unidad.

Elegir Reporte

¿Que reporte desea imprimir:?

Anexo 1 **Anexo 2**

3. Si escogió el Anexo 1: se le mostrará el siguiente reporte:



GERENCIA ADMINISTRATIVA
FICHA DE CONTRATOS
UNIDAD : SISTEMAS

Fecha: 20/08/25
10:55:07

Usuario: Martha Janelly Ayala Buitrago

Año: 2009

No. CONTRATO	OBJETO DEL CONTRATO	EMPRESA CONTRATISTA (ADJUDICADA)	MODALIDAD DEL PROCESO DE CONTRATACION	MONTO USD	DURACION DIAS	FECHA INICIO	FECHA FIN	NOMBRE DEL FISCALIZADOR RESPONSABLE	FECHA CAMBIO
2007320	MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE IMPRESORAS LASER, IMPRESORAS DESKJET, SCANNERS Y PLOTTERS DE PETROECUADOR (MATRIZ) Y LA GERENCIA DE OLEODUCTO	AKROS	PCP	91671.35	30	02/12/2007	02/12/2009	ING. HENRY MARTINEZ	21/09/2009
2008006	SERVICIO DE ACCESO A INTERNET PARA PETROECUADOR	GLOBAL OROS SING COMUNICACIONES DEL ECUADOR	PCP	11982.16	30	02/09/2008	18/02/2009	ING. JAGO CHARRANZI	21/09/2009
2008073	CONCESION DEL USO DE FRECUENCIA RADIOELECTRICA PARA RED MOVIL	SENA TEL	TRAMITE EN SENATEL	675.10		29/01/2008	29/01/2013	ING. WILSON ARCOS G.	
A09-069	ACUERDO DE MANTENIMIENTO DE MOTORES AERONAUTA	ROLLS-ROYCE	PCP	421000.00	60	19/02/2009	18/02/2010	CPM/EMC (S.F) XAVIER RRA	21/09/2009
2009006	SERVICIOS DE SOPORTE LOGISTICO	TRC LOGISTICAS	PCP	800.00		06/03/2009	05/03/2010	CPM/EMC (S.F) XAVIER RRA	
2009203	SERVICIO DE HANDLING EN EL AEROPUERTO DEL COCA	HANDLING TOUR	COMPRAS PUBLICAS	11981.00		19/09/2009	18/09/2010	CPM/EMC (S.F) XAVIER RRA	
2009031	ACUERDO MARIL DE ENTRENAMIENTO RECURRENTE FLIGHT SAFETY DE PILOTADORES		PCP	384000.00	120	29/01/2009	29/01/2011	UNIDAD DE CAPACITACION	21/09/2009
2009050	CONCESION DEL USO DE FRECUENCIAS RADIOELECTRICAS PARA RED DE MCRONDONA	SENA TEL	TRAMITE EN SENATEL	9660.93		20/11/2006	20/11/2011	ING. WILSON ARCOS G.	

4. Si escogio Anexo 2:



GERENCIA ADMINISTRATIVA
SEGUIMIENTO DE CONTRATOS

Fecha: 20/09/2009
10:08:16

MES DE CONTROL:
FECHA DE ACTUALIZACIÓN:

Usuario: Martha Janelin Ayala Buitrago

Año: 2009

No. CONTRATO	OBJETO DEL CONTRATO	EMPRESA CONTRATISTA (R.D.U.D.R.A.)	% AVANCE FÍSICO	MONTO US\$	AVANCE ECONÓMICO			MULTAS	ESTADO	NOMBRE DEL FRECUENTADOR RESPONSABLE	OBSERVACIONES	FECHA CAMBIO
					ANTICIPO	VALOR PAGADO A LA FECHA	SALDO					
2007320	IMPLEMENTACIÓN DE SERVIDOR Y CORRECCIÓN DE ERRORES EN LOS SERVIDORES PARA EL SERVICIO Y ALISTAMIENTO DE SERVIDORES DENTRO DE LA GERENCIA DE INGENIERÍA	AKROS	79.17	91671.55	10.00	67435.20	24217.250	19	EJECUCIÓN	ING. HENRY MARTINEZ	EN EL MONEDARIO DE US\$ SE DEBE CORREGIR EL MONTO DEL CONTRATO A US\$ 24217.250.00 PORQUE EN LA ENTIDAD ADMINISTRATIVA LOCAL SE AUTORIZÓ EL MONTO DEL MONEDARIO DEL CONTRATO.	21/09/2009
2006000	SERVICIO DE ASesorIA INFORMÁTICA Y COMUNICACIONES DEL ECUADOR	GLOBAL CROSSING COMUNICACIONES DEL ECUADOR	99.93	41832.18	10.00	40172.18	1669.000	0	EJECUCIÓN	ING. HUGO CARRANZA	SE DEBE CORREGIR EL MONTO DEL CONTRATO A US\$ 40172.18 PORQUE EN LA ENTIDAD ADMINISTRATIVA LOCAL SE AUTORIZÓ EL MONTO DEL MONEDARIO DEL CONTRATO.	21/09/2009
2006073	CONCESIÓN DEL USO DE FOTOCOPIADORA PARA REDIMOR	SEINTEL	28.33	678.00	0	189.81	340.490	0	EJECUCIÓN	ING. WILSON ARCO S.		
AC08-080	ACUERDO DE COMPARTIMIENTO DE MOTORES AERONÁUTICA	BOLLO-BOYCE	0.42	4210.00.00	0	172359.85	245840.050	0	EJECUCIÓN	CPM/EMC (SP) XAVIER REA	VALOR DEL CONTRATO DE 421000.000	21/09/2009
2009-090	SERVICIO DE SUPORTE LOGÍSTICO	TIC LOGÍSTICOS	0.00	60000.00	0	2558.00	57442.000	0	EJECUCIÓN	CPM/EMC (SP) XAVIER REA		

PRUEBAS FUNCIONALES

SISTEMA GESTIÓN DE CONTRATOS PARA EPPETROECUADOR DESARROLLADO EN LA HERRAMIENTA WEB COLDFUSION MX7 PRUEBA FUNCIONAL

INGRESO DE CONCURSOS

Objetivo General

Registrar un concurso nuevo e ingresar tareas para realizar el seguimiento del concurso.

Objetivos Específicos

- Registro de Concurso: verificar que las validaciones a la información ingresada se ejecuten de forma correcta; validar el código del concurso y verificar que no se ingrese concurso con el mismo código.
- Registro de Tareas: validar que los cálculos automáticos de la duración del concurso de acuerdo al rango de fechas ingresadas sea el correcto y verificar que no permita el ingreso de tareas repetidas.

Ejecución de la Prueba

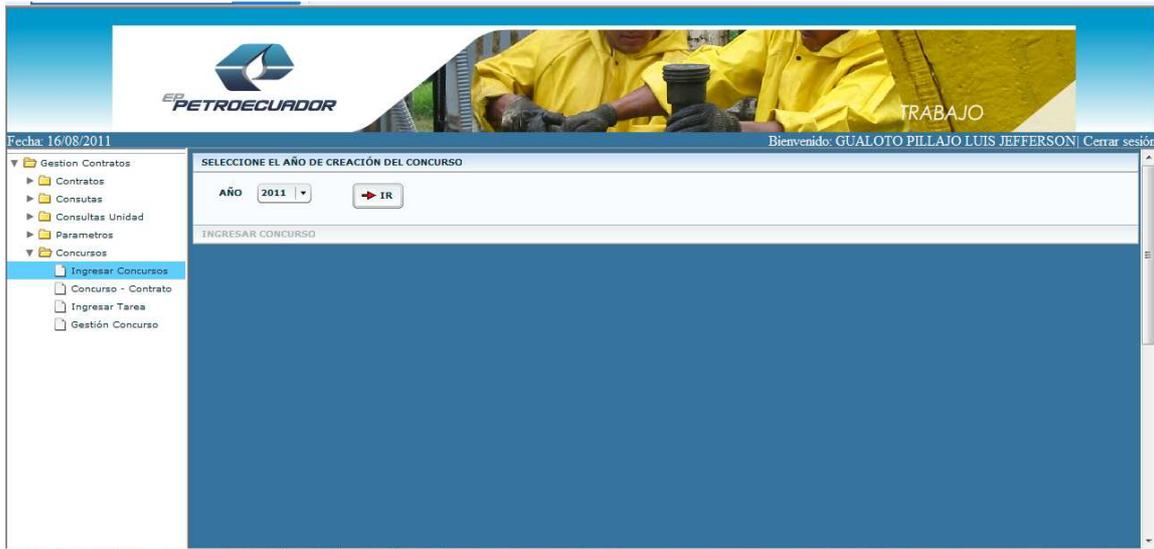
1. Registro de Concurso

1.1 Se ingresa un código de concurso que ya existe.

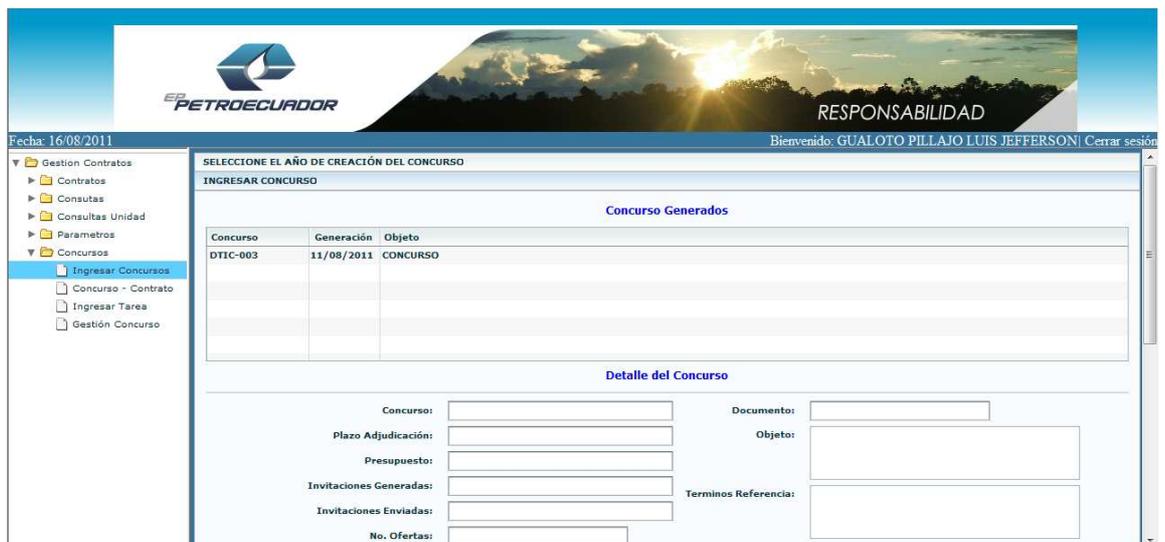
Pantallas de ejecución:

- Ingresar Concurso.

Se despliega la pantalla



- Seleccionar el año del concurso seguidamente presionar el botón *IR*.
- Si existen concursos generados para el año seleccionados los mostrará en el grid de concursos.



- Ingresar un nuevo concurso con un código existente e intentar guardarlo.

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

Detalle del Concurso

Concurso: DTIC-003 Documento: DTIC-DOC-003
 Plazo Adjudicación: 34 Objeto: CONCURSO
 Presupuesto: 1222 Terminos Referencia: CONCURSO
 Invitaciones Generadas: 12
 Invitaciones Enviadas: 12
 No. Ofertas: 11

Inicio Concurso: 11/08/2011 Respuestas: 13/08/2011 Apertura: 15/08/2011
 Preguntas: 12/08/2011 Ofertas: 14/08/2011 Adjudicación: 16/08/2011
 Evaluación Oferta: 18/08/2011

[NUEVO] [GRABAR] [ACTUALIZAR] [ELIMINAR] [CANCELAR]

- Mensaje de salida del sistema

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

Detalle del Concurso

Concurso: [] Documento: []
 Plazo Adjudicación: [] Objeto: []
 Presupuesto: [] Terminos Referencia: []
 Invitaciones Generadas: []
 Invitaciones Enviadas: []
 No. Ofertas: []

Inicio Concurso: [] Respuestas: [] Apertura: []
 Preguntas: [] Ofertas: [] Adjudicación: []
 Evaluación Oferta: []

[NUEVO] [GRABAR] [ACTUALIZAR] [ELIMINAR] [CANCELAR]

Información
 El código ingresado ya existe
 [OK]

Respuesta: el sistema despliega un mensaje informando que el número de concurso ingresado ya existe.

1.2 Ingresar un concurso nuevo

- Llenar los datos del formulario y grabarlos

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

Detalle del Concurso

Concurso: DTIC-004 Documento: DTIC-DOC-004

Plazo Adjudicación: 34 Objeto: CONCURSO

Presupuesto: 100000

Invitaciones Generadas: 12 Terminos Referencia: CONCURSO

Invitaciones Enviadas: 12

No. Ofertas: 11

Inicio Concurso: 10/08/2011 Respuestas: 13/08/2011 Apertura: 15/08/2011

Preguntas: 12/08/2011 Ofertas: 14/08/2011 Adjudicación: 16/08/2011

Evaluación Ofertas: 18/08/2011

- Mensaje de salida del sistema

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

Detalle del Concurso

Concurso: Documento:

Plazo Adjudicación: Objeto:

Presupuesto:

Invitaciones Generadas: Terminos Referencia:

Invitaciones Enviadas:

No. Ofertas:

Inicio Concurso: Respuestas: Apertura:

Preguntas: Ofertas: Adjudicación:

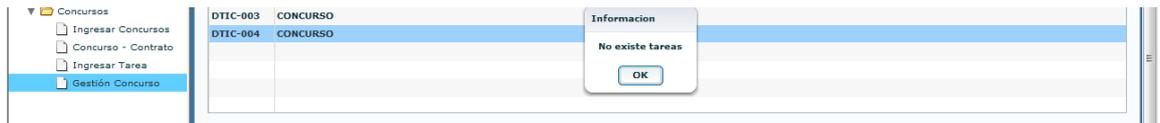
Evaluación Ofertas:

Información
Los Datos han sido Grabados

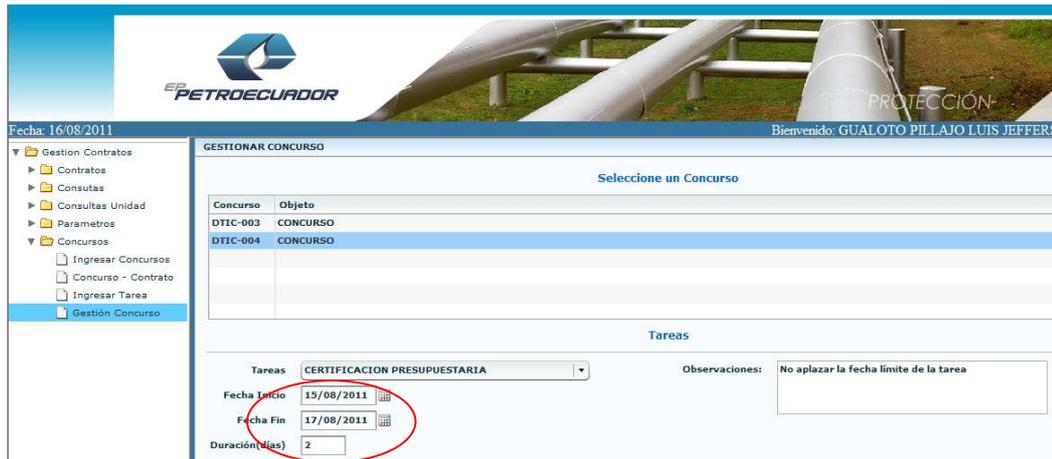
- **Respuesta:** el sistema despliega un mensaje indicando que los datos se guardaron correctamente

Pantallas de ejecución:

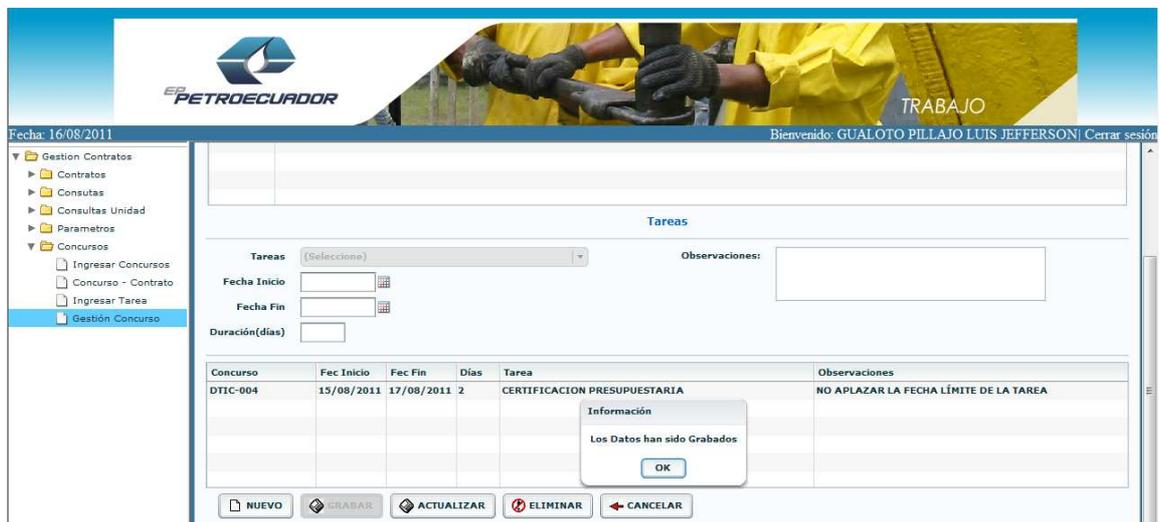
- Gestión Concurso.
- 2.1 Ingresar una tarea que ya se encuentra ingresada para un concurso.
- Seleccionar un concurso, si el concurso seleccionado no tiene tareas despliega el siguiente mensaje: "No existe tareas"



- Ingresar una tarea. Presionamos el botón **Nuevo** e ingresamos los datos.
- Se verifica que efectivamente el cálculo de días se está realizando correctamente.



- Presionamos **Grabar**.
- **Respuesta:** el sistema despliega un mensaje indicando que los datos se guardaron correctamente y la tarea ingresada lo carga en el grid.



- Para terminar con la prueba ingresamos la misma tarea al concurso en cuestión (DTIC-004).
- Seleccionamos el concurso (DTIC-004) e intentamos ingresar la misma tarea anterior.

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

DTIC-004 CONCURSO

Tareas

Tareas: CERTIFICACION PRESUPUESTARIA Observaciones: NO APLAZAR LA FECHA LÍMITE DE LA TAREA

Fecha Inicio: 15/08/2011

Fecha Fin: 17/08/2011

Duración(días): 2

Concurso	Fec Inicio	Fec Fin	Días	Tarea	Observaciones
DTIC-004	15/08/2011	17/08/2011	2	CERTIFICACION PRESUPUESTARIA	NO APLAZAR LA FECHA LÍMITE DE LA TAREA

NUEVO GRABAR ACTUALIZAR ELIMINAR CANCELAR

- **Respuesta:** el sistema despliega un mensaje indicando que la tarea que se está ingresando ya existe.

Fecha: 16/08/2011 Bienvenido: GUALOTO PILLAJO LUIS JEFFERSON | Cerrar sesión

DTIC-004 CONCURSO

Tareas

Tareas: CERTIFICACION PRESUPUESTARIA Observaciones:

Fecha Inicio:

Fecha Fin:

Duración(días):

Información

Tarea ya ingresada

OK

Concurso	Fec Inicio	Fec Fin	Días	Tarea	Observaciones
DTIC-004	15/08/2011	17/08/2011	2	CERTIFICACION PRESUPUESTARIA	NO APLAZAR LA FECHA LÍMITE DE LA TAREA

**ESCUELA DE FORMACION DE TECNOLOGOS
CARRERA DE: ANALISIS DE SISTEMAS INFORMATICOS**

ORDEN DE EMPASTADO

De acuerdo con lo estipulado en el artículo 83 del Reglamento del Sistema de Estudios de las Carreras de Formación Profesional y de Postgrados aprobado por el Consejo Politécnico en sesión del 16 de agosto de 2011 dictado por la Comisión de Docencia y Bienestar Estudiantil el 9 de y una vez verificado el cumplimiento del formato de presentación establecido, se autoriza la impresión y encuadernación final del Proyecto de Titulación presentado por el señor

LUIS JEFFERSON GUALOTO PILLAJO

Fecha de autorización: Quito, DM., 13 de agosto de 2012



Ingeniero Carlos Posso Játiva
DIRECTOR DE LA ESCUELA DE FORMACION DE TECNOLOGOS