

Universidad de las Ciencias Informáticas.
Facultad 9

UCI

**Título: Análisis y Diseño del módulo de
administración del Sistema Automatizado de
Control de Gestión de Indicadores de Refinación
(SACGIR).**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.**

Autor: Zibieta Texidó La O.

Tutor: Lic. David Silva Barrera.

Co tutor: Lic. Gretchen Guillermo Hernández.

Ciudad de la Habana, Julio 2006.

"Año 50 de la Revolución."

"A las estrellas no se llega por caminos llanos."

José Martí.

DEDICATORIA

A mi primo Elicer:

Por lo mucho que te hubiera gustado acompañarme en este momento.

Siempre estarás en mi corazón.

A mis padres:

Anacelis La O Sierra y Luis Antonio Texidó Sánchez.

Por su confianza en que este momento llegaría...

Y su amor durante el trayecto.

A la UCR, a Fidel y a la Revolución:

Por haberme transformado en una mujer de estos tiempos.

AGRADECIMIENTOS

A mi mamita, por su inmenso cariño, su confianza, su apoyo y sus consejos; por haber estado a cada momento a mi lado, aún en la distancia.

A mi papito, por ser el mejor padre del mundo, por quererme tanto y por haber estado a mi lado siempre, sacrificándote por mi, dándome todo lo que siempre necesite: Amor.

A mis hermanos, Dayelis, Tony, Nety, Jorge, Yuni y Yani, los quiero con todo mi corazón, gracias por quererme también.

A mi tía Elia, por ser mi madre durante estos cinco años.

A mi Jesnar, gracias por comprenderme, ayudarme y quererme.

A mis amigos, sin ustedes mi vida no tendría sentido.

A mis compañeros de aula y de proyecto.

A todo el que de una forma u otra colaboró para que se hiciera realidad mi sueño.

RESUMEN

En el presente trabajo se abordan aspectos vinculados a la realización del Análisis y Diseño de un Módulo de Administración que permita definir el control de acceso a la información y brinde facilidad para definir los indicadores, refinerías y nomencladores de forma dinámica, del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR) para la Gerencia de Planificación y Gestión de Refinación de PDVSA.

Como objetivo se persigue desarrollar los artefactos que componen los flujos de trabajo del análisis y diseño de una solución informática que realice todo el trabajo administrativo del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR). El desarrollo de este módulo de administración proporcionaría mejoras significativas en los procesos de gestión y un decremento del tiempo necesario para realizar estas tareas.

PALABRAS CLAVES

Análisis, Diseño, Administración, Sistema de Gestión, Indicadores.

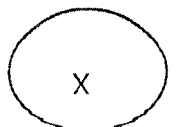
ÍNDICE DE TABLAS Y FIGURAS

FIGURA 1. FUNCIONES GENÉRICAS DE LA ADMINISTRACIÓN.....	18
FIGURA 2. ESQUEMA GENERAL DE LAS TECNOLOGÍAS WEB.....	26
FIGURA 3. ESTRUCTURA DE CAPAS DEL PATRÓN MVC APLICADO EN EL SISTEMA.....	28
TABLA 1. DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO.....	42
TABLA 2. DESCRIPCIÓN DE LOS TRABAJADORES DEL NEGOCIO.....	43
FIGURA 4. DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	43
FIGURA 5. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	54
TABLA 3. DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.....	55
TABLA 4. DESCRIPCIÓN EXTENDIDA DEL CUS GESTIONAR ROLES.....	59
TABLA 5. DESCRIPCIÓN EXTENDIDA DEL CUS GESTIONAR PERMISOS.....	62
TABLA 6. DESCRIPCIÓN EXTENDIDA DEL CUS GESTIONAR REFINERÍAS.....	67
TABLA 7. DESCRIPCIÓN EXTENDIDA DEL CUS GESTIONAR ROLES USUARIOS.....	69
TABLA 8. DESCRIPCIÓN EXTENDIDA DEL CUS GESTIONAR PLANTAS.....	73
FIGURA 6. DIAGRAMA DE CLASES DEL ANÁLISIS CU GESTIONAR PLANTA.....	75
FIGURA 7. DIAGRAMA DE CLASES DEL DISEÑO CU GESTIONAR PLANTA.....	75
TABLA 9. CLASE CP_PRINCIPAL.....	76
TABLA 10. CLASE CP_GESTIONARPLANTA.....	76
TABLA 11. CLASE FRM_GESTIONARPLANTA.....	77
TABLA 12. CLASE CP_INSERTARPLANTA.....	77
TABLA 13. CLASE FRM_INSERTARPLANTA.....	78
TABLA 14. CLASE CP_MODIFICARPLANTA.....	78
TABLA 15. CLASE FRM_MODIFICARPLANTA.....	79
TABLA 16. CLASE SP_GESTIONARPLANTA.....	79
TABLA 17. CLASE SP_INSERTARPLANTA.....	80
TABLA 18. CLASE SP_MODIFICARPLANTA.....	80
TABLA 19. CLASE SP_ELIMINARPLANTA.....	81
TABLA 20. CLASE GESTIONARPLANTASCONTROLLER.....	81
TABLA 21. CLASE CM_GESTIONARPLANTA.....	82
FIGURA 8. SACGIR: PLANTILLA GENERAL DE LA APLICACIÓN.....	83
FIGURA 9. DISEÑO DE INTERFAZ: AUTENTICAR USUARIO.....	83
FIGURA 10. SACGIR: INTERFAZ MÓDULO ADMINISTRACIÓN.....	84
FIGURA 11. DIAGRAMA ENTIDAD RELACIÓN.....	85
FIGURA 12. DIAGRAMA DE CLASES PERSISTENTES.....	86

ÍNDICE

INTRODUCCIÓN	11
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	14
1.1. INTRODUCCIÓN.....	14
1.2. CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	14
1.3. EL PROCESO DE ADMINISTRACIÓN DE LA INFORMACIÓN UTILIZADA EN LA GERENCIA DE PLANIFICACIÓN Y GESTIÓN DE REFINACIÓN DE PDVSA Y EL CONTROL DE ACCESO A LOS DATOS.	17
1.3.1. DESCRIPCIÓN GENERAL.....	17
1.3.2. DESCRIPCIÓN ACTUAL DEL DOMINIO DEL PROBLEMA.	18
1.3.3. SITUACIÓN PROBLEMÁTICA.....	19
1.4. ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES.	21
1.5. CONCLUSIONES.....	23
CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A DESARROLLAR.....	25
2.2.1. APLICACIONES WEB.	25
2.2.2. MODELO CLIENTE - SERVIDOR.	27
2.2.2.1.PATRÓN MODELO – VISTA – CONTROLADOR.....	27
2.3. LENGUAJES DE PROGRAMACIÓN.....	29
2.3.1. HTML.....	29
2.3.2. JAVA SCRIPT.....	29
2.3.3. PHP.....	30
2.3.3.1.FRAMEWORK CAKEPHP.	32
2.4. SERVIDORES WEB.	33
2.4.1. APACHE.....	33
2.5. SISTEMA DE GESTIÓN DE BASES DE DATOS (SGBD).....	34
2.5.1. POSTGRESQL.....	35
2.6. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	36
2.6.1. FUNDAMENTACIÓN DE LA METODOLOGÍA UTILIZADA.....	37
2.6.1.1.LENGUAJE UNIFICADO DE MODELADO.....	38
2.7. HERRAMIENTAS CASE DE DESARROLLO DE SOFTWARE.	39
2.7.1. VISUAL PARADIGM.....	39
2.8. CONCLUSIONES.....	40
CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	41
3.1. INTRODUCCIÓN.....	41
3.2. MODELO DE NEGOCIO.	41
3.2.1. ACTORES Y TRABAJADORES DEL NEGOCIO.....	41
3.2.2. DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	43
3.2.3. PROCESOS Y CASOS DE USO DEL NEGOCIO.	44
3.2.3.1.PROCESO DE ELABORACIÓN DE MARGEN BRUTO REAL.....	44
3.2.3.2.PARAIDA DE PLANTA.	45
3.2.3.3.INDICADORES DE SEGURIDAD INDUSTRIAL.....	46
3.2.3.4.RECEPCIONAR, CONSOLIDAR, EMITIR, CERTIFICAR Y RECTIFICAR.	46
3.2.3.5.GESTIÓN DE RESULTADOS MENSUALES Y EVALUACIÓN DE INDICADORES.....	48
3.3. REQUERIMIENTOS FUNCIONALES.	49

3.4. REQUERIMIENTOS NO FUNCIONALES.....	52
3.5. DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	54
3.5.1. DESCRIPCIÓN DE LOS ACTORES.....	55
3.5.2. DESCRIPCIONES DE LOS CASOS DE USO DEL SISTEMA.....	55
3.6. CONCLUSIONES.....	73
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	74
4.1. INTRODUCCIÓN.....	74
4.2. DIAGRAMAS DE CLASES.....	74
4.2.1. DIAGRAMA DE CLASES DEL ANÁLISIS DEL CUS GESTIONAR PLANTAS.....	75
4.2.2 DIAGRAMA DE CLASES DEL DISEÑO DEL CUS GESTIONAR PLANTAS.....	75
4.3. PRINCIPIOS DE DISEÑO.....	82
4.3.1. ESTÁNDARES DE LA INTERFAZ DE LA APLICACIÓN.....	82
4.4. DISEÑO DE LA BASE DE DATOS.....	84
4.4.1. DIAGRAMA ENTIDAD RELACIÓN.....	84
4.4.2. DIAGRAMA DE CLASES PERSISTENTES.....	86
4.5. DIAGRAMA DE DESPLIEGUE.....	86
4.6. CONCLUSIONES.....	88
CONCLUSIONES GENERALES	89
RECOMENDACIONES	90
TRABAJOS CITADOS	91
BIBLIOGRAFÍA	94
ANEXOS	97
GLOSARIO DE TÉRMINOS	98



INTRODUCCIÓN

Desde la antigüedad el petróleo aparecía de forma natural en ciertas regiones continentales como las que hoy ocupan los países de Oriente Medio. Los asirios y babilonios lo usaban para pegar ladrillos y piedras; los egipcios, para engrasar pieles, hace 6.000 años atrás. En ese entonces, era utilizado con fines medicinales, en el embalsamamiento, para aceitar las ruedas de los carruajes y para engrasar pieles.

Sin embargo, la explotación de este recurso no llegaría sino hasta el año 100 antes de Cristo, cuando los chinos, de una manera bastante rústica extrajeron petróleo de la tierra. A partir de 1850 comienza una etapa que se va a caracterizar por la comercialización, debido al descubrimiento de valiosas aplicaciones para este recurso natural, entre las que se encuentran muchas de las más útiles hoy: el asfalto para la creación de caminos; la parafina para la fabricación de velas, el recubrimiento de quesos y pomadas para calzado; el kerosén y sus derivados refinados, por su parte, utilizado en la calefacción, la iluminación y combustible para el transporte. (1)

Los inicios de la industria petrolera moderna estuvieron ligados fundamentalmente a la iluminación doméstica y urbana y la calefacción. Fue durante el siglo XX, con el desarrollo del motor de combustión interna, que el petróleo adquiere verdadera importancia para la producción industrial misma y el transporte. (1)

La actividad petrolera venezolana se inicia pues con la creación, en 1878, de la Compañía Nacional Minera Petrolera del Táchira, cuya producción apenas alcanzaba para dotar de kerosén a las ciudades vecinas. Desde sus inicios la explotación del petróleo estuvo caracterizada por la penetración de compañías y monopolios extranjeros, quienes en los años posteriores con importantes descubrimientos de yacimientos de hidrocarburo y una exportación intensa permitieron desde 1928 hasta 1970 Venezuela se convirtiera en el primer país exportador de combustible fósil del mundo.

En el contexto de los años 70 del pasado siglo XX se producen una serie de hechos geopolíticos de gran trascendencia internacional, lo que viene a favorecer una nueva orientación en la política petrolera que tuvo como conclusión la promulgación de un conjunto de reformas y leyes que finalmente resultó en la creación de Petróleos de Venezuela, S.A. (PDVSA) por decreto del Ejecutivo Nacional el 30 de agosto de 1975. No obstante a la creación de la nueva empresa responsabilizada con el diseño de políticas nacionales en materia petrolera producto de un proceso de nacionalización, en realidad todo quedó en manos de los mismos que habían dominado el negocio petrolero antes de su creación quienes se encargaron de propiciar las condiciones para una nueva entrega del petróleo a los intereses no nacionales.

Como producto de la Revolución Bolivariana de Venezuela y sus profundos cambios sociales, la Constitución de la República establece que la totalidad de las acciones de Petróleos de Venezuela

Introducción

S.A. pertenecen al Estado Venezolano, en razón de la estrategia nacional y la soberanía económica y política, ejercida por el pueblo venezolano.

La refinación es el proceso que se encarga de la transformación de los hidrocarburos en productos derivados. PDVSA realiza sus operaciones de procesamiento del crudo a través de 22 refinerías, lo que la convierte en la tercera empresa en el proceso de refinación en el mundo, con una capacidad de procesamiento de petróleo de 3.3 millones de barriles por día. (1)

La Gerencia de Planificación y Gestión de Refinación de PDVSA esta dividida en dos niveles, refinerías y corporativo cada uno de ellos compuestos por los analistas de Gestión, superintendentes y gerentes vinculados directamente con la producción.

En la actualidad el Sistema de Refinación de PDVSA tiene en explotación varios sistemas, que apoyan el proceso de refinación y la gestión de datos, como son: Sistema de Información de Manufactura (SIM) que consiste en el soporte de datos de gestión de refinación y el Sistema de Información de Movimiento del Petróleo (SIMP) que soporta las operaciones diarias en las refinerías.

Luego del paro petrolero cometido contra PDVSA entre diciembre del año 2002 y enero de 2003 por la oposición al gobierno bolivariano, estos sistemas sufrieron daños que ocasionaron un mal funcionamiento de los mismos, por lo que ya no cumplen totalmente con los objetivos para los que fueron implantados.

Lo anteriormente expresado denota la necesidad de una herramienta informática más eficiente que le brinde, a la Gerencia de Planificación y Gestión de Refinación, un soporte automatizado a los procesos y a su vez permita definir el control de acceso a la información, ofreciendo además facilidad para definir los indicadores, las refinerías y los nomencladores de forma dinámica.

Teniendo en cuenta lo anterior la principal tarea de este trabajo estaría encaminada a resolver el siguiente **problema**:

Necesidad de un Módulo de Administración que permita definir el control de acceso a la información y brinde facilidad para definir los indicadores, refinerías y nomencladores de forma dinámica, del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR) para la Gerencia de Planificación y Gestión de Refinación de PDVSA.

A partir del problema planteado se tiene como perspectiva el diseño de una herramienta informática que le permita a la Gerencia de Planificación y Gestión de Refinación mejoras significativas en los procesos de gestión y un decremento del tiempo necesario para realizar estas tareas actualmente.

Tomando como punto de partida lo anterior se plantea como **objeto de estudio** los procesos de administración de la información utilizada en la Gerencia de Planificación y Gestión de Refinación de PDVSA y el control de acceso a los datos. Siendo de esta manera, el **campo de acción**, la automatización de los procesos de administración de la información utilizada en la Gerencia de Planificación y Gestión de Refinación de PDVSA y del control de acceso a los datos.

Introducción

Se plantea como **objetivo general** realizar el análisis y diseño del Módulo de Administración de SACGIR para la Gerencia de Planificación y Gestión de Refinación de PDVSA, que permita definir el control de acceso a la información y brinde facilidades para definir los indicadores, refinerías y nomencladores de manera dinámica.

Para dar cumplimiento al objetivo general se trazan los siguientes **objetivos específicos**:

- Determinar el flujo de información relacionado con la gestión de indicadores.
- Establecer las modificaciones necesarias al flujo de información relacionado con la gestión de indicadores logrando su óptimo desempeño.
- Realizar el análisis y diseño del módulo.

Para llevar a cabo la investigación se plantean las siguientes **tareas**:

1. Análisis de la documentación y de los módulos existentes del sistema SACGIR.
2. Investigación de sistemas de Gestión de Información y las tecnologías que sustentan la existencia de aplicaciones informáticas similares.
3. Investigación de la gestión de indicadores.
4. Elaboración del Análisis y Diseño del Módulo de Administración de SACGIR.
5. Desarrollo de un prototipo no funcional del Módulo de Administración que represente las principales funcionalidades del sistema.

El documento está estructurado en cuatro capítulos:

Capítulo 1 "Fundamentación Teórica". En el mismo se incluyen todos los aspectos teóricos que soportan el desarrollo de esta aplicación, así como una descripción de los procesos vinculados al objeto de estudio. Incluye además el análisis de algunos de los sistemas que se dedican a la gestión de información de los procesos de refinación.

Capítulo 2 "Tendencias y tecnologías actuales a desarrollar". Este capítulo resume un análisis descriptivo de las tecnologías y tendencias actuales en las cuales se basa el trabajo, profundizándose en el conocimiento de conceptos necesarios para su comprensión, y exponiendo sobre todo ventajas que llevaron a elegir las herramientas y lenguajes utilizados en la construcción de este Software de Gestión.

Capítulo 3 "Presentación de la solución propuesta". En este capítulo se describen los procesos bajo estudio, determinando reglas del negocio, requisitos funcionales y no funcionales y por último se definen los casos de uso del sistema.

Capítulo 4 "Construcción de la solución propuesta". En este capítulo presentan los diagramas de clases, tanto del análisis como del diseño. Así como los diagramas de interacción y las pautas de diseño de la interfaz de la aplicación, se presentan los diagramas de Entidad Relación y de Clases persistentes, utilizados en el diseño de la Base de Datos de la aplicación.

CAPÍTULO 1

Fundamentación Teórica

1.1. Introducción.

En este capítulo se abordan los aspectos teóricos vinculados al desarrollo de un Módulo de Administración que permita definir el control de acceso a la información y brinde facilidad para definir los indicadores, refinorías y nomencladores de forma dinámica, del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR) para la Gerencia de Planificación y Gestión de Refinación de PDVSA. En vista a lograr un mejor entendimiento del tema, por aquellas personas que lean el trabajo, se exponen los principales conceptos relacionados con el dominio del problema.

Se describen los procesos de administración de la información utilizada por la Gerencia de Planificación y Control de Refinación de PDVSA y el control de acceso a datos, haciendo una valoración del estado actual del entorno donde coexiste el negocio y la situación problemática que rodea a la organización.

Por último se hace un análisis de otros sistemas que se dedican a la gestión de información, específicamente, algunos de los que actualmente utiliza el cliente, para tener una mejor idea del estado del arte a nivel internacional de los Sistemas Automatizados que se dedican a la gestión de los procesos de refinación.

1.2. Conceptos asociados al dominio del problema.

Indicador

Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos, en la ejecución de un proyecto, programa o actividad. Resultado cuantitativo de comparar dos variables. Se mide en porcentajes, tasas y razones para permitir comparaciones.

El término "Indicador" en el lenguaje común, se refiere a datos esencialmente cuantitativos, que nos permiten darnos cuenta de cómo se encuentran las cosas en relación con algún aspecto de la realidad que nos interesa conocer. Los Indicadores pueden ser números, hechos, medidas, opiniones o percepciones que señalen condiciones o situaciones específicas. (2)

En los indicadores deberán reflejarse adecuadamente la naturaleza, peculiaridades y nexos de los procesos que se originan en la actividad económica – productiva, sus resultados, gastos, entre otros, y caracterizarse por ser estables y comprensibles, por tanto, no es suficiente con uno solo de ellos para medir la gestión de la empresa sino que se impone la necesidad de considerar los sistemas de

Capítulo 1 Fundamentación Teórica

indicadores, es decir, un conjunto interrelacionado de ellos que abarque la mayor cantidad posible de magnitudes a medir. (2)

El sistema de indicadores debe caracterizar el nivel técnico - organizativo de desarrollo de la empresa, los recursos que posee y los resultados generales de la actividad productiva con una alta calidad. (2)

Los indicadores que se miden con el sistema que SACGIR son los siguientes:

- Volumetría (Crudos, Insumos y Productos).
- Exportación de productos.
- Costo Refinación.
- Indicadores SHA (Seguridad Industrial).
- Margen de Refino.

Refinación

La refinación es el proceso que se encarga de la transformación de los hidrocarburos en productos derivados. PDVSA realiza sus operaciones de procesamiento del crudo a través de 22 refinerías: tres complejos en Venezuela, y diecinueve en el resto del mundo. (3)

Del petróleo se obtienen muchos productos, desde gases y líquidos sumamente volátiles como la gasolina, hasta fluidos muy espesos como el asfalto y aun sólidos como la parafina o ceras. En líneas generales, los derivados básicos del petróleo son: gases, gasolina de motor, gasolina de aviación, kerosén, gasoil, diesel, solventes, bases lubricantes, parafina, combustible pesado (fuel oil) y asfalto. (3)

Además de esos productos básicos elaborados en instalaciones propias, la industria suministra materias primas a plantas petroquímicas y empresas manufactureras para producir caucho sintético, fibras sintéticas, fertilizantes, explosivos, insecticidas, medicinas, artículos de tocador y miles de otros productos.

PDVSA es la tercera empresa en el proceso de refinación en el mundo, con una capacidad de procesamiento de petróleo de 3.3 millones de barriles por día. (3)

Gerencia de Planificación y Gestión de Refinación de PDVSA

Es la gerencia de PDVSA que se encarga de la gestión de los resultados de refinación y de los procesos de control y planificación de la producción.

PDVSA

Petróleos de Venezuela S.A. es la corporación estatal de la República Bolivariana de Venezuela que se encarga de la exploración, producción, manufactura, transporte y mercadeo de los hidrocarburos, de manera eficiente, rentable, segura, transparente y comprometida con la protección ambiental; con el fin último de motorizar el desarrollo armónico del país, afianzar el uso soberano de los recursos, potenciar el desarrollo endógeno y propiciar una existencia digna y provechosa para el pueblo

Capítulo 1 Fundamentación Teórica

venezolano, propietario de la riqueza del subsuelo nacional y único dueño de esta empresa operadora.
(4)

Refinería

Una refinería es una unidad industrial, que puede estar compuesta por varias plantas, destinada a la refinación de petróleo la cual, mediante un proceso adecuado, obtiene diversos combustibles fósiles capaces de ser utilizados en motores de combustión: gasolina, gasóleo, etc. Adicionalmente, y como parte natural del proceso, obtiene diversos productos tales como aceites minerales y asfaltos.

En el territorio nacional, Petróleos de Venezuela cuenta con la refinería más grande del mundo: el Complejo Refinador Paraguaná (CRP), compuesto por las plantas de Amuay, Cardón y Bajo Grande. Ubicado en la península del mismo nombre, en el estado Falcón, este complejo tiene una capacidad de refinación de 940 mil barriles diarios (MBD). (3)

También se encuentran la refinería de Puerto La Cruz, con una capacidad de procesamiento de 200 MBD, y El Palito, que refina 130 mil barriles diarios. (3)

Fuera de sus fronteras, PDVSA cuenta con ocho refinerías en Estados Unidos, específicamente la de Corpus Christi, en Houston (Texas); Lemont, en Illinois; Paulsboro en New Jersey; Sweeny, Virginia; Lake Charles y Chalmette en Louisiana, y Savannah, en Georgia. La capacidad de refinación en suelo norteamericano es de aproximadamente 1.116 MBD. (3)

En Europa PDVSA procesa 252 mil barriles diarios, gracias a la asociación con AB Nynäs y a través de las refinerías de Nynasshamn en Suecia y Antwerp en Bélgica, así como de las ubicadas en el Reino Unido (específicamente en Dundee, Escocia, y Eastham, en Inglaterra). (3)

Además, PDVSA cuenta con una capacidad de procesamiento de 608 mil barriles diarios en las refinerías ubicadas en el Caribe, específicamente el caso de Isla, en Curazao; que procesa 335 MBD y St. Croix (Bahamas), con 273 mil barriles por día. (3)

Usuario

Es aquella persona a la que después de realizar la solicitud de acceso se le ha concedido la autorización para el uso de un conjunto de permisos y de recursos (o dispositivos) del sistema.

Traza

Una traza es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.

Este sistema trabaja con trazas para guardar incidencias, errores, accesos de usuarios, etc.

A través de las trazas se puede encontrar información para detectar posibles problemas en caso de que no funcione algún sistema como debiera o se haya producido una incidencia de seguridad.

Rol

Un rol es un conjunto de permisos que puede asignarse a un usuario del sistema; por su parte, un permiso es un parámetro que especifica si su poseedor dispone de acceso a una determinada función

del sistema o a una parte de la interfaz de usuario del sistema. Normalmente, los roles se definen de modo que incluyan permisos que guarden cierta relación y suelen corresponderse con algún rol de la vida real. Por ejemplo, para este sistema se define un rol Administrador para que incluya los permisos relacionados con tal función, como el de poder gestionar usuarios (Administradores), lo cual permite realizar acciones de inserción, eliminación y modificación de los mismos. Además se encarga de gestionar los roles de los usuarios que interactuarán con el sistema, así como los permisos que se corresponden con dichos roles. Es su responsabilidad asignarle los diferentes roles definidos a los usuarios del sistema.

1.3. El proceso de administración de la información utilizada en la Gerencia de Planificación y Gestión de Refinación de PDVSA y el control de acceso a los datos.

1.3.1. Descripción general.

Uno de los principios fundamentales para la toma adecuada de decisiones y la calidad de las mismas es basarse en hechos y datos que nos brinden información precisa antes de la ejecución de los planes, durante estos y después. El concepto general de la administración, implica como sus acciones básicas: planear, organizar, dirigir y controlar.

La planeación se refiere a determinar adónde debe ir la organización y cómo se debe llegar allí. Esto requiere la evaluación de los cambios y restricciones internas y externas, pronóstico, fijación de objetivos, desarrollo de estrategias y políticas y preparación de planes de acción. (5)

La organización tiene que ver con decidir quién hace qué. Esto requiere la definición y agrupación de actividades, la asignación de responsabilidades y el establecimiento de medios de comunicación, coordinación y control. (5)

La dirección consiste en determinar qué se debe hacer, y propiciar que todos trabajen en estrecha colaboración brindando lo mejor de sus capacidades. (5)

En cuanto al control se refiere, éste consiste en medir y supervisar los resultados, comparar las mediciones con los planes y, cuando sea necesario, tomar las medidas correctivas. (5)

Se debe considerar el control como la acción de garantizar que un proceso, una acción o en general una situación se mantendrá estable, en el sentido de que las variables o factores claves o vitales se moverán dentro de un rango previamente establecido y deseable, teniendo en cuenta el hecho de que el proceso, la acción o la situación misma se encarga de realizar por sí misma el monitoreo de las variables y el correspondiente ajuste, en caso de que se presenten desviaciones frente a lo establecido o lo esperado. (5)

Tal como se aprecia en la figura 1, desde la fase de planeación se va definiendo y ejerciendo el control, a través de la determinación de las variables clave de éxito asociadas a los objetivos y a los recursos disponibles. Es el control, mediante la generación de la información, el que provee los

insumos para la adecuada y oportuna toma de decisiones, de preferencia preventivas, que mantienen a la organización en la senda del éxito.

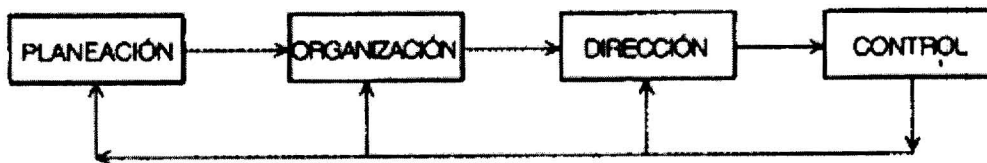


Figura 1 Funciones genéricas de la administración.

Basándose en las acciones básicas de la administración se identificaron los siguientes procesos de administración de la información y control de acceso a datos a desplegar para el desarrollo del módulo de administración para el sistema SACGIR que necesita la Gerencia de Planificación de PDVSA.

Se puede señalar el desarrollo de los procesos de administración de la información referente a los indicadores, plantas, refinerías y nomencladores, de forma dinámica. A través de estos procesos se gestiona la inserción, modificación y eliminación de todos los datos referentes a estas entidades. Aparejado a estos están los procesos de control de acceso a los datos, que implementando las acciones básicas de la administración determinan la organización, al decidir quien hace que; dentro de estos está el proceso que se encarga de especificar los roles, es decir definir los niveles de acceso a la información dentro del sistema y el procesos mediante el que se definen los roles de usuarios, o sea, definir que permisos va a tener asignado cada usuario dentro del sistema, que acciones puede realizar y cuales no. Se tiene también el proceso que define los permisos dentro del sistema, estos permisos son asignados a los usuarios, como una forma de restringir quien puede hacer cada cosa, evidenciando el uso de la dirección, otra de las acciones básicas de la administración. Además se establece un proceso que se encarga de supervisar las acciones de los usuarios sobre el sistema, mediante el uso de las trazas que va dejando el usuario tras cada una de las acciones que se realizan sobre el sistema. Tanto la especificación de los roles, los permisos, como el control de las trazas, ayudan a establecer un mejor control sobre la información que maneja el sistema.

1.3.2. Descripción actual del dominio del problema.

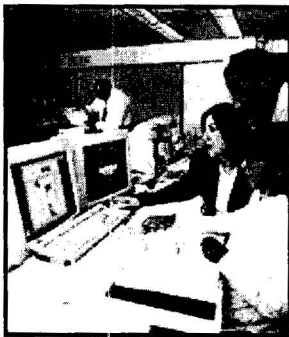
La Gerencia de Planificación y Gestión de Refinación es la encargada de llevar las estadísticas y el análisis del cumplimiento de los planes de producción mensual de las Refinerías. Para llevar a cabo las tareas mencionadas se realizan dos grandes procesos: la gestión de la información desde las Refinerías hasta la gerencia corporativa y la evaluación de la información a través del cálculo de indicadores. El apoyo a estos procesos con herramientas automatizadas es decisivo, pues implica ajustes de tiempo así como seguridad, precisión y oportunidad de la información.

El proceso de gestión tiene como objetivo importar, revisar y certificar los resultados de refinación. Mientras que el proceso de evaluación permite conocer el comportamiento de un indicador en el

período y compararlo con períodos anteriores para evaluar su similar comportamiento o no y de ser necesario investigar las causas. La evaluación del resultado del indicador se realiza a través de la revisión de la información llegada de las refinerías.

Las soluciones informáticas, utilizadas en el proceso de Gestión, son Sistemas Proprietarios y no abarcan toda la gama de información, ni las herramientas necesarias para el control del proceso de Refinación, además no tienen en cuenta el proceso de Evaluación. Actualmente este último proceso se realiza de forma semiautomática, no se tiene un control de versiones, al estar la mayor parte de la información en hojas de calculo Excel, cualquier usuario puede modificar datos sin que otro tenga conocimiento de los cambios realizados o utilizar un documento que no es la ultima versión, además esta información se transmite vía correo electrónico o mediante la red, sin que exista una total confidencialidad de la misma. La aparición de la mano del hombre, en el proceso, propicia la introducción de errores y ocasiona que los tiempos de respuestas a las demandas de información sean elevados. Además ha originado el uso de distintas plantillas de datos.

1.3.3. Situación Problemática.



El paro petrolero cometido contra la industria petrolera venezolana PDVSA entre diciembre del 2002 y enero del 2003, ocasionó pérdidas económicas valoradas en 14.430 millones de dólares, por concepto de ventas no realizadas. (6)

La participación de INTESA ,compañía venezolana, encargada de proveer a PDVSA los servicios de tecnología de información requeridos , en el colapso provocado a PDVSA fue determinante, considerando que esta empresa desde 1997 controlaba toda la infraestructura, facilidades, equipos, data financiera, geológica, técnica, presupuestaria y de negocios de la empresa y, además, el personal de soporte tecnológico de información de PDVSA, en el cual reposa el más decisivo control de la producción, refinación y distribución del crudo y sus derivados. (7)

INTESA ejerció su poder de control computarizado para paralizar la carga, descarga y almacenamiento de crudo en los diferentes terminales de embarque, así como para detener el funcionamiento de la mayoría de las estaciones de flujo, plantas compresoras, plantas de procesamiento, llenaderos de combustible, tuberías automatizadas, etc., asegurándose que la manipulación de las redes informáticas fuera posible solamente por parte de los poseedores de las claves secretas de acceso al sistema, todos empleados de INTESA sumados al "paro", quienes secuestraron el sistema mediante un acceso clandestino preparado cuidadosamente con anticipación. (7)

Este contundente sabotaje informático implicó la utilización de módems ocultos en el interior de paredes y escritorios para operar con acceso remoto, vía telefónica. Cabe anotar el uso de la red INTERNET para lograr acceso a la intranet de la corporación, en donde tanto las comunicaciones

Capítulo 1 Fundamentación Teórica

satelitales como de microondas y radio son parte del sistema. Toda esta plataforma tecnológica sirvió a los fines del sabotaje: impedir el control de las operaciones por parte de quienes siguieron laborando, aislar Occidente con respecto a las oficinas centrales de PDVSA en Caracas, destruir la bases de datos contentivas de las operaciones rutinarias y evitar la identificación de operadores en distintas instalaciones y maquinarias; eventualmente, suprimir los sistemas operativos de los servidores, desconfigurar los routers (dispositivo encargado de encaminar y transmitir paquetes de información entre diferentes redes informáticas) y finalmente, dismantelar todo el sistema de información y control a distancia de las operaciones automatizadas. Esto condujo al personal que continuó en sus puestos de trabajo a una incertidumbre total en torno a los valores reales establecidos en los puntos de control de los procesos, y obligó a presumir un excesivo riesgo en el manejo de las instalaciones que les indujo a proceder a la verificación pormenorizada de los parámetros de control de plantas y equipos, antes de adelantar el arranque de los mismos; como consecuencia, el reinicio de las operaciones en este caso fue muy lento y minucioso. (7)

El Sistema de Información de Manufactura (SIM), sistema que apoya los procesos de refinación y la gestión de datos, no estuvo exento de daños durante el vergonzoso sabotaje tecnológico. Producto de todo este caos actualmente las interfaces del SIM están inactivas en todas las refinerías, exceptuando el Complejo de Refinerías Paraguaná (CRP), el resto de las refinerías realizan los procesos de gestión de la información de forma manual, utilizando hojas de cálculo Excel, trayendo consigo errores en cifras y tratamiento de los datos, aumentando así el tiempo necesario para la gestión.

Sumado a todo esto está el hecho de que PDVSA posee actualmente una amplia y heterogénea plataforma de software la cual por sus dimensiones y características se encuentra expuesta a diversos factores que progresivamente afectan su efectividad y eficiencia. Dentro de esos factores se pueden mencionar entre otros: obsolescencia, pérdida de uniformidad en las versiones, componentes sin soporte por parte del proveedor, aplicaciones o sistemas que no están siendo usados o sin usuarios definidos. En razón de lo anterior, se plantea la necesidad de diseñar una estrategia orientada a la optimización de la plataforma tecnológica. (8)

Se puede agregar además que Petróleos de Venezuela (PDVSA) avanza en la migración de su plataforma tecnológica a software libre, en cumplimiento de los lineamientos establecidos por el Gobierno Nacional, en el decreto presidencial 3390. Esta resolución, de fecha de 23 de febrero de 2004, orienta a todas las empresas gubernamentales venezolanas a migrar sus plataformas tecnológicas al software libre y al empleo de estándares abiertos. (9)

La decisión de utilizar software libre, que no genera costos por adquisición y uso, estimula la democratización del conocimiento y conduce hacia la independencia tecnológica, permitiría que PDVSA se ahorrara buena parte de los 160 MM US\$ de inversión anuales en licencias de software. (10)

Capítulo 1 Fundamentación Teórica

Todo lo abordado anteriormente lleva a la conclusión de que la Gerencia de Planificación de PDVSA necesita un sistema que le suministre facilidades para gestionar la información de forma dinámica, que le brinde seguridad en el manejo de los datos y que este desarrollado utilizando software libre.

1.4. Análisis de otras soluciones existentes.

En el mercado de hoy en día, existen varias empresas dedicadas al desarrollo de aplicaciones informáticas de gestión de información, aunque la mayoría de los productos que elaboran utilizan software propietario muy costoso. A continuación se analizan algunas soluciones que se dedican específicamente a la gestión de información de los procesos de refinación.

El Sistema de Refinación de PDVSA cuenta con una variedad de sistemas informáticos que apoyan el proceso de refinación y a su vez alimentan la gestión de datos para los entes interesados, entre estos sistemas están: SIM (soporte de datos de gestión de Refinación), SIMP (soporte de operaciones diarias en las Refinerías), STARS (operaciones de buques en puertos), PIMS (planificación de producción); con todos ellos se necesita la integración con el objetivo de aumentar el nivel de automatización. No obstante el proceso de Refinación no está totalmente cubierto por esos sistemas, dando espacio al procesamiento de datos de forma manual y a los inconvenientes que esto conlleva.

Seguidamente se analizan dos de estos productos informáticos.

SISTEMA DE INFORMACIÓN DE MANUFACTURA (SIM)

El Sistema de Información de Manufactura (SIM), tiene como objetivo gestionar la información relevante de los resultados de producción de las refinerías de PDVSA, apoyando la cadena de valor del Negocio Refinación, mediante el uso de una base de datos única que permite mantener la uniformidad, confiabilidad y consistencia de la información que suministran las refinerías, requerida para la toma de decisiones en la corporación.

El SIM consolida los resultados operacionales, financieros y de personal para la gestión de las refinerías del circuito Refinador Venezuela – Isla, necesarios para informar a la junta directiva de Refinación, Suministro y Comercio y emitir reportes al MEM.

El SIM lo conforman tres módulos y dos interfaces:

Módulo de Carga/Actualización: en el cual se puede incluir, modificar y eliminar información de la Gestión de Refinación que se maneja en este sistema.

Módulo de Consulta: en el cual se puede generar información consolidada de una o más Refinerías o Complejos, en un periodo determinado y selección de los otros criterios correspondientes a cada opción, con la información cargada y confirmada en el Módulo de Carga/Actualización.

Módulo de Administración: en el cual los usuarios con el rol de administrador, pueden incluir, modificar y eliminar información de las tablas maestra y administración de usuarios. Así como también permite la

Capítulo 1 Fundamentación Teórica

consulta de la trazabilidad de las acciones ejecutadas en la carga de los datos transaccionales. Y por otra parte, se maneja la consulta de información concerniente a los datos de los usuarios.

Contiene las opciones: Tablas Maestras, Trazabilidad, Usuarios, Cambiar Password, Consulta General de Usuarios, Consulta Privilegios por Usuarios.

Interfaz de Volúmenes Reales Mensuales con el Sistema de Información de Movimiento del Petróleo (SIMP): Trae los datos de Volúmenes Reales Mensuales de las Refinerías Amuay y Cardón, la cual es activada desde la aplicación SIMP y una vez que en el SIM sea confirmada esta información, no se podrá activar nuevamente esta interfaz para dicha refinería mes y año.

Interfaz de Mermas con el Sistema de Información de Movimiento del Petróleo (SIMP): Trae los datos de Mermas de las Refinerías Amuay y Cardón, la cual es activada desde la aplicación SIMP y una vez que en el SIM sea confirmada esta información, no se podrá activar nuevamente esta interfaz para dicha refinería mes y año.

Sistema Integrado de Movimientos y Embarque de Petróleo (SIMP)

El Sistema Integrado de Movimiento y Embarques de Petróleo (SIMP) tiene por principales objetivos el control y supervisión de todas las operaciones de movimiento de crudo entre los patios de tanque y los terminales de embarque, el control del movimiento de crudo y productos en refinería, la planificación, ejecución y control de las operaciones de carga y descarga de los tanqueros en el terminal de embarque y la generación y emisión de toda la documentación de embarque.

El proceso se inicia con el manejo de órdenes de movimiento y la supervisión de la ejecución de los movimientos a través de conexión directa a sistemas de telemetría de tanques.

El sistema contempla la definición y registro de las disponibilidades de crudos y productos para el mes, las cuales sirven de insumo para establecer el programa de nominaciones y luego el programa diario de embarques en función de la capacidad y disponibilidad de las facilidades del terminal.

Cuando el buque llega al puerto genera su plan de carga y se procede a la ejecución del embarque, fase en la cual se lleva un registro detallado de las operaciones realizadas y de los tiempos utilizados en cada una de ellas.

Al concluir la ejecución del embarque se procede a la generación de la documentación que debe acompañar a cada embarque.

Luego se generan estadísticas de eficiencia del puerto y del factor de experiencia de los buques.

El sistema fue puesto en producción en Lagunillas, Puerto Miranda y Refinería Cardón.

Características resaltantes del proyecto:

- Disparo de alarmas; cálculo de los volúmenes y masa manejados en cada transacción.
- Monitoreo de la información en tiempo real (Movimientos planificados activos / inactivos, Movimientos detectados no programados, Sistemas de transporte utilizados, Escaneo de tanques,

Capítulo 1 Fundamentación Teórica

Seteo de alarmas, Imagen del tanque - nivel, volumen, masa, temperatura, Tasas de bombeo, Imagen de los oleoductos), programación de muelles.

- Registro y control de ETAs, registro de datos del cargamento y de las actividades de carga y descarga de los tanqueros en puerto, control de demoras, registro de aforos de tanque, registro y control del personal de muelle, comparación de cifras buque versus cifras puerto
- Emisión de documentación oficial y reportes de control: hojas de tiempo, certificados de calidad y cantidad, certificado de origen, guía de embarque, facturas y tasas portuarias, cartas de protestas o notificaciones.
- Estadísticas de desempeño del puerto: eficiencias de carga/descarga, deslastre y fondeo, estadísticas de demoras, eficiencia portuaria.
- Levantamiento de información, análisis y diseño de los procesos a automatizar, construcción y puesta en marcha.
- Manejador relacional Oracle Enterprise
- Utilización de herramientas Case y tools de Oracle
- Más de DE 200 TABLAS, aprox. 200 FORMAS, 50 REPORTE y 70 VISTAS
- Utilización de pantallas gráficas para la visualización y programación de la disponibilidad de facilidades de muelle;
- Uso de teoría de grafos para el cálculo y reconciliación de volúmenes y masas movidas entre instalaciones
- Desarrollo en ANSI C (sobre el compilador de C de Berkley GCC)
- Uso de librerías de X11/Motif para el desarrollo de interfaces gráficas.
- Simplificación de procesos en el control de movimientos y emisión de documentación
- Eliminación de documentos manuales.

Aunque estos sistemas están llenos de facilidades que ayudan a un mejor desarrollo de los procesos de refinación, se debe señalar que hay actividades que no están cubiertas y que son de importancia para el desempeño exitoso de los procesos del negocio. A esto se suma que son sistemas desarrollados utilizando software propietario y PDVSA tiene expectativas de migrar a software libre. Lo que hace que esta no sea una buena alternativa a seguir utilizando por los clientes.

1.5. Conclusiones.

Después de hacer un estudio exhaustivo de los procesos de administración de la información utilizada por la Gerencia de Planificación y Control de Refinación de PDVSA y de control de acceso a los datos, analizar el estado actual de estos procesos en el entorno de la organización para la que se trabaja y

Capítulo 1 Fundamentación Teórica

examinar el estado del arte de los sistemas de gestión de la información que utiliza PDVSA actualmente, se ha llegado a la conclusión de que:

La Gerencia de Planificación y Control de Refinación de PDVSA necesita un software que automatice los procesos de administración de la información y le brinde a su vez la posibilidad de llevar un control del acceso a los datos por parte de los usuarios, definiendo roles, permisos y que implemente un informe de seguimiento de las acciones realizadas por los usuarios.

Este sistema no debe estar desarrollado utilizando software propietario, dado que el estado venezolano no invertirá más en la compra de este tipo de software, dado que está migrando completamente a software libre.

CAPÍTULO 2

Tendencias y tecnologías actuales a desarrollar

2.1 Introducción.

En este capítulo se hace un estudio de los lenguajes de programación más apropiados para el desarrollo del sistema, teniendo en cuenta las características del entorno donde se aplicará la solución propuesta, se realizó un análisis detallado de las tendencias y tecnologías actuales adecuadas a la solución, el Sistema de Gestión de Base de Datos utilizado y se considera además la metodología a utilizar.

2.2 Descripción de las tendencias y tecnologías actuales.

2.2.1. Aplicaciones Web.

Una aplicación web es aquella que los usuarios usan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero, además de la facilidad para actualizar y mantener los sistemas sin la necesidad de distribuir e instalar software en miles de potenciales clientes.

Las aplicaciones web generan dinámicamente páginas en un formato estándar, soportado por navegadores web comunes como HTML o XHTML. Se utilizan lenguajes interpretados del lado del cliente, como Java Script, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

Las aplicaciones Web son una especialización y concreción de las aplicaciones cliente/servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son estándar, y no han de ser creados por el desarrollador.

Algunas ventajas de las Aplicaciones Web son:

- **Multiplataforma e interoperabilidad:** A diferencia de las aplicaciones de escritorio, que sólo pueden funcionar bajo el sistema para el que fueron diseñadas, las aplicaciones web son multiplataforma por diseño. Esto significa que podrá conectar con el software desde cualquier versión de Windows -presente o futura-, o incluso otros sistemas operativos como GNU/Linux, Solaris, Symbian (teléfonos móviles GPRS). (11)
- **Actualización instantánea:** Debido a que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.

Capítulo 2 Tendencias y tecnologías actuales

- **Suave curva de aprendizaje:** Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- **Fácil de integrar con otros sistemas:** Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- **Acceso móvil:** El usuario puede acceder a la aplicación con la única restricción de que cuente con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.
- **Multiusuario:** A diferencia de las aplicaciones de escritorio, donde sólo el usuario frente al ordenador puede utilizar la aplicación, en las aplicaciones web puede haber varios usuarios conectados al sistema simultáneamente, cada uno a través de un ordenador distinto y en una ubicación geográfica separada (empresas con varios almacenes, por ejemplo), y todos utilizar la aplicación con normalidad.
- **Seguridad.** Al albergarse en un servidor remoto, el funcionamiento de la aplicación y los valiosos datos que contiene son totalmente independientes del ordenador utilizado para la gestión. Así, la normal operación de la aplicación es inmune a una avería de hardware, virus informáticos, o cualquier otro problema local. En caso de necesidad, basta con retomar la conexión al servidor desde cualquier otro ordenador personal o portátil.

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web es la primera capa, un motor usando alguna tecnología web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario. (12)

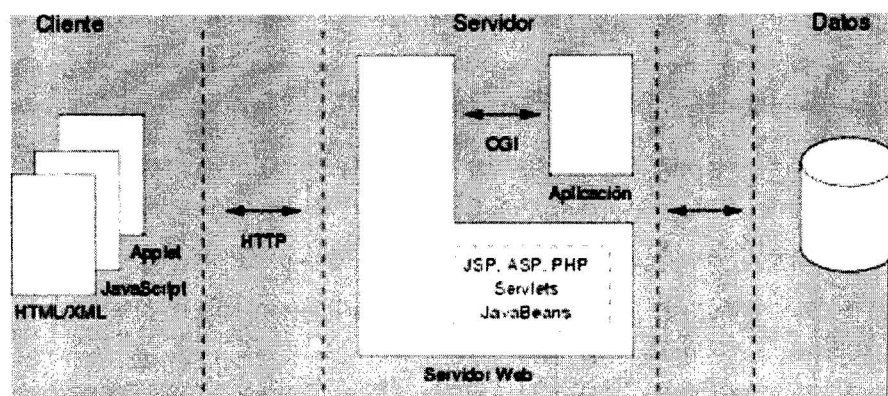


Figura 2. Esquema general de las tecnologías Web.

Capítulo 2 Tendencias y tecnologías actuales

Las aplicaciones Web implementan lógica de negocios. Toda aplicación refleja parte del funcionamiento del mundo real, para automatizar tareas que de otra manera serían llevadas a cabo de modo ineficiente, o bien no podrían realizarse, tienen que hacerlo bien, lo mejor posible, o lo más real o cercano posible, para ello, es necesario que cada aplicación refleje las restricciones que existen en el negocio.

2.2.2. Modelo Cliente - Servidor.

La arquitectura cliente – servidor, conocida también como modelo cliente – servidor, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realizada se efectúe con la mayor eficiencia, y permita simplificarlas. (13)

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura.

La arquitectura cliente - servidor tiene las siguientes características:

- **Compartición de recursos:** Servicios ofrecidos a muchos clientes. Un servidor puede atender muchos clientes que solicitan esos servicios.
- **Transparencia de ubicación:** El servidor es un proceso que puede residir en el mismo ordenador que el cliente o en uno distinto a lo largo de una red. Un programa puede ser un servidor en un momento y convertirse en un cliente posteriormente.
- **Mezcla e igualdad:** Esta es una de las más importantes ventajas de este paradigma. Una aplicación cliente - servidor, idealmente es independiente del hardware y de sistemas operativos; mezclando e igualando estas plataformas.
- Interacción a través de mensajes, para envío y respuesta de servicios.
- Servicios encapsulados, exponiendo los servicios a través de interfaces, lo que facilita la sustitución de servidores sin afectar los clientes; permitiendo a la vez una fácil escalabilidad.

El sistema se basa en la arquitectura cliente-servidor sobre la plataforma Web, donde cada instancia en el cliente es independiente de la ejecución de otra.

2.2.2.1. Patrón Modelo – Vista – Controlador.

En la arquitectura cliente – servidor se pueden implementar varios patrones de arquitectura entre los que se encuentra el patrón de Capas y el Modelo – Vista – Controlador (MVC).

El patrón MVC separa los datos de una aplicación, de la interfaz de usuario, y la lógica de control en tres componentes distintos. Este patrón se ve frecuentemente en aplicaciones Web, donde la vista es

Capítulo 2 Tendencias y tecnologías actuales

la página HTML, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

Importancia

MVC se encuentra en diseños de aplicaciones reconocidas por sus interfaces sofisticadas, dando razones para apostar por su utilización. La lógica de una interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si en contraposición con MVC se realiza un diseño ofuscado (mezclar los componentes de interfaz y de negocio), la consecuencia sería que al necesitar cambiar la interfaz, se debe modificar también los componentes de negocio, dando lugar a mayor esfuerzo y la posible introducción de errores.

Elementos del patrón

Divide la aplicación interactiva en 3 áreas, para ello utiliza las siguientes abstracciones:

- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada. Es el responsable de Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento (definido así en el sistema). Definir además las reglas de negocio.
- **Vista (View):** Intercambia la información con el usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. Recibir los eventos de entrada y comunicar las acciones que ellos derivan a la capa de Modelo.
- **Controlador (Controller):** Recibe las entradas, traducidas a solicitudes de servicio para el modelo. Se responsabiliza con recibir datos del modelo a través del Controlador para mostrar al usuario.

El siguiente diagrama representa la estructura de capas del sistema.

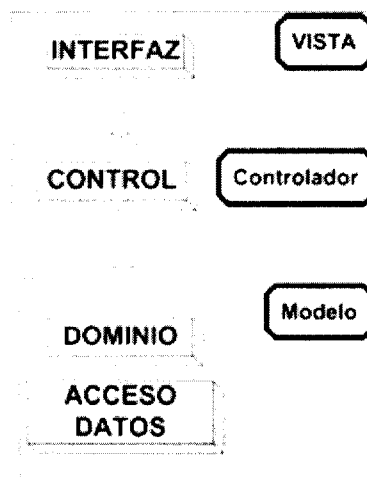


Figura 3. Estructura de capas del patrón MVC aplicado en el sistema.

Se ha modelado haciendo uso del patrón Modelo - Vista - Controlador con el objetivo de utilizar la separación de las responsabilidades de cada una de las capas que lo conforman y así lograr facilidades de desarrollo.

2.3. Lenguajes de programación.

Hoy en día existen diferentes lenguajes de programación para Web. Dichos lenguajes se clasifican en dos partes fundamentales: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes por el auge que estos han tenido, algunos como PERL, ASP, PHP y Java. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos, tratamiento de la Información, entre otras funciones. Del lado del cliente se encuentra principalmente el Java Script, encargado de aportar dinamismo a la aplicación en los navegadores.

Seguidamente se realizará una descripción de los lenguajes utilizados para el desarrollo de este trabajo.

2.3.1. HTML

Hyper Text Markup Language (HTML, por sus siglas en inglés), en español Lenguaje de Etiquetas de Hipertexto, es el lenguaje de marcado predominante para la construcción de páginas web. Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada navegador. Se puede decir por lo tanto que el HTML es el lenguaje usado por los navegadores para mostrar las páginas webs al usuario, siendo hoy en día la interface más extendida en la red. (14)

HTML es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. Es un lenguaje muy sencillo presentado de forma estructurada y agradable, con enlaces (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas.

Este lenguaje permite aglutinar textos, sonidos e imágenes así como la combinación de estos. Además, es aquí donde reside su ventaja con respecto a libros o revistas, el HTML permite además la introducción de referencias a otras páginas por medio de los enlaces hipertexto.

2.3.2. Java Script.

Conocido lenguaje de programación script que cuenta con amplio soporte en navegadores y herramientas de desarrollo web, es un lenguaje interpretado, orientado a objetos y con una sintaxis semejante a la del lenguaje Java y el lenguaje C. (14)

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de

interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

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

Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el más utilizado de su tipo. Es un lenguaje con muchas posibilidades, que permite la programación de pequeños scripts, pero también de programas más grandes con funciones y estructuras de datos complejas. Además de poner a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente sin necesidad de volver a cargar la página. (15)

2.3.3. PHP.

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web, se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios.

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente PHP Tools, o, *Personal Home Page Tools*), fue primero escrito por Rasmus Lerdorf como un simple conjunto de scripts de Perl para guiar a los usuarios en sus páginas. Luego para satisfacer inquietudes del mismo tipo por parte de otra gente lo reescribe, pero esta vez como un lenguaje de script agregándole entre otras características soporte para formularios. Al ver como la popularidad del lenguaje aumenta, un grupo de desarrolladores crea para él un API, convirtiéndose así en el PHP3. Fue en ese momento cuando el parser de scripts PHP es completamente reescrito (el Zend Engine) dando vida al PHP4 mucho más rápido. La más reciente versión principal del PHP fue la versión 5.2.6 del 1 de mayo de 2008. (16)

PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Ventajas adicionales.

PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de Unix, Windows. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al OS.

PHP es completamente expandible. Está compuesto de un sistema principal (escrito por Zend), un conjunto de módulos y una variedad de extensiones de código.

Capítulo 2 Tendencias y tecnologías actuales

Puede interactuar con muchos motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, y otros muchos.

PHP generalmente es utilizado como modulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.

PHP es Open Source, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos. Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. (17)

Después que la comunidad ha revisado el paquete de legados que ha dejado el PHP, se han realizado cambios estructurales en el lenguaje para ofrecer innovación en el nuevo PHP 5 y solucionar muchos de los problemas encontrados en PHP 4.

Afortunadamente, lo nuevo de PHP 5 mejora muchas áreas en el lenguaje y su ejecución, como por ejemplo:

- Programación orientada a objetos (OOP).
- MySQL.
- XML.
- Integración nativa con el Zend Engine.

Los diseñadores de PHP5 han realizado un cambio radical en el tratamiento de las variables objeto: en PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador '&' ni en las asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de potenciales errores. La principal novedad en las clases de PHP5 es la inclusión de modificadores de control de acceso para implementar la encapsulación piedra angular en la programación orientada a objetos de la que adolecía PHP4. (18)

De acuerdo con las características y comparaciones anteriormente descritas, php5 es mucho más favorecido ya que resulta provechoso a la hora de realizar una migración de Sistemas Operativos, tiene mayor velocidad de ejecución, es compatible con el framework y el Sistema de Gestión de Bases de Datos utilizados, además incorpora Programación Orientada a Objetos lo que lo convierte en un lenguaje aún más versátil por tanto es el adecuado para implementar la propuesta de sistema de este trabajo.

2.3.3.1. Framework CakePHP.

CakePHP es un *framework* de desarrollo de aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails (*framework* de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby). Empezó en 2005, cuando Ruby On Rails estaba ganando popularidad. Desde entonces la comunidad creció y generó muchos subproyectos. Facilita al usuario la interacción con la base de datos. Además hace uso del patrón Modelo – Vista - Controlador.

CakePHP está en la versión 1.1.19.6305 y la versión 1.2.0.x se encuentra en fase beta de desarrollo. (19)

La fuerza de CakePHP no sólo reside en su versatilidad y potencia, sino también en que la comunidad de programadores lo ha acogido con muchas ganas, y se ha creado una comunidad muy activa, lo que propicia que se desarrollen más módulos y se tenga código muy competitivo y eficaz.

Por otra parte, la compatibilidad con las versiones 4 y 5 de PHP, la disponibilidad de plantillas flexibles, la instalación cómoda y casi sin requisitos en servidor, junto con otras características como son el llevar un módulo de validación incorporada uso de sesiones, helpers en Vistas para AJAX, Javascript, Formularios HTML, y URL amigables, se puede tener la web entera cacheada, para que su acceso sea más rápido. (20)

Otras características destacables de CakePHP son una licencia muy flexible, ya que, al usar la licencia MIT, permite no sólo usar, copiar o modificar el código, sino que también se podría publicar, sublicenciar o vender copias de otra aplicación desarrollada con esta plataforma, otra característica que facilita mucho el uso de CakePHP en el mayor número de entornos posibles, es que funciona incluso sin reconfigurar Apache. (21)

CakePHP incorpora además una característica muy útil, especialmente en el período de puesta en marcha de una aplicación, el *Scaffolding* (*método para construir aplicaciones basadas en bases de datos*). Al activar el *Scaffolding* para la aplicación, o para una parte de ella, es el propio código de CakePHP el que se encargará de analizar la estructura de la base de datos y crear listados de los registros con botones para insertar nuevos registros, borrar o editar los existentes. Asimismo también crea los formularios para añadir o editar los registros.

CakePHP también aporta componentes internos para el manejo de sesiones, seguridad y listas de control de acceso, que agilizarán la incorporación de estas funcionalidades a las aplicaciones sin perder ni un ápice de flexibilidad.

CakePHP es el *framework* mas conveniente a usar en este caso puesto que tiene una licencia flexible, es compatible con el lenguaje de programación utilizado, facilita la tarea de crear la aplicación, y con su utilización es posible dotarse de una buena metodología a la hora de programar, lo que incentivará la reutilización de código en futuras aplicaciones.

2.4. Servidores Web.

Un servidor web es un programa que implementa el protocolo HTTP (*Hypertext transfer protocol*). Este protocolo está diseñado para transferir los llamados hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que se conoce como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. (22)

Algunos servidores web que cuentan con gran popularidad en la actualidad son:

- Apache.
- IIS (Internet Information Server).

Para el desarrollo de la solución propuesta se decidió utilizar el primero, dando los elementos del por qué de esta decisión en el siguiente subepígrafe.

2.4.1. Apache.

El servidor Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo.

Hoy en día Apache es el servidor web más utilizado del mundo (23), encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable.

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP). Entre sus características se destacan:

- Multiplataforma.

Capítulo 2 Tendencias y tecnologías actuales

- **Modular:** Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API¹ de programación de módulos, para el desarrollo de módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta
- **Extensible:** gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

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, aceptando 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 multiprocesos 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 consigue 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. (24)

2.5. Sistema de Gestión de Bases de Datos (SGBD).

Sistema de Gestión de Bases de Datos (SGBD) paquete de software que proporciona todas las facilidades para la creación, recuperación, manipulación y mantenimiento de bases de datos, asegurando su integridad, confidencialidad y seguridad.

Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son:

¹ Application Programming Interface - Interfaz de Programación de Aplicaciones.

Capítulo 2 Tendencias y tecnologías actuales

- **Control de la redundancia:** La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- **Restricción de los accesos no autorizados:** cada usuario ha de tener unos permisos de acceso y autorización.
- **Cumplimiento de las restricciones de integridad:** el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. (19)

Ventajas

1. Facilidad de manejo de grandes volúmenes de información.
2. Gran velocidad en muy poco tiempo.
3. Independencia del tratamiento de información.
4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
6. Integridad referencial al terminar los registros.

Los SGBD utilizados en el mundo son Oracle, MySQL, SQL Server de Microsoft, PostgreSQL entre otros. Estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional. Todos los sistemas mencionados anteriormente facilitan el trabajo con la base de datos y tienen características que los diferencian, por ejemplo:

- **Oracle:** requiere de una licencia para poderlo utilizar (no es gratis).
- **Microsoft SQL Server:** no es multiplataforma, solo puede ser utilizado con el sistema operativo Windows que está patrocinado por la compañía Microsoft.
- **MySQL:** licencia GPL a partir de la versión 3.23.19, su principal objetivo de diseño fue la velocidad, otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria. (25)
- **PostgreSQL:** soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos. (25)

Cada sistema tiene características, ventajas e inconvenientes, la elección de uno u otro sistema para gestionar una base de datos vendrá definida por sus particularidades y necesidades.

2.5.1. PostgreSQL

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD (*Berkeley Software Distribution*).

Capítulo 2 Tendencias y tecnologías actuales

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.

Ventajas

1. Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
2. Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
3. Tiene mejor soporte para triggers y procedimientos en el servidor.
4. Soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

Para sistemas más serios en las que la consistencia de la BD sea fundamental PostgreSQL es una mejor opción.

En este caso se decidió tomar PostgreSQL, ya que se caracteriza por su estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. Es el motor de bases de datos de código abierto más potente del momento y en sus últimas versiones empieza a no tener que envidiarle nada a otras bases de datos comerciales.

La última serie de producción es la 8.2, que es la utilizada en el proyecto, se puede ejecutar en la gran mayoría de los sistemas operativos existentes en la actualidad, entre ellos Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows. (26)

2.6. Metodologías de Desarrollo de Software.

Una metodología es un conjunto ordenado de pasos a seguir para cumplir un objetivo. Dentro de la Ingeniería de Software, el objetivo es el desarrollo de software de alta calidad que cumpla con las necesidades del cliente.

Para dar una idea de qué metodología se puede utilizar y cual se adapta más al medio, a continuación se mencionan cuatro de ellas, consideradas las más importantes, tales como: *Rational Unified Process* (RUP), *Extreme Programming* (XP), *Feature Driven Development* (FDD) y *Microsoft Solution Framework* (MSF).

A continuación se exponen las razones para escoger RUP como metodología de desarrollo del sistema.

2.6.1. Fundamentación de la metodología utilizada.

Para controlar y planificar la propuesta que presenta este trabajo, se decidió utilizar por sus características y las facilidades que aporta el Proceso Unificado de Desarrollo (RUP), ya que es más adaptable para proyectos de largo plazo, característica muy compatible al sistema pues es de larga duración y además cuenta con un equipo de trabajo relativamente grande. Todo lo contrario de XP y FDD que se recomiendan para proyectos de corto plazo y equipos pequeños, en el caso de MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

RUP es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema en construcción está formado por componentes de software interconectados a través de interfaces bien definidas.

Lo que hace único al proceso unificado es que está dirigido por casos de usos, centrado en la arquitectura, es iterativo e incremental, muy acorde con la naturaleza cambiante de los requisitos en muchos proyectos lo que permite la eliminación de errores en etapas previas. Es debido a todas estas características que se utiliza en toda la etapa de modelo del negocio, análisis de requerimientos, diseño, y construcción, lo que permite obtener una mejor calidad del producto, y facilita el control del proceso de desarrollo.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

1. **Inicio:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
2. **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
3. **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
4. **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

El proceso define una serie de roles que se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado que se espera de ellos.

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la empresa. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

2.6.1.1. Lenguaje Unificado de Modelado.

En esta metodología se utiliza el Lenguaje Unificado de Modelado (UML, *Unified Model Language*) como lenguaje representativo para visualizar, especificar, construir y documentar los artefactos del sistema.

UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar.

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo.

UML ha evolucionado durante los años y está ahora en su versión 2.x. La versión de UML para la herramienta de modelado Visual Paradigm se ha sincronizado con el nuevo desarrollo de UML 2.1 para proporcionar un ambiente modelado visual que satisface la tecnología del software de hoy y las necesidades de comunicación.

2.7. Herramientas CASE de Desarrollo de Software.

El uso de las herramientas CASE (*Computer Aided Software Engineering*) puede mejorar la productividad en el desarrollo de un software. La ingeniería de sistemas asistida por ordenador es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

CASE es una combinación de herramientas software (aplicaciones) y de metodologías de desarrollo:

1. Las herramientas permiten automatizar el proceso de desarrollo del software.
2. Las metodologías definen los procesos automatizar.

Estas herramientas tienen como objetivos mejorar la planificación de un proyecto, mejorar la productividad en el desarrollo y mantenimiento del software. Además ayudan a mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos, aumentar la calidad del software y facilitar el uso de las distintas metodologías propias de la Ingeniería del Software.

Aunque no es fácil y no existe una forma única de clasificarlas, las herramientas CASE se pueden clasificar en base a los parámetros siguientes:

1. Las plataformas que soportan.
2. Las fases del ciclo de vida del desarrollo de sistemas que cubren.
3. La arquitectura de las aplicaciones que producen.
4. Su funcionalidad.

CASE cuenta además con una amplia lista de aplicaciones, dos de las más usadas son Rational Rose Enterprise y Visual Paradigm.

La más adecuada a utilizar en este caso es el Visual Paradigm, ya que es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software, es multiplataforma y es orientado a objeto, todo lo contrario del Rational Rose que es una herramienta con plataforma independiente. Es muy sencillo de usar, fácil de instalar y actualizar. Genera código para varios lenguajes. Tiene integrado el MS Visio y es compatible con otras ediciones, posibilita un entorno de creación de diagramas para UML 2.x.

2.7.1. Visual Paradigm.

El Visual Paradigm es una Suite de herramientas CASE, dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, lo cual garantiza la calidad del

Capítulo 2 Tendencias y tecnologías actuales

producto final. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. En el proyecto la versión a utilizar es la Enterprise Edition, entre las principales características de esta herramienta se encuentran:

1. Es profesional: da la posibilidad de crear un gran conjunto de artefactos de las distintas fases del desarrollo del software.
2. Contiene facilidades para redactar y generar documentación de la Ingeniería de Software.
3. Sincronización entre Diagramas de Entidad Relación y Diagramas de Clases.
4. Generación de Código / Ingeniería Inversa: entre los lenguajes conocidos por Visual.
5. Integración con distintos Ambientes de Desarrollo Integrados (IDE).

2.8. Conclusiones.

En el capítulo se realizó un análisis descriptivo de las tecnologías y tendencias actuales en las cuales se basó el trabajo, profundizándose en el conocimiento de conceptos necesarios para su comprensión, y exponiendo sobre todo ventajas que llevaron a elegir las herramientas y lenguajes utilizados. Para el desarrollo del sistema se hace uso de la tecnología para la programación de páginas dinámicas, el lenguaje PHP5 con soporte de base de datos en PostGreSQL. Además el proceso de desarrollo utilizado es RUP, el cual está basado en la orientación a objetos y la modelación visual usando UML.

CAPÍTULO 3

Presentación de la solución propuesta

3.1. Introducción.

En el presente capítulo se hace la descripción de la propuesta que trae este trabajo, para ello se describen los procesos del negocio que tienen que ver con el objeto de estudio, presentando el modelo de negocio.

Además se enumeran los requisitos funcionales y no funcionales que debe cumplir el sistema que se propone, lo que permite hacer una concepción general del software, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, además de mostrar las descripciones de los casos de uso del sistema.

3.2. Modelo de Negocio.

Una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio el que determine los requisitos. Este modelo permite comprender la estructura y la dinámica de la organización en la cual se va a implantar el sistema y entender los problemas actuales de la organización e identificar las mejoras potenciales. (27)

3.2.1. Actores y trabajadores del negocio.

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. (28)

Actores del Negocio	Justificación
Gerencia Crudos y Productos.	Es el que inicia el caso de uso, recibe la información de la Gerencia de Comercio y Suministro, que se encarga de enviar la información de la refinería.
Gerencia de Finanzas.	Es quien envía los costos de crudos e insumos totales y el costo de procesamiento por Correo Electrónico iniciando así el caso de uso Recepcionar de costos Real.
Evento Resultados de Refinación.	Es la acción de recepción del listado de precios ventas y de mercado, además es la manera de

Capítulo 3 Presentación de la solución propuesta

	conocer que se debe calcular el Margen Bruto y el Margen Neto.
Superintendente de Programación.	El Superintendente de Programación de cada refinería envía antes de las 10 de la mañana diariamente el reporte operacional, que contiene una sección de comentarios donde aparecen aquellas plantas que están paradas ya sea programadas o no y en el caso de ésta última una explicación de la causa. Contiene además la carga de cada una de las plantas que permite confirmar la información sobre parada de planta informada.
Comité de Evaluación de Parada de Planta.	Su función es conciliar y autorizar el plan de Paradas de Plantas de todas las Refinerías y en caso de ser necesario hacer modificaciones al Plan de Paradas de Plantas debe autorizarlas previa conciliación en el Sistema.
Evento Recepción de Información.	Es el Evento Recepción de Información encargado de iniciar los casos de usos Realizar seguimiento de paradas y cargas a planta y Evaluar el Indicador Paradas de Plantas.
Evento Reporte de resultados operacionales y financieros.	Una vez que se ha recibido de las refinerías el Reporte de resultados operacionales y financieros se procede a la evaluación y agrupamiento de este Indicador.

Tabla 1. Descripción de los Actores del Negocio.

Un trabajador es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol. (28)

Trabajadores del Negocio	Justificación
Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación.	Es la persona que vela por que se cumpla con los Precios de Ventas en las refinerías a la hora de vender un producto determinado, se encarga de

Capítulo 3 Presentación de la solución propuesta

	revisar los márgenes recibidos de las refineries, es quien grafica y evalúa el Margen Bruto y el Margen Neto.
Superintendente de Economía de Refinería.	Calcula el Margen Bruto de la Refinería y de cumplir con que se vendan los productos al precio recibido, se encarga además de enviar el Archivo de Exportaciones e Importaciones y el Margen Bruto de la Refinería.
Analista de Parada de Planta de la Gerencia de Refinación.	Es la persona encargada de recepcionar el Reporte diario de Operaciones, el Plan de Parada de Planta y Consolidar las Paradas de Plantas.
Analista de Gestión de Refinación de la Gerencia de Planificación y Gestión de Refinación.	Es quien revisa e incluye en el Reporte de resultados operacionales de un periodo, los Indicadores de Seguridad.

Tabla 2. Descripción de los Trabajadores del Negocio.

3.2.2. Diagrama de Casos de Uso del Negocio.

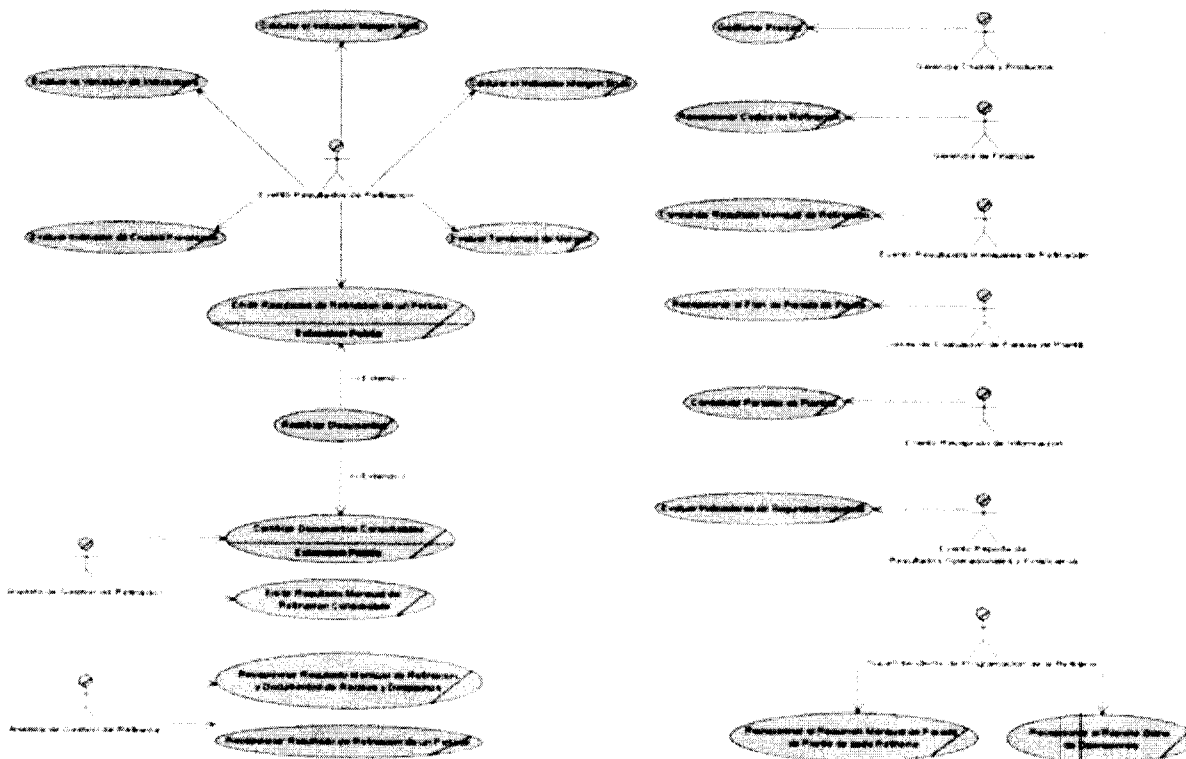


Figura 4. Diagrama de Casos de Uso del Negocio.

3.2.3. Procesos y Casos de Uso del Negocio.

Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Un caso de uso del negocio representa a un proceso del negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio.

En el entorno de PDVSA, se encuentran establecidos 5 procesos fundamentales, los cuales a su vez engloban otros procesos. A continuación se exponen estos procesos generales y se hace un resumen de cada uno de los casos de uso del negocio. Para un mejor entendimiento del negocio, en el **Anexo 1** se exponen los Diagramas de actividades de estos CU del Negocio.

3.2.3.1. Proceso de Elaboración de Margen Bruto Real.

CUN1- Gestionar Precios.

Actores: Gerencia Crudos y Productos.

Trabajadores: Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación. Superintendente de Economía de refinería.

Resumen: El caso de uso se inicia cuando se recibe de la Gerencia de Crudos y Productos, los Precio Reales y Precios de Mercado. Los Precios Reales se consolidan dando origen a los Precios Reales de Ventas, que se añaden al Archivo de Exportaciones e Importaciones que se recibe de la refinería, éste regresa a la refinería.

CUN2- Recepcionar Costos de Refinación.

Actores: Gerencia de Finanzas.

Trabajadores: Analista de Gestión de Refinación de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando se recibe de la Gerencia de Finanzas los datos relacionados los Costos de Refinación. Para el cálculo del Margen Neto.

CUN3. Calcular el Indicador Margen Bruto Real.

Actores: Evento Resultados de Refinación.

Trabajadores: Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de Uso se inicia cuando se tiene el Balance Volumétrico Consolidado, los Precios Reales de Crudo y Producto, se hace un análisis de mercado con los precios enviados por la gerencia de Crudo y Producto que permiten conocer la situación de los precios reales con respecto a estos,

Capítulo 3 Presentación de la solución propuesta

luego se calcula el Margen Bruto, el Margen que se ha recibido de las refinarias se revisa y rectifica para su posterior incorporación en las diapositivas que contienen este Indicador y sus notas explicativas.

CUN4. Calcular Margen Neto.

Actores: Evento Resultados de Refinación.

Trabajadores: Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación. Analista de Gestión de Refinación de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando una vez calculado el margen bruto se le envía al Analista de Gestión de refinación y este que ha recibido previamente el Listado de costo de crudos e insumos totales y el costo de procesamiento, restándole al margen bruto el costo de procesamiento y finaliza así el caso de uso.

CUN5. Evaluar Tendencia de Margen.

Actores: Evento Resultados de Refinación.

Trabajadores: Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando se recibe ya calculado el margen bruto por el Analista de Economía de Refinación y el margen Neto ya calculado por el Analista de Gestión de Refinación se comparan el periodo analizado con igual periodo del año anterior, además se calcula la tendencia del margen bruto de cada refinería en el periodo analizado y se grafica cada resultado.

3.2.3.2. Parada de Planta.

CUN1- Recepcionar el Reporte Diario de Operaciones.

Actores: Superintendente de Programación.

Trabajadores: Analista de Parada de Planta de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando el Superintendente de programación envía el Reporte Diario de Operaciones, luego es revisado por el Analista de Parada de Planta de la Gerencia de Refinación, de no entenderse las notas explicativas se solicita ampliar el comentario por teléfono o correo electrónico.

CUN2- Recepcionar el Plan de Parada de Planta.

Actores: Comité de Parada de Planta.

Trabajadores: Analista de Parada de Planta de la Gerencia de Planificación y Gestión de Refinación.

Capítulo 3 Presentación de la solución propuesta

Resumen: El Plan de Parada de Planta de cada refinería se envía al Comité de Parada de Planta conformado por la Gerencia de refinación, Planificación, y Estrategia. Es entregado a la Gerencia de Refinación para su monitoreo.

CUN3- Consolidar Parada de Planta.

Actores: Evento Recepción de Información.

Trabajadores: Analista de Parada de Planta de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando se recibe el Resumen Mensual de Parada Planta y el Reporte Operacional y Financiero de cada Refinería. Estos documentos se comparan para evaluar la exactitud de la información y se elabora las láminas de PPT que forman parte del Resultado de Refinación del Período.

CUN4- Recepcionar Resumen Mensual de Paradas de Planta de cada Refinería.

Actores: Superintendente de Programación de la Refinería.

Trabajadores: Analista de Parada de Planta de la Gerencia de Planificación y Gestión de Refinación.

Resumen: El caso de uso se inicia cuando se recibe, del Superintendente de Programación de la refinería, el Resumen Mensual de Parada de Planta de cada Refinería, el mismo es revisado teniendo en cuenta las Paradas de Plantas y las Cargas a Plantas en el Reporte Diario de Operaciones que previamente se encuentra archivado. De no existir coincidencias entre estos documentos se solicita aclaración a la refinería que corresponda.

3.2.3.3. Indicadores de Seguridad Industrial.

CUN1- Evaluar Indicadores de Seguridad.

Actores: Evento Reporte de resultados operacionales y financieros

Trabajadores: Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación.

Resumen: Una vez que se han recepcionado los Indicadores de Seguridad de las refinerías en el Reporte de resultados operacionales y financieros, se procede a su revisión, y se incluye en el Reporte de resultados operacionales de un período que se envía al Comité Operacional.

3.2.3.4. Recepcionar, Consolidar, Emitir, Certificar y Rectificar.

CUN1- Recepcionar Resultado Mensual de Refinación y Documento de Recibos y Despacho.

Actores: Analista de Gestión de Refinería.

Trabajadores: Analista de Gestión de Refinación.

Capítulo 3 Presentación de la solución propuesta

Resumen: Las refinерías envían sus resultados volumétricos en un documento con los datos de: (Crudos Procesados, Productos en Inventario, Productos Recibidos, Productos Entregados, Productos Consumidos en Refinería, Volumen Producido, Productos Reprocesados) y el Documento de Recibos y Despacho de Productos (Notas Explicativas) que contiene información necesaria para la eliminación de transferencias. Este documento se archiva y se notifica a las refinерías de su recepción.

CUN2- Recepcionar Resultado de Refinación de un Período.

Actores: Analista de Gestión de Refinería.

Trabajadores: Analista de Gestión de Refinación.

Resumen: El Analista de Gestión de Refinería envía un documento en formato PPT elaborado con la información de Resultados Mensual de Refinación, este documento contiene la información real y el plan de (Crudo, Insumos, Producción Obtenida, Costo, Paradas de Planta, Margen Bruto, Margen Neto, Indicadores de Seguridad, Presupuesto Operacional y de Inversiones, fuerza laboral e Inversión Social). En el documento se incluye la explicación de las variaciones del plan con respecto al real. Este documento se archiva y se notifica a las refinерías de su recepción.

CUN3- Consolidar Resultado Mensual de Refinación.

Actores: Evento Resultados Mensual de Refinación.

Trabajadores: Analista de Gestión de Refinación. Documento Máster.

Resumen: El caso de uso se inicia cuando se reciben todos los documentos de las refinерías, la información se introduce por el Analista de Gestión en un Sistema, (Documento Máster (Excel)) y este Sistema mediante un conjunto de fórmulas elimina las transferencias entre refinерías y elabora el consolidado. Se obtiene como resultado un documento consolidado de los resultados del Balance Volumétrico de Refinación.

CUN4- Emitir Resultado Mensual de Refinación Consolidado.

Actores: Analista de Gestión de Refinación.

Trabajadores: Gerente de Gestión de Refinación.

Resumen: El caso de uso se inicia cuando el Analista de Gestión de Refinación solicita enviar el Resultado Mensual de Refinación Consolidado a los entes relacionados (Ministerio de Energía y Petróleo, Superintendente de Refinerías, Gerencia de Finanzas, Comercio y Suministro). El Gerente de Gestión de Refinación aprueba el Resultado Mensual Consolidado de Refinación y luego el Analista de Gestión de Refinación lo envía a los entes implicados.

Capítulo 3 Presentación de la solución propuesta

CUN5- Certificar Documentos Consolidados.

Actores: Analista de Gestión de Refinación.

Trabajadores: Analista de Gestión de Refinación.

Resumen: El Caso de Uso se inicia cuando el Analista de Gestión de Refinación informa al Gerente de Gestión de Refinación que los documentos se encuentran consolidados, el Gerente de Gestión de Refinación los evalúa, si no contiene imprecisiones ni nada que corregir los aprueba. En el caso de contener errores se ejecuta el caso de uso " Rectificar Documentos".

CUN6- Rectificar Documentos.

Actores: Analista de Gestión de Refinación.

Trabajadores: Analista de Gestión de Refinería.

Resumen: El caso de uso comienza cuando el Analista de Gestión de Refinación analiza los errores encontrados en los documentos revisados, si son rectificables por él los rectifica y si no se apoya de las refinerías.

3.2.3.5. Gestión de Resultados Mensuales y Evaluación de Indicadores.

CUN1- Consolidar Resultado Mensual de Refinación.

Actores: Evento Resultados Mensual de Refinación.

Trabajadores: Analista de Gestión de Refinación. Documento Máster.

Resumen: El caso de uso se inicia cuando se reciben todos los documentos de las refinerías, la información se introduce por el Analista de Gestión en un Sistema Documento Máster (Excel) y este Sistema mediante un conjunto de fórmulas elimina las transferencias entre refinerías y elabora el consolidado. Se obtiene como resultado un documento consolidado de los resultados del Balance Volumétrico de Refinación.

CUN2- Emitir Resultado Mensual de Refinación Consolidado.

Actores: Analista de Gestión de Refinación.

Trabajadores: Gerente de Gestión de Refinación.

Resumen: El caso de uso se inicia cuando el Analista de Gestión de Refinación solicita enviar el Resultado Mensual de Refinación Consolidado a los entes relacionados (Ministerio de Energía y Petróleo, Superintendente de Refinerías, Gerencia de Finanzas, Comercio y Suministro). El Gerente de Gestión de Refinación aprueba el Resultado Mensual Consolidado de Refinación y luego el Analista de Gestión de Refinación lo envía a los entes implicados.

CUN3- Emitir Resultados de Refinación de un Período.

Actores: Evento Resultados Refinación.

Trabajadores: Gerente de Gestión de Refinación. Analista de Gestión de Refinación.

Resumen: El caso de uso se inicia cuando los Resultados de Refinación de un Período ya están certificados, El Gerente de Gestión de Refinación lo presenta al Comité Operacional, si aparece alguna imprecisión en la presentación, el Gerente de Gestión de Refinación se las informa al Analista de Gestión de Refinación para que la arregle.

CUN4- Evaluar la Variación de Indicadores.

Actores: Evento Resultados Refinación.

Trabajadores: Evento Resultados Refinación.

Resumen: Este caso de uso se inicia una vez que se tiene el valor real y el plan de un indicador, permite hacer una evaluación cualitativa de la variación de los indicadores.

3.3. Requerimientos Funcionales.

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del sistema, es decir definen las funciones que el sistema será capaz de realizar.

R1. El Sistema debe permitir gestionar los roles y permisos.

Descripción: El sistema debe permitir al usuario con el rol de administrador gestione los roles y permisos de los usuarios de la aplicación.

R1.1. Insertar rol.

Descripción: El sistema debe permitir al usuario con el rol de administrador insertar un nuevo rol.

R1.2. Modificar rol.

Descripción: El sistema debe permitir al usuario con el rol de administrador modificar un rol.

R1.3. Eliminar roles existentes.

Descripción: El sistema debe permitir que el usuario con el rol de administrador pueda realizar la acción de eliminar roles existentes.

R1.4. Crear nuevo permiso.

Descripción: El sistema debe permitir al usuario con el rol de administrador crear un nuevo permiso.

R1.5. Eliminar permisos existentes.

Descripción: El sistema debe permitir que el usuario con el rol de administrador pueda realizar la acción de eliminar permisos existentes.

R1.6. Asignar Permisos.

Descripción: El sistema debe permitir al usuario con el rol de administrador asignar los permisos.

Capítulo 3 Presentación de la solución propuesta

R2. El sistema debe permitir cambiar la contraseña del administrador local.

Descripción: El sistema debe permitir al usuario con el rol de súper administrador cambiar su contraseña cuando este lo desee.

R3. Permitir gestionar Indicadores.

Descripción: El sistema debe permitir el manejo de información referente a los indicadores, de manera dinámica.

R3.1. Permitir insertar un nuevo Indicador.

Descripción: El sistema debe permitir la posibilidad de adicionar un nuevo indicador.

R3.2. Permitir modificar un Indicador.

Descripción: El Sistema debe permitir modificar un indicador que haya sufrido cambios en sus parámetros o en su nombre.

R3.3. Permitir eliminar un Indicador.

Descripción: Si ya no se necesita un indicador, el sistema debe brindar la posibilidad de eliminarlo.

R3.4. Crear Fórmula del Indicador.

Descripción: El sistema debe permitir al usuario con el rol de especialista crear la formula del indicador.

R3.5. Crear Rango de Evaluación del Indicador.

Descripción: El sistema debe permitir al usuario con el rol de especialista crear el rango de evaluación del indicador.

R3.6. Crear la tendencia del Indicador.

Descripción: El sistema debe permitir al usuario con el rol de especialista de crear la e tendencia del Indicador.

R3.7. Definir los tipos de Gráficos para el indicador.

Descripción: El sistema debe permitir al usuario con el rol de especialista definir los tipos de gráficas para visualizar la información de un indicador.

R4. El sistema debe permitir gestionar refinерías.

Descripción: El sistema debe permitir gestionar las refinерías de PDVSA.

R4.1. Crear nueva refinерía.

Descripción: El sistema debe permitir al usuario con el rol de especialista crear una nueva refinерía para el circuito de PDVSA.

R4.2. Modificar refinерía.

Descripción: El sistema debe permitir al usuario con el rol de especialista modificar la información de una refinерía del circuito de PDVSA.

R4.3. Definir tipo de refinерía.

Capítulo 3 Presentación de la solución propuesta

Descripción: El sistema debe permitir al usuario con el rol de especialista definir el tipo de la refinería, o sea, de tipo simple o complejo.

R4.4. Definir operatividad de la refinería.

Descripción: El sistema debe permitir al usuario con el rol de especialista modificar la operatividad de una refinería.

R5. El sistema debe permitir gestionar los nomencladores.

Descripción: El sistema debe permitir al usuario con el rol de especialista que gestione los nomencladores.

R5.1. Insertar Nomencladores.

Descripción: El sistema debe permitir al usuario con el rol de especialista insertar un nuevo nomenclador.

R5.2. Eliminar Nomencladores.

Descripción: El sistema debe permitir al usuario con el rol de especialista, eliminar un nomenclador específico.

R5.3. Modificar Nomencladores.

Descripción: El sistema debe permitir al usuario con el rol de especialista, modificar los datos de un nomenclador seleccionado.

R5.4. Visualizar Nomencladores.

Descripción: El sistema debe permitir al usuario con el rol de especialista, visualizar los nomencladores que están almacenados en la Base de Datos del sistema.

R6. El sistema debe permitir al usuario con el rol de especialista que gestione las plantas de las refinerías.

Descripción: El sistema debe permitir que el rol del usuario indicado pueda registrar, modificar, y eliminar los datos referentes a los perfiles de plantas.

R6.1. Insertar Planta.

Descripción: El sistema debe permitir al usuario con el rol de especialista que registre una nueva planta de refinería.

R6.2. Modificar Planta.

Descripción: El sistema debe permitir al usuario con el rol de especialista que modifique una planta de refinería existente en el sistema.

R.6.3. Eliminar Planta.

Descripción: El sistema debe permitir al usuario con el rol de especialista que elimine una planta de refinería existente.

R.7. El sistema debe permitir al usuario con el rol de administrador gestionar los roles asociados a los usuarios.

Descripción: Una de las facilidades que debe brindar el sistema es permitir al usuario con el rol indicado asignar y eliminar roles asociados a usuarios.

R.7.1. Asignar rol o roles a un usuario de la aplicación.

Descripción: El sistema debe permitir al usuario con el rol de administrador asignar uno o varios roles de los existentes a los usuarios de la aplicación.

R.7.2. Eliminar rol o roles a un usuario de la aplicación.

Descripción: El sistema debe permitir al usuario con el rol de administrador eliminarle uno o varios roles de los que tiene asignados un usuario de la aplicación.

3.4. Requerimientos No Funcionales.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares, o sea los requerimientos que expresan las propiedades o cualidades que debe tener el sistema, que hacen al producto atractivo, usable, rápido y confiable.

Usabilidad.

– Preparar a los Administradores en la gestión de Roles y Permisos.

Descripción: Esta acción específica del sistema llega a un alto grado de complejidad, se debe preparar un curso para instruir a los que trabajarán con estas funcionalidades.

Rendimiento.

– El sistema debe responder en un tiempo relativamente rápido a las peticiones del usuario (menos de 5 segundos).

Descripción: Teniendo en cuenta que el sistema no es de tiempo real y no necesita de respuestas de milisegundos se plantea éste como un tiempo máximo estimado el cual debe ser mucho menor, pero si se encuentra en estos límites no debe afectar el funcionamiento del sistema.

– El sistema debe permitir conectados trabajar concurrentemente 200 usuarios como máximo.

Descripción: La cantidad de usuarios que se encuentran en PDVSA que usarían el sistema no deben pasar esa cifra.

– El sistema necesita un servidor de base de datos de 8 gigas de RAM.

Descripción: Para ganar en velocidad a la hora del trabajo con datos.

– El sistema necesita un servidor de WEB de 8 gigas de RAM.

Descripción: Para posibilitar un mejor tiempo de respuesta de las peticiones a las paginas servidoras.

– El sistema necesita PCs clientes con 256 Mb de RAM y un navegador WEB que soporte java script.

Capítulo 3 Presentación de la solución propuesta

Descripción: Hay funcionalidades que están desarrolladas en java script por lo que es imprescindible que las PCs clientes lo soporten.

Restricciones de diseño.

- El sistema debe guiarse por las pautas de diseño de la Intranet de PDVSA.

Descripción: El sistema va a ser accedido desde la intranet, además que como política de PDVSA todos los sistemas deben estar integrados a la intranet.

Interfaces de Usuarios.

- Interfaz de interacción con el usuario.

Descripción: Interfaces amigables, fáciles de interactuar con ellas.

Requerimientos de Hardware.

Servidor Web.

- Hardware de la estación de trabajo del servidor Web, donde se ejecutará el sistema.

Descripción: Procesador Pentium D 2x2 cache. 3.2 GHz, Memoria RAM >= 4 – 8 GB.

Servidor de Bases de Datos

- Hardware de la estación de trabajo servidor de Base de Datos.

Descripción: Procesador Pentium D 2x2 cache. 3.2 GHz, Memoria RAM 4 – 8 GB. Almacenamiento en discos SCSI en espejo y capacidad igual o superior a los 120 GB cada disco. Tecnología de respaldo de datos históricos.

Cliente Web.

- Hardware de la estación de trabajo del cliente.

Descripción:

Procesador Pentium 3 (o superior). Memoria RAM mínima de 256 MB. Procesador Pentium 3 (o superior). Memoria RAM mínima de 256 MB.

Requerimientos de Software.

PC Cliente.

- Software instalado en la estación de trabajo del cliente.

Descripción: Sistema Operativo tanto Windows (win9.x o versión superior) como Linux (cualquiera de sus distribuciones).

El Navegador Web compatible con HTML 2.0 y CSS, podrá ser Netscape 3 (o superior), Internet Explorer 4.2 (o superior) y compatibles.

PC Servidor Web.

- Software instalado en el Servidor Web.

Descripción: Servidor Web Apache.

Servidor de Bases de Datos.

- Software instalado en el Servidor de Base de datos.

Descripción: Servidor de Bases de Datos PostGreSQL.

Seguridad.

- Seguridad del sistema.

Descripción: Confidencialidad: La información manejada por el sistema esta protegida de acceso no autorizado y divulgación.

Integridad: la información manejada por el sistema es objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma es considerada igual a la fuente o autoridad de los datos. Se incluye también mecanismos de chequeo de integridad y realización de auditorias por personal calificado de la entidad.

Disponibilidad: los usuarios autorizados (autenticados por dominio y según su roll) se les garantizarán el acceso a la información, los dispositivos o mecanismos utilizados para lograr la seguridad, no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

3.5. Descripción del Sistema Propuesto.

En el siguiente subepígrafe se presenta una propuesta de solución para el módulo de administración del sistema SACGIR, se expone una descripción de los actores y de los casos de uso del sistema. Mediante el siguiente Diagrama de Casos de Uso del Sistema se muestra la relación entre los casos de uso y los actores.

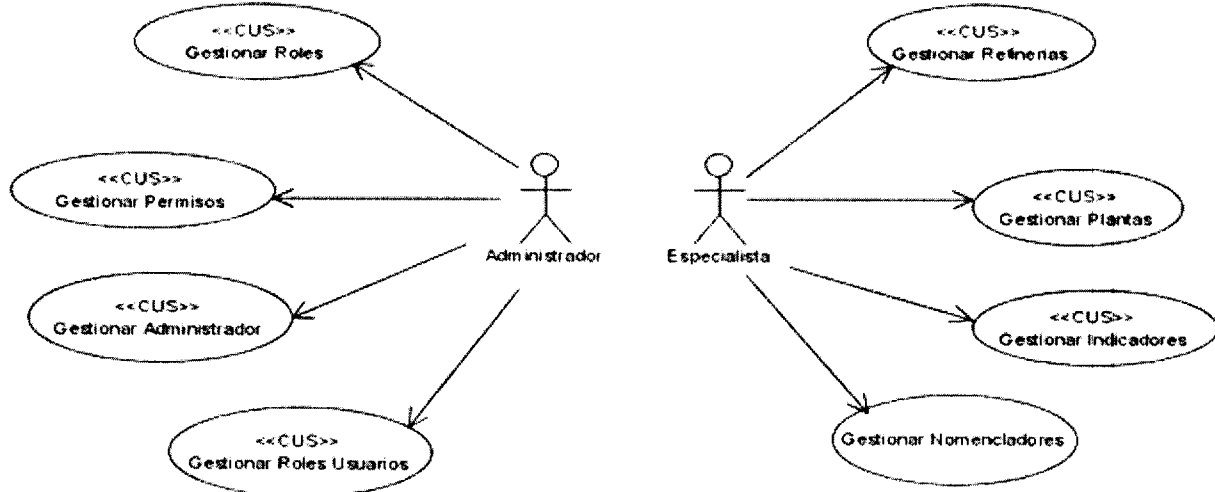


Figura 5. Diagrama de Casos de Uso del Sistema.

Capítulo 3 Presentación de la solución propuesta

3.5.1. Descripción de los actores.

El sistema SACGIR es una aplicación de gestión, todas las personas que utilicen el software serán especialistas en el tema. Teniendo en cuenta los requerimientos todas las personas que interactúan con el sistema tendrían que autenticarse antes para poder interactuar con la aplicación y recibir los beneficios que esta brinda, en dependencia de sus permisos. A continuación los actores del sistema y su justificación.

Actor	Descripción
Administrador	Este rol tiene permisos asignados para gestionar usuarios (Administradores), lo cual permite realizar acciones de inserción, eliminación y modificación de los mismos. Además se encarga de gestionar los roles de los usuarios que interactuarán con el sistema, así como los permisos que se corresponden con dichos roles. Es su responsabilidad asignarle los diferentes roles definidos a los usuarios del sistema.
Especialista	Este Rol es asignado por el Administrador del sistema, dicho Rol tiene como objetivo gestionar la información de: refineries, indicadores y nomencladores, según los permisos definidos por el administrador en cada caso.

Tabla 3. Descripción de los Actores del Sistema.

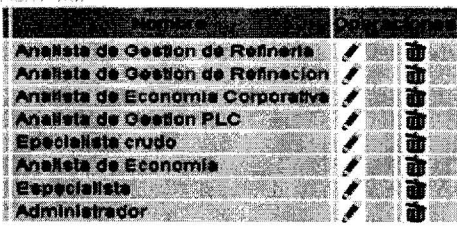
3.5.2. Descripciones de los Casos de Uso del Sistema.

A continuación se muestran las descripciones de los CUS críticos, los casos de uso secundarios, se pueden encontrar en el Anexo 2.

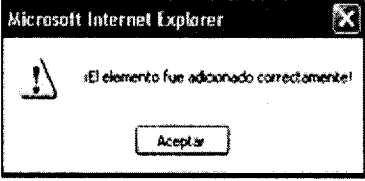
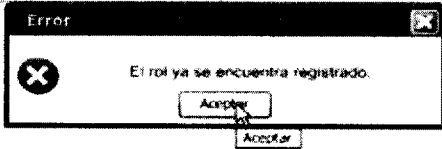
CUS 1 Gestionar Roles

Caso de Uso:	Gestionar Roles
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador solicita insertar, editar o eliminar roles y concluye cuando ha sido realizada la solicitud.
Precondiciones:	El usuario debe estar previamente autenticado como Administrador.


Capítulo 3 Presentación de la solución propuesta

Referencias	R1, R1.1, R1.2, R1.3
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
<p>1. El Administrador selecciona la opción que desea realizar:</p> <ul style="list-style-type: none"> • Insertar Rol, ir a sección "Insertar Rol". • Editar Rol, ir a sección "Editar Rol". • Eliminar Rol, ir a sección "Eliminar Rol". 	<p>2. El sistema va a la sección insertar, editar o eliminar rol, (según haya sido seleccionado en paso 1).</p>
Gestionar Roles	
	
Sección "Insertar Rol"	
Actor	Sistema
	<p>1. El Sistema muestra los controles que permiten introducir los datos Nombre (textField), Permisos (select) y permisos Asociados (select) del nuevo rol.</p>
<p>2. El Administrador introduce el nombre del nuevo rol que desea agregar, selecciona los permisos que se deben corresponder con dicho rol y selecciona la opción de Aceptar.</p>	<p>3. El Sistema verifica que ha sido introducido el nombre del nuevo rol y otorgado al menos un permiso ha dicho rol.</p>
	<p>4. El Sistema verifica que el nombre de rol introducido no esté registrado en la Base de Datos.</p>
	<p>5. El Sistema inserta el nuevo rol en la Base de Datos y los permisos otorgados al mismo.</p>

Capítulo 3 Presentación de la solución propuesta

	6. El sistema muestra un mensaje indicando que ha guardado los datos satisfactoriamente.
Adicionar un nuevo Rol	
<p>Nombre del nuevo Rol <input type="text"/></p> <p>Permisos: <input type="text"/> exportar plantas eliminar precio certificar reporte</p> <p>Permisos Otorgados <input type="text"/></p> <p style="text-align: center;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p>	
	
Flujos Alternos	
Actor	Sistema
2.1. El Administrador selecciona la opción de Cancelar(botón)	2.2. El Sistema regresa a la interfaz Anterior "Gestionar Roles".
	4.1 El Sistema muestra un mensaje de error indicando que el nombre del rol se encuentra registrado en la Base de Datos.
	
Sección "Modificar Rol"	
Actor	Sistema
	1. El Sistema muestra un listado (table) que contiene los roles almacenados en la BD, que permite seleccionar un rol.
2. El Administrador selecciona el rol que desea Editar (botón).	3. El Sistema muestra controles Nombre (textField), Permisos y Permisos Otorgados, con la información del Rol especificado, controles de selección

Capítulo 3 Presentación de la solución propuesta

	(ListBox).
4. El Administrador cambia los datos de los campos correspondientes a Nombre y/o Permisos y selecciona la opción de Modificar (botón).	5. El Sistema verifica si los datos introducidos son correctos.
	6. El Sistema actualiza los nuevos datos en la Base de datos.
	7. El sistema muestra un mensaje indicando que ha guardado los datos satisfactoriamente.
<p>Modificar Rol</p> <p>Nombre del Rol: <input type="text" value="Analista de Gestión de"/></p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px;"> <p>Permisos</p> <ul style="list-style-type: none"> exportar plantas certificar reporte </div> <div style="text-align: center;"> <p>>></p> <p><<</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>Permisos Otorgados</p> <ul style="list-style-type: none"> eliminar precio </div> </div> <p style="text-align: center;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> 	
Flujos Alternos	
Actor	Sistema
4.1 El Administrador selecciona la opción de Cancelar(botón)	4.2. El Sistema muestra la interfaz de Gestionar Rol.
Sección "Eliminar Rol"	
Actor	Sistema
	1. El Sistema muestra un control de selección (Select) en el que el Administrador seleccionará el nombre del rol que quiere eliminar.
2. El Administrador selecciona el nombre del rol que desea eliminar y selecciona la opción Buscar (botón).	3. El Sistema muestra los datos (Nombre, Permisos) del rol seleccionado.
4. El Administrador selecciona la opción de Eliminar.	5. El Sistema muestra un cartel de confirmación para verificar que el usuario

Capítulo 3 Presentación de la solución propuesta

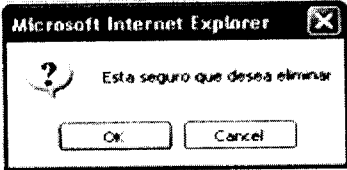






6. El Administrador selecciona la opción Aceptar eliminar el rol especificado.	desea realizar la acción Eliminar. 7. El Sistema elimina el rol especificado en la Base de Datos.
	
Flujos Alternos	
Actor	Sistema
4.1. El Administrador selecciona la opción de Cancelar(botón)	4.2. El Sistema muestra la interfaz de Gestionar Rol.
Pos condiciones	Quedan Insertados, Modificados y Eliminados los datos de los Roles en el Sistema.

Tabla 4. Descripción extendida del CUS Gestionar Roles.

CUS 2 Gestionar Permisos

Caso de Uso:	Gestionar Permisos
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador solicita insertar, editar o eliminar un permiso, y concluye cuando ha sido realizada la solicitud.
Precondiciones:	El usuario debe estar previamente autenticado como Administrador.
Referencias	R1,R1.4, R1.5,R1.6
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
1. El Administrador selecciona la opción que desea realizar: <ul style="list-style-type: none"> • Nuevo Permiso, ir a la sección "Insertar Permiso". • Eliminar Permiso, ir a sección "Eliminar Permiso." • Editar Permiso, ir a sección "Editar Permiso". 	2. El sistema va a la sección insertar, editar o eliminar, (según haya sido seleccionado en paso 1).

Capítulo 3 Presentación de la solución propuesta

GESTIONAR PERMISOS				
Nombre	Acción	Objeto	Refinería	
exportar plantas	actualizar	n_tipo_planta (entidad)	CRN	 
eliminar precio	eliminar	precio (entidad)	ISLA	 
certificar reporte	certificar	reporte (modulo)	ELP	 

Nuevo Permiso

Sección "Insertar Permisos"


Actor	Sistema
	1. El Sistema muestra los controles para ingresar una Acción (selectTag), Objeto (select), Refinería (select) y Nombre (textField) necesarios para introducir un nuevo Permiso.
2. El Administrador introduce los datos (acción, objeto, refinería y nombre) necesarios para insertar un nuevo Permiso y selecciona la opción Adicionar.	3. El Sistema verifica que han sido seleccionados correctamente los datos del paso anterior teniendo en cuenta la combinación del Objeto, Acción y Refinería seleccionados.
	4. El Sistema almacena el nuevo permiso en la Base de Datos.
	5. El Sistema muestra un mensaje indicando que la acción se ha realizado satisfactoriamente.

ADICIONAR PERMISO

Acción: << Seleccione >> ▼ Objeto: << Seleccione >> ▼ Refinería: << Seleccione >> ▼

Nombre: _____

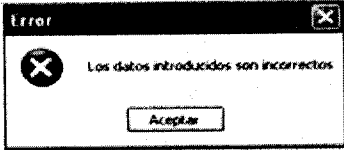
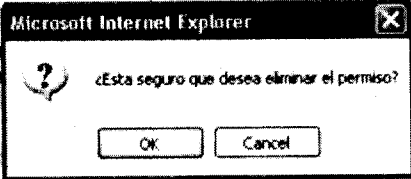
Adicionar Cancelar



Flujo Alternativo

2.1 El Administrador selecciona la opción de Cancelar (botón).	2.2. El Sistema regresa a la Interfaz anterior: "Gestionar Permisos".
--	---

Capítulo 3 Presentación de la solución propuesta

	3.1. El Sistema muestra un mensaje de error indicando que los datos (acción, objeto, refinería y nombre) introducidos están incorrectos y regresa a la acción 2 de esta sección.
	
Sección "Eliminar Permisos"	
	1. El Sistema muestra un listado de los permisos en el que se especifica Nombre, Acción, Objeto y Refinería mediante una tabla, donde se selecciona el permiso que se desea eliminar.
2. El Administrador selecciona la opción Eliminar (botón).	3. El Sistema muestra un mensaje de advertencia indicando si se desea eliminar o no el permiso seleccionado.
4. El Administrador selecciona la opción de Aceptar para eliminar el permiso.	5. El Sistema elimina el permiso de la Base de Datos.
	
Flujo Alterno	
4.1 El Administrador selecciona la opción de Cancelar (botón).	4.2 El Sistema regresa a la Interfaz anterior: ir a Flujo Normal de Eventos.
Sección "Editar Permisos"	
	1. El Sistema muestra un listado con los permisos existentes (Table), que brinda la posibilidad de seleccionar el permiso que se desea modificar.
2. El Administrador selecciona el permiso que desea Editar (botón).	3. El Sistema muestra los controles Acción (select), Objeto (select), Refinería (select) y Nombre (textField), con los datos del elemento señalado para permitir modificar los mismos.

Capítulo 3 Presentación de la solución propuesta

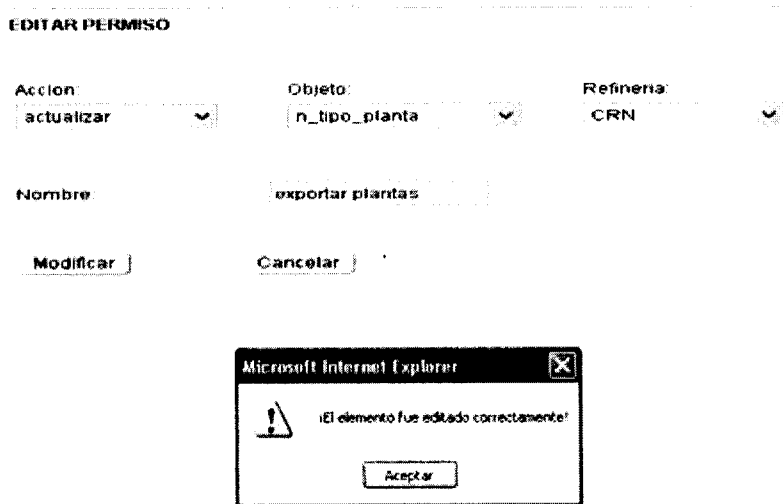
4. El Especialista modifica acción, objeto, refinería y nombre) según desee y selecciona la opción de Aceptar.	5. El Sistema verifica que los datos que fueron modificados en el paso anterior sean correctos.
	6. El Sistema actualiza los datos introducidos en la Base de Datos.
	7. El Sistema muestra un mensaje indicando que el permiso ha sido editado satisfactoriamente.
 <p>The screenshot shows a web form titled "EDITAR PERMISO". It contains three dropdown menus: "Acción" (set to "actualizar"), "Objeto" (set to "n_tipo_planta"), and "Refinería" (set to "CRN"). Below these is a text input field for "Nombre" containing "exportar plantas". At the bottom of the form are two buttons: "Modificar" and "Cancelar". Below the form is a screenshot of a Microsoft Internet Explorer dialog box with a warning icon and the text "¡El elemento fue editado correctamente!". There is an "Aceptar" button at the bottom of the dialog box.</p>	
Flujo Alterno	
4.1 .El Administrador selecciona la opción de Cancelar (botón).	4.2 . El Sistema regresa a la Interfaz anterior.
Pos condiciones	Quedan Insertados, Editados y Eliminados los Permisos en el Sistema.

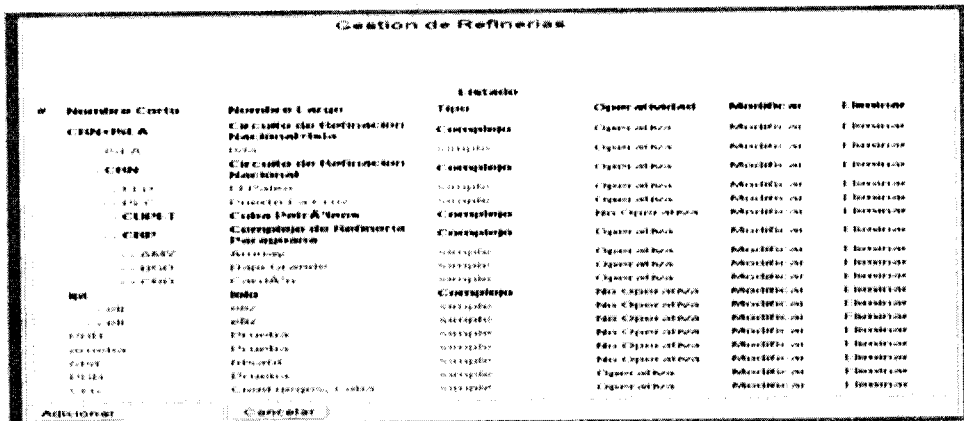
Tabla 5. Descripción extendida del CUS Gestionar Permisos.

CUS 3 Gestionar Refinerías

Caso de Uso:	Gestionar Refinerías
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista solicita insertar nuevas refineries, modificar las existentes o eliminarlas y concluye cuando ha sido realizada la solicitud.
Precondiciones:	El usuario debe estar previamente autenticado como Especialista.
Referencias	R4, R4.1, R4.2, R4.3, R4.4
Prioridad	Crítico
Flujo Normal de Eventos	

Capítulo 3 Presentación de la solución propuesta


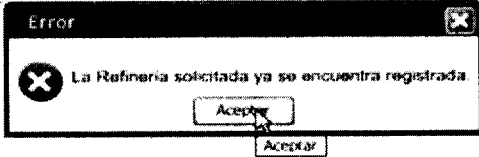
Actor	Sistema
<p>1. El Especialista selecciona la opción que desea realizar:</p> <ul style="list-style-type: none"> • Opción de Añadir Refinería. • Opción Modificar Refinería. • Opción Eliminar Refinería. • Opción de Operatividad. 	<p>2. El sistema va a la sección insertar, editar o eliminar, (según haya sido seleccionado en paso 1).</p>



Sección "Añadir Refinería"

Actor	Sistema
<p>2. El Especialista introduce el nombre de la refinería y selecciona el tipo de refinería, así como si desea agruparla en un complejo o no y selecciona la opción Aceptar (botón).</p>	<p>1. El Sistema muestra un conjunto de controles que permiten especificar el nombre de la nueva refinería (Corto y Largo) (textField) que se desea añadir así como seleccionar el tipo de Refinería al cual pertenece la misma (select) y la opción de agruparla en un complejo (select).</p>
<p>3. El Especialista introduce el nombre de la refinería y selecciona el tipo de refinería, así como si desea agruparla en un complejo o no y selecciona la opción Aceptar (botón).</p>	<p>3. El Sistema verifica que la refinería que se desea añadir no esté registrada en la Base de Datos y que los datos introducidos por el usuario sean correctos.</p>
<p>4. El Especialista introduce el nombre de la refinería y selecciona el tipo de refinería, así como si desea agruparla en un complejo o no y selecciona la opción Aceptar (botón).</p>	<p>4. El Sistema guarda la nueva refinería en la Base de Datos.</p>
<p>5. El Especialista introduce el nombre de la refinería y selecciona el tipo de refinería, así como si desea agruparla en un complejo o no y selecciona la opción Aceptar (botón).</p>	<p>5. El sistema muestra un mensaje indicando que se han guardado los datos satisfactoriamente.</p>

Capítulo 3 Presentación de la solución propuesta

Adicionar Refinería	
Nombre Corto:	<input type="text"/>
Nombre Largo:	<input type="text"/>
Tipo de Refinería:	<<seleccione>> ▼
Agrupar en el Complejo:	<< seleccione >> ▼
	<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>
	
Flujos Alternos	
Actor	Sistema
2.1. El Especialista selecciona la opción Cancelar (botón)	2.2. El sistema regresa a la interfaz de inicio del CU: Gestionar Refinería.
	3.1. El Sistema muestra un mensaje de error indicando que la refinería especificada ya se encuentra registrada en el sistema y regresa a la acción 2 de la sección "Añadir Refinería".
	
Sección "Modificar Refinería"	
Actor	Sistema
	1. El Sistema muestra un Listado con las refinerías existentes (Table) que brinda la posibilidad de seleccionar un elemento.
2. El Especialista selecciona la refinería que desea modificar.	3. El Sistema muestra los controles nombre Corto (textField), nombre Largo (textField), agrupar en complejo (select) con los datos del elemento señalado para permitir modificar los mismos, muestra además el tipo de refinería, aunque no permite modificar el mismo.

Capítulo 3 Presentación de la solución propuesta

<p>4. El Especialista modifica el nombre Largo y/o Corto, Agrupar en Complejo, de la refinería, según desee y selecciona la opción de Aceptar (botón).</p>	<p>5. El Sistema verifica si los datos introducidos son correctos.</p>
	<p>6. El Sistema actualiza los datos introducidos en la Base de datos.</p>
	<p>7. El Sistema muestra un mensaje indicando que ha guardado los datos satisfactoriamente.</p>

Modificar Refinería

Nombre Corto:

Nombre Largo:

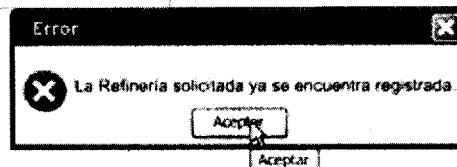
Tipo de Refinería: **complejo**

Agrupar en el Complejo: ▼



Flujos Alternos

Actor	Sistema
<p>2.1. El Especialista selecciona la opción Cancelar (botón).</p>	<p>2.2. El sistema regresa a la pantalla de inicio del CU: Gestionar Refinería.</p>
	<p>5.1. El Sistema muestra un mensaje de error en caso de que el nombre de la Refinería ya se encuentre registrado en la Base de Datos.</p>

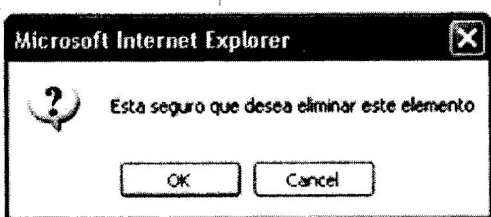


Sección "Eliminar Refinería"

Actor	Sistema
	<p>1. El Sistema muestra un listado de las refinerías</p>

Capítulo 3 Presentación de la solución propuesta

	en el que se especifica Nombre Corto y Largo, Tipo y Operatividad y Refinería mediante una tabla, donde se selecciona la refinería que se desea eliminar.
2. El Especialista selecciona la opción Eliminar (botón).	3. El Sistema muestra un cartel de confirmación para verificar que el usuario desea realizar la acción, en caso de que la refinería sea de tipo Complejo informa además, las refinerías que contiene.
4. El Especialista selecciona la opción de Aceptar para eliminar la refinería.	5. El Sistema elimina la refinería de la Base de Datos.



Flujos Alternos

Actor	Sistema
4.1. El Especialista selecciona la opción Cancelar (botón)	4.2. El sistema regresa a la interfaz de inicio del CU: Gestionar Refinería.

Sección "Operatividad _ Refinería"

Actor	Sistema
	1. El Sistema muestra un listado de las refinerías en el que se especifica Nombre Corto y Largo, Tipo y Operatividad y Refinería mediante una tabla, donde se selecciona la refinería a la que se desea cambiar la operatividad.
2. El Especialista selecciona la refinería que desea poner operativas o no y selecciona la opción Operativa.	3. El Sistema muestra un mensaje de confirmación para verificar que el usuario desea realizar la acción.
4. El Especialista selecciona la opción de Aceptar para cambiar la operatividad de la refinería.	5. El Sistema Guarda en la Base de Datos los datos correspondientes a la operatividad de las refinerías.

Capítulo 3 Presentación de la solución propuesta

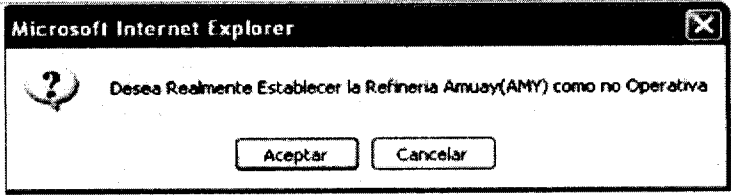
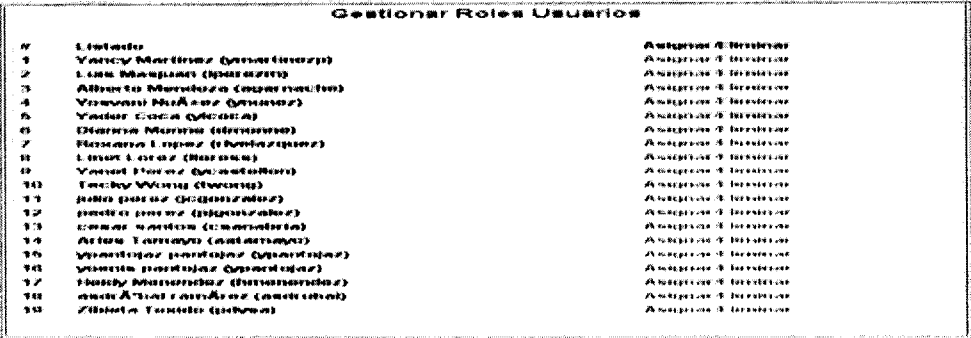
	
Flujos Alternos	
Actor	Sistema
4.1. El Especialista selecciona la opción Cancelar (botón).	4.2. El sistema regresa a la interfaz de inicio del CUS Gestionar Refinería.
Poscondiciones	Quedan Insertados, Modificados y Eliminados los datos de los Refinerías en el Sistema.

Tabla 6. Descripción extendida del CUS Gestionar Refinerías.

CUS 4 Gestionar Roles Usuarios

Caso de Uso:	Gestionar Roles Usuarios	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el Administrador solicita asignar_ eliminar roles asociados a usuarios, y concluye cuando ha sido realizada la operación.	
Precondiciones:	El usuario debe estar previamente autenticado como Administrador.	
Referencias	R9, R9.1, R9.2	
Prioridad	Critico	
Flujo Normal de Eventos		
Actor	Sistema	
1. El Administrador selecciona la opción de Gestionar Rol Usuario.	2. El sistema muestra un listado (table) que contiene todos los usuarios del sistema.	
3. El Administrador selecciona la opción de Asignar/Eliminar Rol a Usuario.	4. El Sistema va a la sección Asignar/Eliminar Rol a Usuario.	
		

Capítulo 3 Presentación de la solución propuesta

Sección "Asignar/Eliminar Rol Usuario "

Actor	Sistema
	1. El Sistema muestra un listado (table) que contiene los nombres de los usuarios del sistema) necesarios para localizar al usuario que se le quiere asignar o eliminar el/los roles.
2. El Administrador selecciona el usuario al que desea Asignar o Eliminar algún rol.	3. El Sistema muestra los controles (TextArea [izquierda]: Roles registrados en base de datos, select [derecha]: rol/roles asignados al usuario) necesarios para asignar el/los rol(es) al usuario localizado anteriormente.
5. El Administrador selecciona los roles que le desea asignar o eliminar al usuario.	6. El Sistema actualiza los controles (TextArea) Roles Registrados y Roles Otorgados.
7. El Administrador selecciona la opción Aceptar.	8. El Sistema guarda los datos seleccionados en la Base de datos.
	9. El sistema muestra un mensaje indicando que se han guardado los datos satisfactoriamente.

Asignar Rol a Usuario : pdvsa

Roles:

- Analista de Gestion de Refineria ▲
- Analista de Gestion de Refinacion
- Analista de Economia Corporativa
- Analista de Gestion PLC
- Especialista crudo ▼

<<

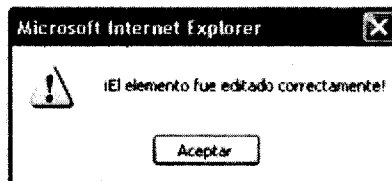
>>

Rol Otorgado:

Administrador

Aceptar

Cancelar



Flujos Alternos

Actor	Sistema
7.1 El Administrador selecciona la opción	7.2 El Sistema muestra la interfaz de

Capítulo 3 Presentación de la solución propuesta

de Cancelar.	Gestionar Roles Usuarios.
Pos condiciones	Se actualizaran en el sistema los roles de los usuarios manejados en el mismo.

Tabla 7. Descripción extendida del CUS Gestionar Roles Usuarios.

CUS 5 Gestionar Planta

Caso de Uso:	Gestionar Planta
Actores:	Especialista
Resumen:	El caso de uso se inicia cuando el Especialista solicita insertar, eliminar o modificar una Planta y concluye cuando ha sido realizada la solicitud.
Precondiciones:	El usuario debe estar previamente autenticado como Especialista.
Referencias	R6, R6.1, R6.2, R6.3, R6.4, R7, R8
Prioridad	Crítico

Flujo Normal de Eventos

Actor	Sistema
<p>1. El Especialista selecciona la opción que desea realizar:</p> <ul style="list-style-type: none"> • Insertar Planta, ir a la sección "Insertar nueva Planta". • Modificar Planta, ir a sección "Modificar" Planta. • Eliminar Planta, ir a sección "Eliminar" Planta. 	<p>2. El Sistema va a la sección insertar, modificar o eliminar, (según haya sido seleccionado en paso 1).</p>


GESTIONAR NPLANTAS:

millon	millon Jergo	rere	Amuay	/	
DST	Destiladora	FE	El Palto	/	
PLT	Planta Cacaguaca	FE	Amuay	/	
claudie	claudie_largo	rere	El Palto	/	
yoenis	Yoenis_Largo	FE	El Palto	/	
Cipos	Cianfuegos	FE	El Palto	/	

Administrar

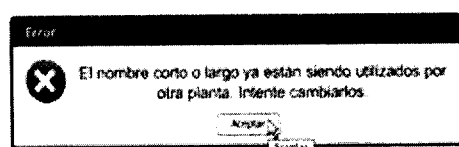
Flujo Alterno

Capítulo 3 Presentación de la solución propuesta

1.1. El Especialista selecciona la opción de Cancelar.	1.2. El Sistema muestra la pantalla que originó esta Sesión.
Sesión "Insertar Planta"	
Actor	Sistema
	1. El Sistema muestra los controles necesarios para introducir el nombre corto y largo (textField) de la Planta así como, el tipo de Planta de la cual forma parte (Select) y para seleccionar la refinería a la que pertenece (Select).
2. El Especialista introduce el Nombre Corto, Nombre Largo, selecciona el Tipo de Planta y la refinería a la que pertenece la Planta que se desea agregar y selecciona la opción de Adicionar.	3. El Sistema verifica que han sido introducidos correctamente los datos (nombre: Corto y Largo).
	4. El Sistema inserta la nueva Planta en la Base de Datos (nombre: Corto y Largo, Tipo de Planta, Refinería a la que pertenece).
	5. El sistema muestra un mensaje indicando que se han guardado los datos satisfactoriamente.
<div style="text-align: center;"> <p>ANADIR NUEVA PLANTA</p> <p>Nombre Corto <input type="text"/></p> <p>Nombre Largo <input type="text"/></p> <p>Tipo de Planta <input type="text" value="Selecione"/></p> <p>Refinería Tipo Simple <input type="text" value="Selecione"/></p> <p><input type="button" value="Cancelar"/> <input type="button" value="Adicionar"/></p> </div> <div style="text-align: center;">  </div>	
Flujo Alterno	
2.1. El Especialista selecciona la opción de Cancelar	2.2. El Sistema muestra la pantalla inicial: Gestionar Planta.

Capítulo 3 Presentación de la solución propuesta

	3.1 El Sistema muestra un mensaje de error indicando que los campos de los datos [nombres (corto y largo) y/o tipo de Planta] están vacíos. (Señala los controles con problemas en rojo e indica cual es el error).
	3.2 El sistema muestra un mensaje indicando que los nombres corto y/o largo introducidos ya existen en la Base de Datos.



Sección "Modificar Planta"

Actor	Sistema
	1. El Sistema muestra un Listado con las plantas existentes (table) que brinda la posibilidad de seleccionar un elemento.
2. El Especialista selecciona el elemento planta que desea modificar.	3. El Sistema muestra los Datos a Modificar: Nombre Largo (textfield), Nombre Corto (textfield), Tipos de Plantas (select) y Refinería Pertenece (select) de la Planta especificada.
4. El Especialista introduce los datos en dichos campos y selecciona la opción de Editar (botón).	5. El Sistema verifica si los datos (Nombre Largo, Nombre Corto, Tipos de Plantas y Refinería Pertenece) introducidos son correctos.
	6. El Sistema actualiza los nuevos datos en la Base de datos.
	7. El Sistema muestra un mensaje de que la operación se ha realizado satisfactoriamente y termina el caso de uso.

Capítulo 3 Presentación de la solución propuesta

EDITAR PLANTA

Nombre Corto:

Nombre Largo:

Tipo de Planta:

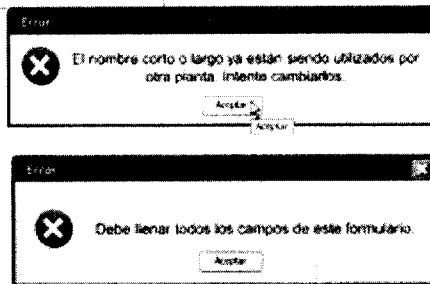
Refinería Tipo Simple:

Microsoft Internet Explorer

! ¡El elemento fue editado correctamente!

Flujo Alternativo

- | | |
|--|---|
| <p>2.1. El especialista selecciona la opción Cancelar (botón).</p> | <p>2.2. El Sistema regresa a la interfaz Gestionar Planta.</p> |
| | <p>3.1. El Sistema muestra un mensaje de error en caso de que el nombre corto o largo de la Planta ya se encuentre registrado en la Base de Datos.</p> |
| | <p>3.2. El Sistema muestra un mensaje de error indicando que los campos de los datos [nombres (corto y largo) y/o tipo de Planta] están vacíos. (Señala los controles con problemas en rojo e indica cual es el error).</p> |



Sección "Eliminar Planta"

Actor	Sistema
	<p>1. El Sistema muestra un mensaje de confirmación para verificar que el usuario desea realizar la acción. (Aceptar o Cancelar)</p>

Capítulo 3 Presentación de la solución propuesta

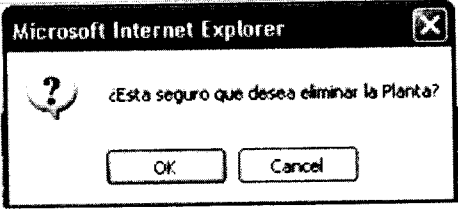
2. El Especialista selecciona la opción aceptar eliminar la planta localizada.	3. El Sistema elimina la Planta de la Base de Datos.
	4. El Sistema muestra un mensaje indicando que la operación se ha realizado satisfactoriamente y termina el caso de uso.
	
Flujo Alterno	
6.1. El Especialista selecciona la opción de Cancelar (botón).	6.2. El sistema muestra la interfaz Gestionar Planta.
Poscondiciones	Se actualizarán en el sistema la información de las plantas a ser manejadas por el mismo.

Tabla 8. Descripción extendida del CUS Gestionar Plantas.

3.6. Conclusiones.

En el presente capítulo, y sobre la propuesta de RUP para el desarrollo de software, se realizó una modelación del negocio para una mejor comprensión de la estructura y dinámica de los procesos que se desarrollan en el entorno para el cual se desarrolla la aplicación. Como resultado de esto se obtuvo un diagrama de casos de uso del negocio donde se definieron los actores y casos de uso del negocio así como las relaciones entre los mismos. Cada caso de uso fue descrito en formato expandido y mediante diagramas de actividad donde fueron sombreadas las actividades a automatizar por el sistema. Además se presenta el listado de los requisitos funcionales del módulo de administración del sistema SACGIR y los requisitos no funcionales del sistema en general. También se definió el diagrama de casos de uso del sistema con los actores y casos de uso del sistema y relaciones entre ellos, así como una descripción en formato expandido para cada caso de uso.

CAPÍTULO 4

Construcción de la solución propuesta

4.1. Introducción.

En el siguiente capítulo se presentan los diagramas de clases generados en el flujo de trabajo de Análisis y Diseño, se plantean los principios y estándares que fueron tenidos en cuenta para el desarrollo de la interfaz de la aplicación. Todo lo que unido al diseño de los diagramas de interacción y de los diagramas para el diseño de la Base de Datos forman parte de la construcción de la solución propuesta.

4.2. Diagramas de Clases.

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones, estos son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). (29)

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas. Los diagramas de clases también son la base para un par de diagramas relacionados: los diagramas de componentes y los diagramas de despliegue. Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. (29)

A continuación se muestran los diagramas de clases del análisis y diseño, así como las descripciones de las clases de estos diagramas del caso de uso representativo del sistema Gestionar Planta.

Los diagramas pertenecientes a los 6 casos de uso restantes se muestran en los Anexos 3 y 4.

Capítulo 4 Construcción de la solución propuesta

4.2.1. Diagrama de clases del análisis del CUS Gestionar Plantas.

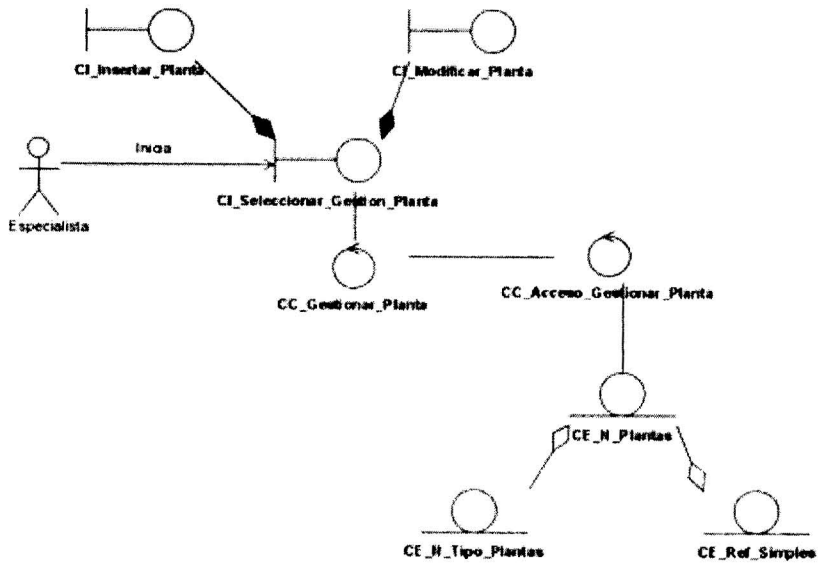


Figura 6. Diagrama de clases del análisis CU Gestionar Planta.

4.2.2 Diagrama de clases del diseño del CUS Gestionar Plantas.

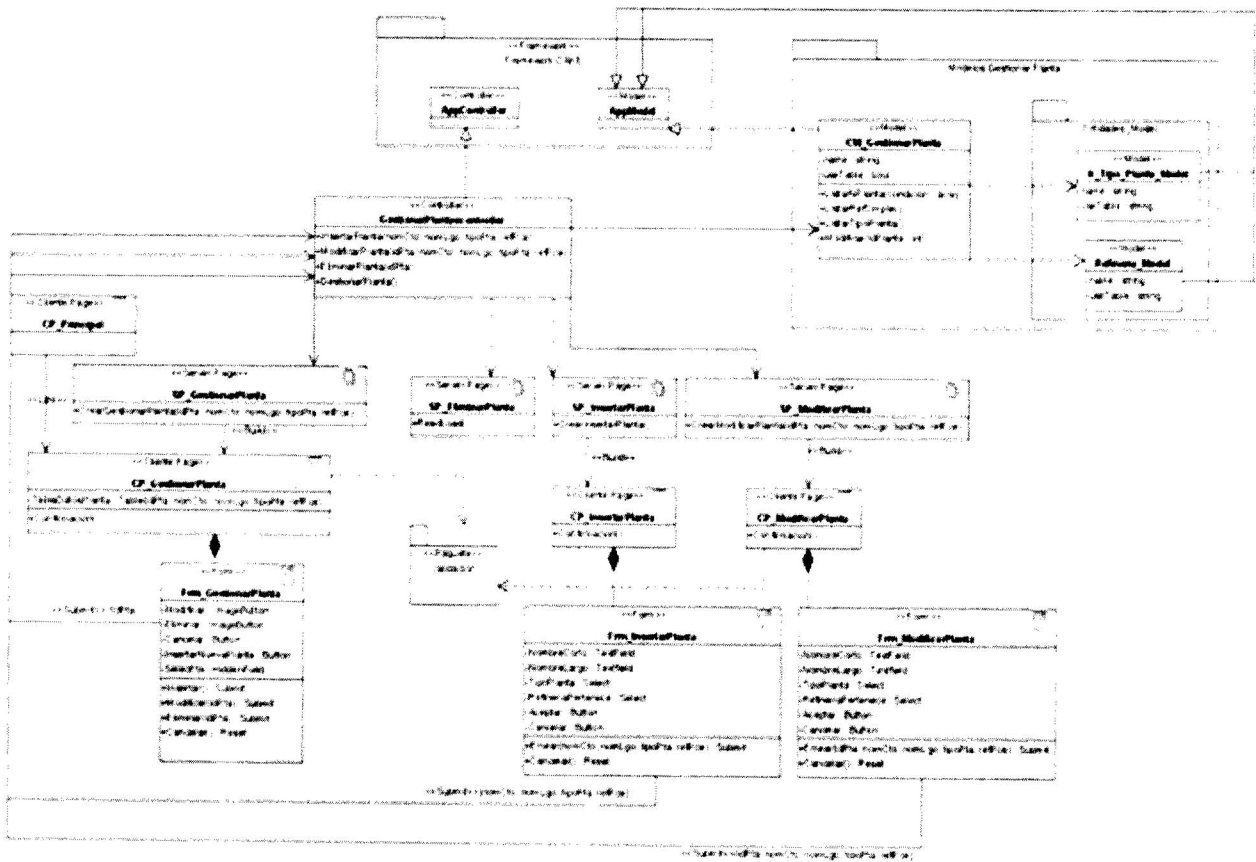


Figura 7. Diagrama de clases del diseño CU Gestionar Planta.

Capítulo 4 Construcción de la solución propuesta

Descripción de Clases del Diseño.

Capa Vistas.

Módulo Administración. Clase CP_Principal

Nombre de la clase:	CP_Principal
Tipo:	Client Page
Descripción:	Es la clase vista encargada de facilitarle al usuario seleccionar la opción de Gestionar Plantas.
Atributos	No aplica
Métodos	No aplica

Tabla 9. Clase CP_Principal.

Módulo Administración. Clase CP_GestionarPlanta

Nombre de la clase:	CP_GestionarPlanta
Tipo:	Client Page
Descripción:	Es la clase vista encargada de mostrar la relación de las plantas existentes en el Sistema con la información correspondiente a las mismas. La clase está compuesta por el formulario Frm_GestionarPlanta.
Atributos	<i>TablaDatosPlanta: Table(idPta, nomCto, nomLgo, tipoPta, refPce, Modificar, Eliminar):</i> Es el atributo que contiene: <i>idPta:</i> identificador de la Planta. <i>nomCto:</i> nombre corto que referencia a una Planta del Sistema. <i>nomLgo:</i> nombre completo de una Planta del Sistema. <i>tipoPta:</i> tipo de producción con la que se caracteriza la Planta. <i>refPce:</i> es la refinería a la que está asociada la Planta.
Métodos	<i>Confirmacion():</i> Esta función es la encargada de confirmar todas las acciones que realice el usuario en la interfaz.

Tabla 10. Clase CP_GestionarPlanta.

Módulo Administración. Clase Frm_GestionarPlanta

Nombre de la clase:	Frm_GestionarPlanta
Tipo:	Form
Descripción:	Es el formulario encargado de la gestión de las Plantas del Sistema. Está compuesto por los atributos: modificar, eliminar, insertar nueva planta, cancelar, seleccionar planta.

Capítulo 4 Construcción de la solución propuesta

Atributos	<p><i>Eliminar</i>: Es el atributo tipo botón que permite Eliminar la planta deseada del Sistema.</p> <p><i>Modificar</i>: Es el atributo tipo botón que permite ejecutar la acción de Modificar los datos correspondientes a una Planta.</p> <p><i>InsertarNuevaPlanta</i>: Es un atributo de tipo botón que permite ejecutar la acción para introducir una nueva Planta al Sistema.</p> <p><i>Cancelar</i>: Es un atributo de tipo botón que permite anular la acción de gestionar Plantas.</p> <p><i>SelectPta</i>: Es el atributo de tipo campo oculto que permite identificar una fila de la tabla.</p>
Métodos	<p><i>Eliminar(idPta)</i>: Es el método encargado de eliminar las Plantas de las lista.</p> <p><i>Modificar(idPta)</i>: Es el método encargado de modificar la información de la Planta.</p> <p><i>Insertar()</i>: Es el método encargado de insertar una nueva Planta</p> <p><i>Cancelar()</i>: Es el método encargado de cancelar la acciones de envío del formulario.</p>

Tabla 11. Clase Frm_GestionarPlanta.

Módulo Administración. Clase CP_InsertarPlanta

Nombre de la clase:	CP_InsertarPlanta
Tipo:	Client Page
Descripción:	Es la clase vista encargada de mostrar el formulario Frm_InsertarPlanta donde el usuario registrará una nueva Planta.
Atributos	No aplica
Métodos	<p><i>Confirmacion()</i>: Esta función es la encargada de confirmar todas las acciones que realice el usuario en la interfaz.</p> <p>Validaciones de campos.</p> <p>Confirmaciones de acciones de botones u otros componentes que se encuentre dentro de la interfaz.</p>

Tabla 12. Clase CP_InsertarPlanta.

Capítulo 4 Construcción de la solución propuesta

Módulo Administración. Clase Frm_InsertarPlanta

Nombre de la clase:	Frm_InsertarPlanta
Tipo:	Form
Descripción:	Es el formulario encargado de permitir insertar una Planta en el Sistema.
Atributos	<p><i>NombreCorto</i>: Es el atributo que almacena el nombre corto de la Planta de tipo TextField.</p> <p><i>NombreLargo</i>: Es el atributo que almacena el nombre largo de la Planta de tipo TextField.</p> <p><i>TipoPlanta</i>: Es el atributo que contiene la relación de tipos de Planta existentes en el Sistema de tipo Select.</p> <p><i>RefineriaPertenece</i>: Es el atributo que contiene la relación de las Refinerías existentes en el Sistema de tipo Select.</p> <p><i>Aceptar</i>: Es el atributo de tipo botón que permite ejecutar la acción de ingresar una nueva Planta en el Sistema.</p> <p><i>Cancelar</i>: Es un atributo de tipo botón que permite ejecutar la acción cancelar el insertar una Planta.</p>
Métodos	<p><i>Enviar(nomCto, nomLgo, tipoPta, refPce)</i>: Es el método encargado de insertar la Planta en el Sistema.</p> <p><i>Cancelar()</i>: Es el método encargado de cancelar la acciones de envío del formulario.</p>

Tabla 13. Clase Frm_InsertarPlanta.

Módulo Administración. Clase CP_ModificarPlanta

Nombre de la clase:	CP_ModificarPlanta
Tipo:	Client Page
Descripción:	Es la clase vista encargada de mostrar el formulario Frm_ModificarPlanta donde el usuario modificará la información de la Planta en cuestión.
Atributos	No aplica
Métodos	<i>Confirmacion()</i> : Es el método encargado de confirmar todas las acciones que realice el usuario en la interfaz.

Tabla 14. Clase CP_ModificarPlanta.

Capítulo 4 Construcción de la solución propuesta

Módulo Administración. Clase Frm_ModificarPlanta

Nombre de la clase:	Frm_ModificarPlanta
Tipo:	Form
Descripción:	Es el formulario compuesto por los atributos: NombreCorto, NombreLargo, TipoPlanta, RefineriaPertenece, Aceptar, Cancelar.
Atributos	<p><i>NombreCorto</i>: Es el atributo que almacena el nombre corto de la Planta de tipo TextField.</p> <p><i>NombreLargo</i>: Es el atributo que almacena el nombre largo de la Planta de tipo TextField.</p> <p><i>TipoPlanta</i>: Es el atributo que contiene la relación de tipos de Planta existentes en el Sistema de tipo Select.</p> <p><i>RefineriaPertenece</i>: Es el atributo que contiene la relación de las Refinerías existentes en el Sistema de tipo Select.</p> <p><i>Aceptar</i>: Es el atributo de tipo botón que permite ejecutar la acción de ingresar una nueva Planta en el Sistema.</p> <p><i>Cancelar</i>: Es un atributo de tipo botón que permite ejecutar la acción cancelar el insertar una Planta.</p>
Métodos	<p><i>Enviar(idPta, nomCto, nomLgo, tipoPta, refPce)</i>: Es el método encargado de modificar la Planta en el Sistema.</p> <p><i>Cancelar()</i>: Es el método encargado de cancelar la acciones de envío del formulario.</p>

Tabla 15. Clase Frm_ModificarPlanta.

Módulo Administración. Clase SP_GestionarPlanta

Nombre de la clase:	SP_GestionarPlanta
Tipo:	Server Page
Descripción:	Es la clase servidora encargada de construir la página cliente CP_GestionarPlanta.
Atributos	No aplica
Métodos	<p><i>CrearGestionarPlanta(idPta, nomCto, nomLgo, tipoPta, refPce)</i>: Es el método encargado de crear la página cliente CP_GestionarPlanta con la funcionalidad de mostrar las plantas existentes en el Sistema.</p>

Tabla 16. Clase SP_GestionarPlanta.

Capítulo 4 Construcción de la solución propuesta

Módulo Administración. Clase SP_InsertarPlanta

Nombre de la clase:	SP_InsertarPlanta
Tipo:	Server Page
Descripción:	Es la clase servidora encargada de construir la página cliente CP_InsertarPlanta que contiene el formulario Frm_InsertarPlanta.
Atributos	No aplica
Métodos	<i>CrearInsertarPlanta()</i> : Es el método encargado de crear la página cliente CP_InsertarPlanta la cual contiene el formulario Frm_InsertarPlanta cuya funcionalidad es Insertar una nueva planta en el Sistema.

Tabla 17. Clase SP_InsertarPlanta.

Módulo Administración. Clase SP_ModificarPlanta

Nombre de la clase:	SP_ModificarPlanta
Tipo:	Server Page
Descripción:	Es la clase servidora encargada de recibir los datos de la clase controladora CC_GestionarPlantasscontroller y crear la página cliente CP_ModificarPlantas la cual contiene el formulario Frm_ModificarPlanta.
Atributos	No aplica
Métodos	<i>CrearModificarPlanta(idPta, nomCto, nomLgo, tipoPta, refPce)</i> : Es el método encargado de crear la página cliente CP_ModificarPlanta cuya funcionalidad es modificar las plantas registradas en el Sistema.

Tabla 18. Clase SP_ModificarPlanta.

Módulo Administración. Clase SP_EliminarPlanta

Nombre de la clase:	SP_EliminarPlanta
Tipo:	Server Page
Descripción:	Es la clase servidora encargada de invocar la clase SP_GestionarPlanta para refrescar los atributos de la misma, una vez que la clase CC_GestionarPlantasscontroller ejecuta el método <i>EliminarPlanta(idPta)</i> .
Atributos	No aplica

Capítulo 4 Construcción de la solución propuesta

Métodos	<i>ReadLoad()</i> : Es el método encargado de la redirección a la clase servidora SP_GestionarPlanta con la funcionalidad de mostrar las Plantas existentes en el Sistema.
----------------	--

Tabla 19. Clase SP_EliminarPlanta.

Capa Controladora.

Módulo Administración. Clase GestionarPlantascontroller

Nombre de la clase:	GestionarPlantascontroller
Tipo:	Controller
Descripción:	Es la clase controladora encargada de todas las acciones realizadas sobre las Plantas manejadas por el Sistema. Invoca a los métodos en la clase modelo CM_GestionarPlanta.
Atributos	No aplica
Métodos	<p><i>InsertarPlanta(nomCto, nomLgo, tipoPta, refPce)</i>: Es el método encargado de adicionar una nueva Planta al Sistema.</p> <p><i>ModificarPlanta(idPta, nomCto, nomLgo, tipoPta, refPce)</i>: Es el método encargado de modificar la información de una Planta en el Sistema.</p> <p><i>EliminarPlanta(idPta)</i>: Este método permite eliminar una Planta existente en el Sistema.</p> <p><i>GestionarPlanta()</i> Este método permite obtener la información de las Plantas existentes en el Sistema.</p> <p>Esta clase hereda todos los métodos de la clase ApplicationController del framework CAKE.</p>

Tabla 20. Clase GestionarPlantascontroller.

Capa Modelo.

Módulo Administración. Clase CM_GestionarPlanta

Nombre de la clase:	CM_GestionarPlanta
Tipo:	Model
Descripción:	Es la clase modelo encargada de la lógica del caso de uso Gestionar Planta, además del Acceso a los Datos permitiendo acceder a la información referente a las Plantas del Sistema.

Capítulo 4 Construcción de la solución propuesta

Atributos	No aplica
Métodos	Esta clase hereda todos los métodos de la clase AppModel del framework CAKE para acceso a los datos.

Tabla 21. Clase CM_GestionarPlanta.

4.3. Principios de diseño.

El diseño Web no se limita a la apariencia estética, a la combinación de colores, o a un logo más o menos acertado. De él depende que la información sea útil, que los servicios se puedan usar, es decir, el diseño convierte a una aplicación Web en algo atractivo para el usuario, por su estética y su utilidad. (30)

Construir y diseñar una aplicación web es una tarea que requiere de una serie de cuidados específicos, a nivel formal, organizativo, visual, de contenidos, y para ello hay que observar una serie de normas básicas enfocadas a la consecución exitosa del mismo. A continuación se abordan las pautas o criterios seguidos para el diseño del Módulo de Administración de SACGIR, con el objetivo de ofrecer entornos fáciles de utilizar, prácticos y, sobre todo, pensados para cumplir con las necesidades y expectativas de los usuarios. (31)

- Mostrar al usuario solamente aquellas opciones a las que, dado su rol en el negocio, tiene derecho a acceder.
- La interfaz ha de ser simple (aunque no simplista), fácil de aprender y usar, con funcionalidades accesibles y bien definidas.
- Mantener una estructura visual constante, donde los diferentes elementos mostrados en la pantalla no varíen en su ubicación entre páginas.
- Proveer de un feedback a los usuarios, de manera que éstos siempre conozcan y comprendan lo que sucede en todos los pasos del proceso, con el objetivo de mantenerlo informado del estado del sistema.
- Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse de que es correcta la opción seleccionada.

4.3.1. Estándares de la interfaz de la aplicación.

El diseño de interfaces de usuario es una tarea que ha adquirido relevancia en el desarrollo de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso (32). Con vistas a lograr un diseño consistente de la interfaz de la aplicación, se utilizaron plantillas y hojas de estilos para guardar la configuración del diseño para todas las páginas.

Capítulo 4 Construcción de la solución propuesta

La estructura de la plantilla utilizada es la siguiente: en la parte superior se encuentra el encabezado de la página donde están localizados, a la izquierda el logo de la compañía cliente y a la derecha el nombre del Sistema, además de vínculos a opciones de uso común para todos los usuarios, ej. Cambiar Rol, Notas y Avisos, Salir. En la parte central izquierda están colocados los menús de navegabilidad del usuario que varían en dependencia del rol que desempeñen y contienen las operaciones a realizar y el centro de la plantilla estará conformado por todos los formularios. A continuación se muestra la plantilla general de la aplicación:



Figura 8. SACGIR: Plantilla general de la aplicación.

Cuando se accede al sistema la primera interfaz que se muestra es la de autenticación de usuarios.

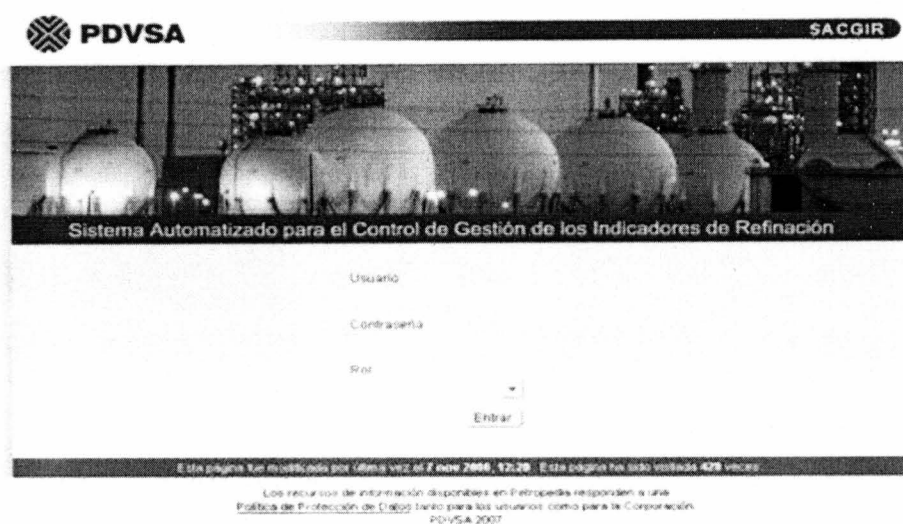


Figura 9. Diseño de Interfaz: Autenticar Usuario.

Capítulo 4 Construcción de la solución propuesta

La siguiente interfaz es la principal del módulo de administración.

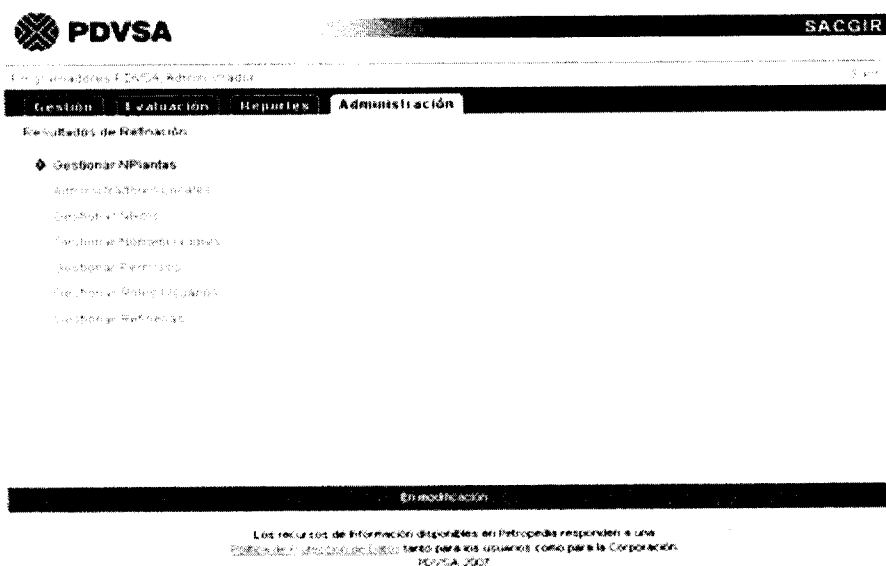


Figura 10. SACGIR: Interfaz Módulo Administración.

4.4. Diseño de la Base de Datos.

El sistema cuenta con 53 tablas que permiten la persistencia de datos para los módulos de forma común. Todas las tablas son arquitectónicamente significativas para el sistema. Como Gestor de Bases de Datos se utilizó PostgreSQL, ya que es Software Libre, con funcionalidades para el tratamiento y seguridad de grandes volúmenes de datos. La utilización del Framework de desarrollo CakePHP ofrece facilidades en el manejo de los datos, aunque restringe en la utilización de Vistas, y Triggers declarados en la Base de Datos.

En los siguientes subepígrafes se presentan el Diagrama de Entidad Relación y el Diagrama de Clases Persistentes de la aplicación.

4.4.1. Diagrama Entidad Relación.

Un diagrama de Entidad Relación constituye una herramienta para el modelado de datos de un sistema de información. Este está formado por un conjunto de conceptos (entidades, relaciones, atributos, entre otros) que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. (33)

En la figura se muestra el Diagrama Entidad Relación que sirvió de base para el posterior desarrollo de la base de datos del Sistema Automatizado para el Control de Indicadores de Refinación (SACGIR). En el mismo, están señaladas en rojo las entidades que tienen que ver directamente con el módulo de administración, en el Anexo 5 se podrán encontrar las descripciones de las mismas.

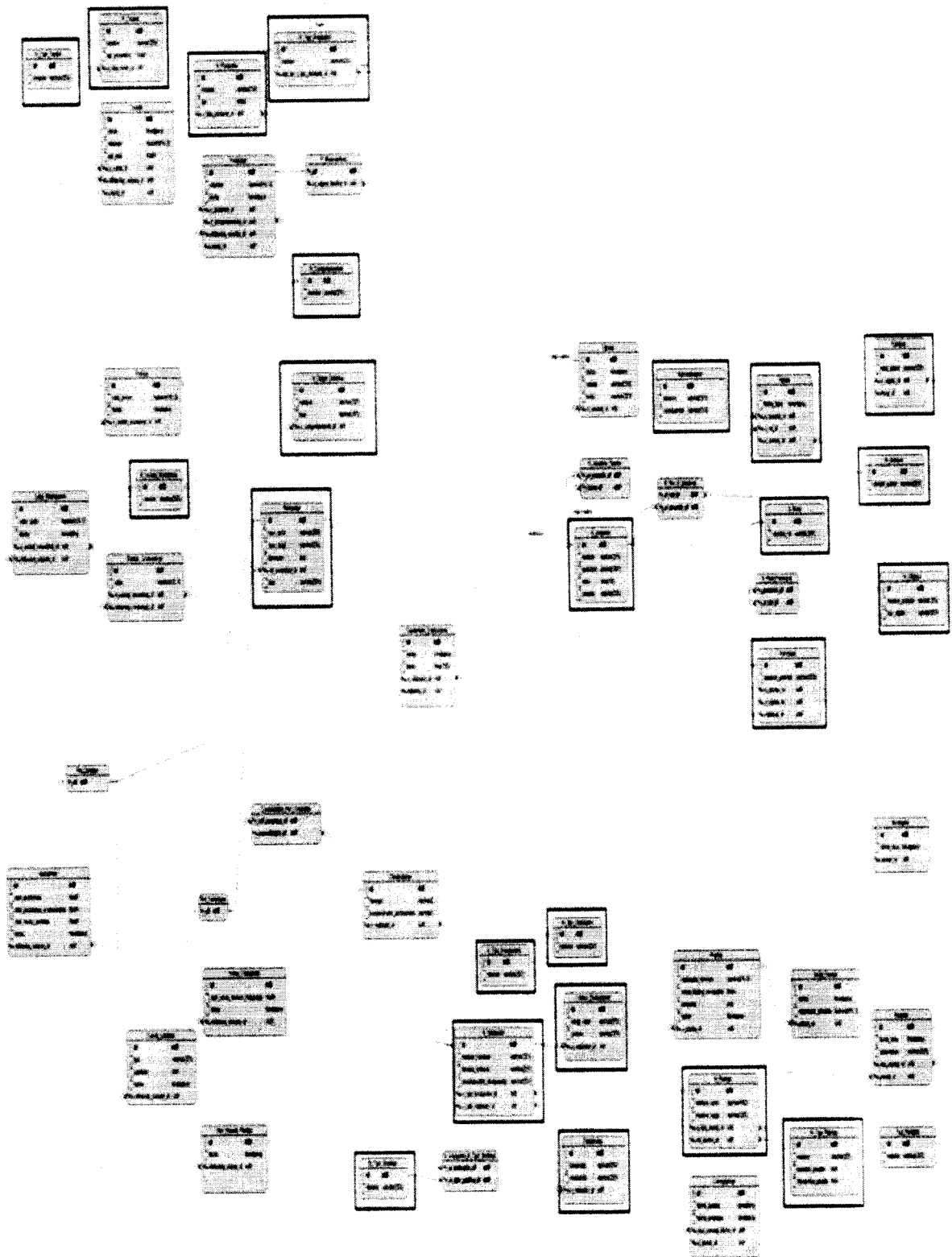


Figura 11. Diagrama Entidad Relación.

4.4.2. Diagrama de Clases Persistentes.

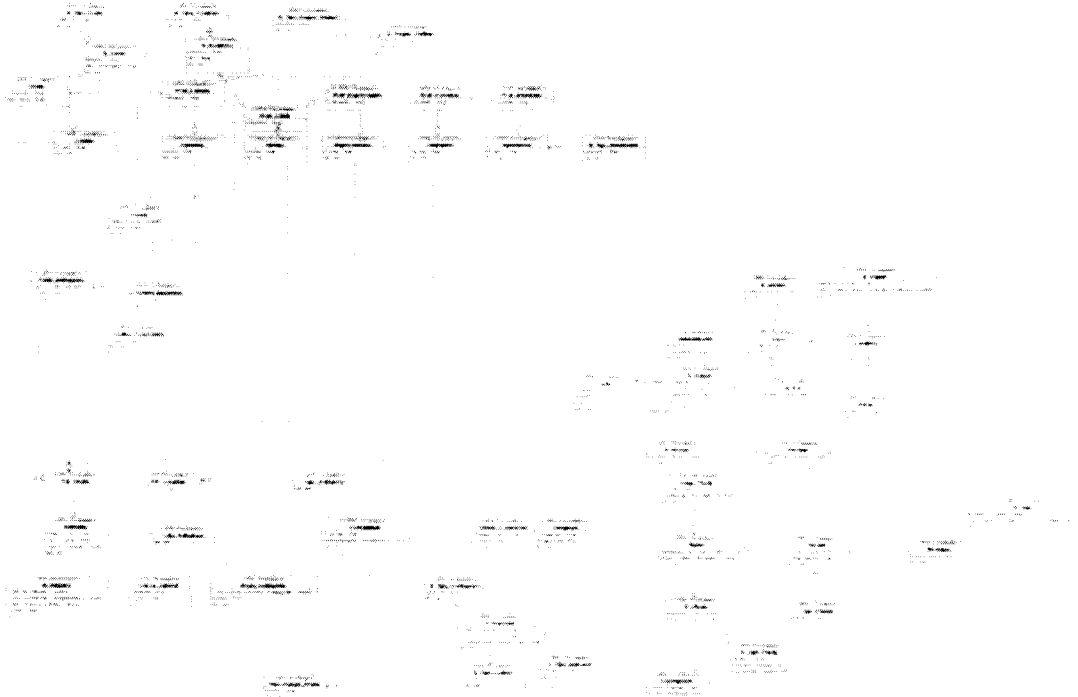


Figura 12. Diagrama de Clases Persistentes.

4.5. Diagrama de Despliegue.

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema. Es un conjunto de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos y procesos.

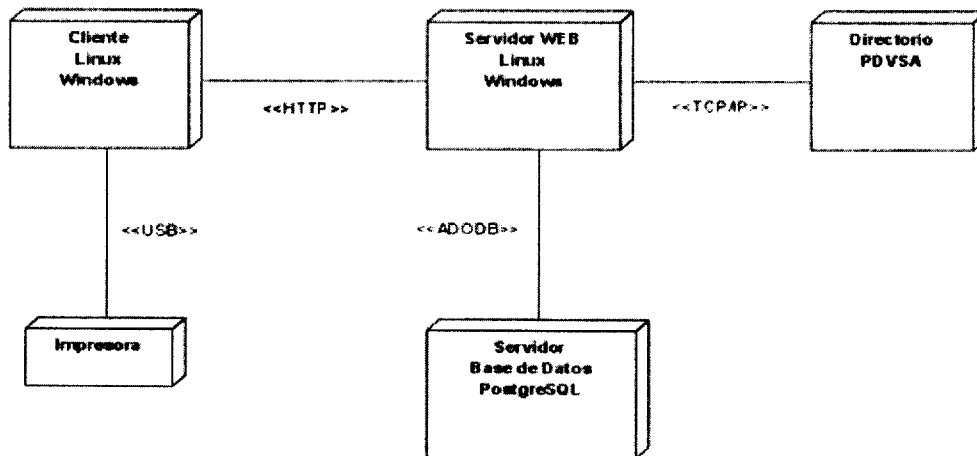


Figura 13. Diagrama de Despliegue.

Capítulo 4 Construcción de la solución propuesta

A continuación se describirá los nodos físicos representados en el diagrama de despliegue:

Servidor Web

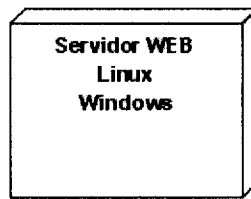


Figura 14. Servidor Web.

Subsistemas de Implementación.

En este nodo se ejecutarán todas las funcionalidades del servidor Web, entre ellas se encuentra la construcción de interfaces de usuarios, el procesamiento de datos, y el control de flujo.

- Capa Presentación
- Capa de Control
- Capa de Modelo

Servidor de Bases de Datos

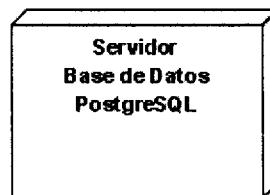


Figura 15. Servidor de Bases de Datos.

Subsistemas de Implementación.

En este nodo estará ejecutándose el servidor PostgreSQL. La lógica del tratamiento de los datos no se implementará aquí sino en el servidor Web en la misma aplicación.

- Capa de Datos

Ciente Web

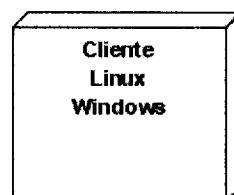


Figura 16. Clientes Web.

Capítulo 4 Construcción de la solución propuesta

Subsistemas de Implementación.

- Representación de la Capa Presentación en el cliente a través de Javascript.

Descripción de los elementos e interfaces de comunicación

En el caso de nuestro proyecto, el cliente ya tiene establecido la configuración de las redes de datos en la empresa. Por lo que el sistema solamente haría uso de las mismas para su ejecución.

Protocolos TCP/IP y HTTP

Se utilizarán los protocolos de comunicación mencionados debido a que el sistema se desarrolla sobre tecnología Web siguiendo los estándares para la misma.

Cliente/Servidor

Como servidor Web se utiliza Apache 2.x, por sus características de Software Libre y compatibilidad con el lenguaje de desarrollo utilizado, PHP.

El cliente podrá ser Netscape 3 (o superior), Internet Explorer 4.2 (o superior) y compatibles; ejecutándose tanto en Sistema Operativo Windows como Linux.

4.6. Conclusiones.

En el presente capítulo se presentaron los resultados de la etapa de Análisis y Diseño, se presentan los diagramas de clases e interacción, se explican además los estándares y principios que fueron tenidos en cuenta a la hora de diseñar la interfaz de la aplicación. Se muestran asimismo los diagramas que se desarrollan para apoyar el diseño de la Base de Datos, es decir el diagrama de Entidad Relación y Clases Persistentes. Todos estos artefactos conforman una propuesta de solución para el Módulo de Administración del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR).

CONCLUSIONES GENERALES

Las empresas, independientemente de su tamaño, enfrentan demandas respecto a rentabilidad, calidad, tecnología y desarrollo sostenible. Un sistema de gestión eficiente, diseñado a la medida de sus procesos comerciales, puede ayudar a enfrentar los desafíos del cambiante mercado global de hoy. Un sistema de gestión puede ayudar a centrar, organizar y sistematizar los procesos para la gestión y mejora.

El objetivo del presente trabajo de diploma ha estado encaminado a realizar el análisis y diseño del Módulo de Administración del Sistema Automatizado de Control de Gestión de Indicadores de Refinación (SACGIR) para la Gerencia de Planificación y Gestión de Refinación de PDVSA, que permita definir el control de acceso a la información y brinde facilidades para definir los indicadores, refinerías y nomencladores de manera dinámica.

A lo largo de todo el ciclo de desarrollo de la tesis se le ha dado solución a una serie de tareas trazadas para dar cumplimiento a los objetivos específicos que rigen el desarrollo de este trabajo, obteniendo los siguientes resultados:

- ✓ Se documentaron los procesos fundamentales: Los procesos de administración de la información y el control de acceso a la información.
- ✓ Se realizó el análisis y diseño del sistema propuesto, quedando completa la documentación de Ingeniería de Software del mismo.
- ✓ Se desarrolló un prototipo no funcional del Módulo de Administración, acorde con los estándares de diseño definidos, que le permite conocer al usuario la forma en que se presentará la información.

RECOMENDACIONES

Con el objetivo de apoyar el futuro de este trabajo investigativo, su aplicación o generalización futuras, se recomienda lo siguiente:

- ❖ Basados en el Análisis y Diseño desarrollados, realizar la Implementación del módulo de Administración.
- ❖ Realizar un estudio más profundo de los procesos de gestión de indicadores.

Trabajos Citados

1. **PDVSA.** Acerca de PDVSA Historia. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 2 de Diciembre de 2007.] <http://www.pdvs.com>.
2. **Mingarro, Lic. Angela Mérida y Hernández, Lic. Margarita Vila.** Validación de un sistema de indicadores para medir el desempeño en la empresa de materiales de construcción de Holguín. *Monografías.com*. [En línea] 1997. [Citado el: 15 de Enero de 2008.] <http://www.monografias.com/trabajos15/valoracion/valoracion.shtml#TEORICA>.
3. **PDVSA.** Acerca de PDVSA Procesos Refinación. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 5 de Diciembre de 2007.] <http://www.pdvs.com>.
4. **PDVSA.** Acerca de PDVSA. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 11 de Noviembre de 2007.] <http://www.pdvs.com/>.
5. **Jaramillo, Jesús Mauricio Beltrán.** *Indicadores de Gestión, herramientas para lograr la competitividad*. s.l. : 3R Editores.
6. **PDVSA.** Acerca de PDVSA, PDVSA en Cifras. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 22 de Diciembre de 2007.] <http://www.pdvs.com/>.
7. **Hinestroza, Mg. Jorge, y otros.** El sabotaje tecnológico de PDVSA y los derrames petroleros en el Lago de Maracaibo. *Mindfully.org*. [En línea] Junio de 2003. [Citado el: 11 de Diciembre de 2007.] <http://www.mindfully.org/Plastic/Polyvinylchloride/PDVSA-Pet>.
8. **Comisión Presidencial Compras del Estado.** *V Rueda de Negocios Mesa # 3 Informática-Estudios*. Caracas : s.n., 2004.
9. **Pdvs** avanza en migración tecnológica a software libre. *Software Libre Chile*. [En línea] Noviembre de 2002. [Citado el: 10 de Enero de 2008 de 2008.] <http://www.softwarelibre.cl/drupal//?q=node/336>.
10. **PDVSA.** *PDVSA en ruta al software libre*. Caracas : PDVSA, 2005.
11. **Aplicaciones Web.** *ITIESA: Código Abierto y Comunicación Global*. [En línea] ITIESA, S.C, 2004. [Citado el: 19 de junio de 2008.] <http://www.iteisa.com/aplicaciones-web/>.
12. **Programación Web.** *Arsys*. [En línea] Arsys Internet S.L, 2008. [Citado el: 2 de Marzo de 2008 .] <http://www.arsys.es/soporte/programacion/windows.htm>.
13. **Valido, Y. y Moreira, Y.** *SAIMM. Sistema de Apoyo Integral a la Misión Milagro*. Ciudad de la Habana : Instituto Superior Politécnico "José Antonio Echeverría", 2005.
14. **E. Rivera.** Definiciones. *Asesoría Informática*. [En línea] 2004. [Citado el: 27 de Febrero de 2008.] http://www.asesoriainformatica.com/definiciones_h.htm.
15. **Miguel Angel Alvarez.** Introducción a Javascript. *Desarrolloweb.com*. [En línea] [Citado el: 2008 de Marzo de 2008.] <http://www.desarrolloweb.com/articulos/490.php>.

Trabajos Citados

16. PHP. *Wikipedia, la enciclopedia libre*. [En línea] Wikipedia Foundation, Inc., 18 de junio de 2008. [Citado el: 19 de junio de 2008.] <http://es.wikipedia.org/wiki/.php>.
17. **Der Henst S Christian Van**. ¿Qué es el PHP? . *Maestro del Web*. [En línea] Maestros del Web, 23 de Mayo de 2001. [Citado el: 12 de Marzo de 2008.] <http://www.maestrosdelweb.com/editorial/phpintro>.
18. **Dondo, Agustín**. Por qué elegir PHP. *PHP en castellano*. [En línea] Programación en castellano, S.L, 1999-2008. [Citado el: 11 de Marzo de 2008.] <http://www.programacion.net/php/articulo/porquephp>.
19. **Luis Almirante Brown**. Lo de Miguel, Blog de desarrollo. *WordPress.com*. [En línea] 6 de Julio de 2007. [Citado el: 20 de marzo de 2008.] <http://lodemiguel.wordpress.com/2007/07/06/cakephp>.
20. Características fundamentales de Cake PHP. *Mujeres TIC*. [En línea] Wordpress, 23 de Septiembre de 2007. [Citado el: 20 de Marzo de 2008.] <http://www.mujerestic.com/caracteristicas-fundamentales-de-cake-php>.
21. Página Oficial de CakePHP. *CakePHP.org*. [En línea] [Citado el: 10 de Enero de 2008.] <http://www.cakephp.org>.
22. Servidores Web. *Wikipedia*. [En línea] 1 de Mayo de 2008. [Citado el: 5 de mayo de 2008.] http://es.wikipedia.org/wiki/Servidor_web.
23. **Félix, Alvaro del Castillo San**. El servidor de web Apache: Introducción práctica. *Barrapunto.com*. [En línea] 2000. [Citado el: 30 de Marzo de 2008.] <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/c20.html>.
24. **Carlos Luis Cuenca**. Arquitectura del Servidor Apache. *Desarrollo Web*. [En línea] [Citado el: 9 de Febrero de 2008.] <http://www.desarrolloweb.com/articulos/1112.php>.
25. ¿Qué son los sistemas gestores de Base de datos? *Garbage Collector*. [En línea] 1 de Noviembre de 2004. [Citado el: 22 de Febrero de 2008.]
26. Introducción a PostgreSQL - Instalación e inicialización. *El rincón de Linux*. [En línea] 8 de Junio de 2007. [Citado el: 19 de Junio de 2008.] <http://www.linux-es.org/node/536>.
27. **Germán Arroyo Moreno** . Modelado del Negocio. *Guía virtual del proceso de desarrollo de Software*. [En línea] [Citado el: 1 de Abril de 2008.] http://lsi.ugr.es/~arroyo/inndoc/doc/negocio/negocio_d.php.
28. **Sergio Sánchez Ríos**. Metodologías de Análisis y Diseño. [En línea] 1 de Marzo de 2008. [Citado el: 23 de Marzo de 2008.] http://www.uvmsf.d/~ssanchez/images/Metodologias/Unidad2_MAD.pdf.
29. **Vilas, Ana Fernandez**. Grupo de Redes e Ingeniería del Software. *GRIS*. [En línea] 20 de Marzo de 2001. [Citado el: 23 de Febrero de 2008.] <http://www.gris.det.uvigo.es/~avilas/UML/node31.html>.
30. **Legarda, David Maniega**. Aplicación de criterios de usabilidad en sitios web: consejos y pautas para una correcta interpretación . *E-LIS*. [En línea] 2003. [Citado el: 1 de Abril de 2008.] Aplicación de criterios de usabilidad en sitios web: consejos y pautas para una correcta interpretación .

Trabajos Citados

31. **Eduardo Manchón.** Principios generales de usabilidad en diseño de sitios web . *Alzado.org*. [En línea] 9 de Febrero de 2003. [Citado el: 10 de Marzo de 2008.] http://www.alzado.org/articulo.php?id_art=56.
32. **Leopoldo Sebastián M. Gómez.** Diseño de Interfaces de Usuario. Principios, Prototipos y Heurísticas para Evaluación. *Monografías.com*. [En línea] 2006. [Citado el: 23 de Abril de 2008.] <http://www.monografias.com/trabajos10/diusuar/diusuar.shtml>.
33. **Chavez, Antonia y Sancho, Fernando.** El modelo Entidad-Relación. *Bases de Datos*. [En línea] 2002. [Citado el: 25 de Abril de 2008.] <http://www.cs.us.es/cursos/bd-2001/temas/disen.html>.

Bibliografía

1. **PDVSA.** Acerca de PDVSA Historia. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 2 de Diciembre de 2007.] <http://www.pdvs.com>..
2. **Mingarro, Lic. Angela Mérida y Hernández, Lic. Margarita Vila.** Validación de un sistema de indicadores para medir el desempeño en la empresa de materiales de construcción de Holguín. *Monografias.com*. [En línea] 1997. [Citado el: 15 de Enero de 2008.] <http://www.monografias.com/trabajos15/valoracion/valoracion.shtml#TEORICA>..
3. **PDVSA.** Acerca de PDVSA Procesos Refinación. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 5 de Diciembre de 2007.] <http://www.pdvs.com>.
4. **PDVSA.** Acerca de PDVSA. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 11 de Noviembre de 2007.] <http://www.pdvs.com/>.
5. **Jaramillo, Jesús Mauricio Beltrán.** *Indicadores de Gestión, herramientas para lograr la competitividad*. s.l. : 3R Editores.
6. **PDVSA.** Acerca de PDVSA, PDVSA en Cifras. *Sitio Oficial de PDVSA*. [En línea] PDVSA, 2005. [Citado el: 22 de Diciembre de 2007.] <http://www.pdvs.com/>.
7. **Hinestroza, Mg. Jorge, y otros.** El sabotaje tecnológico de PDVSA y los derrames petroleros en el Lago de Maracaibo. *Mindfully.org*. [En línea] Junio de 2003. [Citado el: 11 de Diciembre de 2007.] <http://www.mindfully.org/Plastic/Polyvinylchloride/PDVSA-Pet>.
8. **Comisión Presidencial Compras del Estado.** *V Rueda de Negocios Mesa # 3 Informática-Estudios*. Caracas : s.n., 2004.
9. **Pdvs** avanza en migración tecnológica a software libre. *Software Libre Chile*. [En línea] Noviembre de 2002. [Citado el: 10 de Enero de 2008 de 2008.] <http://www.softwarelibre.cl/drupal//?q=node/336>.
10. **PDVSA.** *PDVSA en ruta al software libre*. Caracas : PDVSA, 2005.
11. **Aplicaciones Web.** *ITIESA: Código Abierto y Comunicación Global*. [En línea] ITIESA, S.C, 2004. [Citado el: 19 de junio de 2008.] <http://www.iteisa.com/aplicaciones-web/>.
12. **Programación Web.** *Arsys*. [En línea] Arsys Internet S.L, 2008. [Citado el: 2 de Marzo de 2008 .] <http://www.arsys.es/soporte/programacion/windows.htm>..
13. **Valido, Y. y Moreira, Y.** *SAIMM. Sistema de Apoyo Integral a la Misión Milagro*. Ciudad de la Habana : Instituto Superior Politécnico “José Antonio Echeverría”, 2005.
14. **E. Rivera.** Definiciones. *Asesoría Informática*. [En línea] 2004. [Citado el: 27 de Febrero de 2008.] http://www.asesoriainformatica.com/definiciones_h.htm.

Bibliografía

15. **Miguel Angel Alvarez**. Introducción a Javascript. *Desarrolloweb.com*. [En línea] [Citado el: 2008 de Marzo de 2008.] <http://www.desarrolloweb.com/articulos/490.php>.
16. PHP. *Wikipedia, la enciclopedia libre*. [En línea] Wikipedia Foundation, Inc., 18 de junio de 2008. [Citado el: 19 de junio de 2008.] <http://es.wikipedia.org/wiki/.php>.
17. **Der Henst S Christian Van**. ¿Qué es el PHP? . *Maestro del Web*. [En línea] Maestros del Web, 23 de Mayo de 2001. [Citado el: 12 de Marzo de 2008.] <http://www.maestrosdelweb.com/editorial/phpintro>.
18. **Dondo, Agustín**. Por qué elegir PHP. *PHP en castellano*. [En línea] Programación en castellano, S.L, 1999-2008. [Citado el: 11 de Marzo de 2008.] <http://www.programacion.net/php/articulo/porquephp>.
19. **Luis Almirante Brown**. Lo de Miguel, Blog de desarrollo. *WordPress.com*. [En línea] 6 de Julio de 2007. [Citado el: 20 de marzo de 2008.] <http://lodemiguel.wordpress.com/2007/07/06/cakephp>.
20. Características fundamentales de Cake PHP. *Mujeres TIC*. [En línea] Wordpress, 23 de Septiembre de 2007. [Citado el: 20 de Marzo de 2008.] <http://www.mujeestic.com/caracteristicas-fundamentales-de-cake-php>.
21. Página Oficial de CakePHP. *CakePHP.org*. [En línea] [Citado el: 10 de Enero de 2008.] <http://www.cakephp.org>.
22. Servidores Web. *Wikipedia*. [En línea] 1 de Mayo de 2008. [Citado el: 5 de mayo de 2008.] http://es.wikipedia.org/wiki/Servidor_web.
23. **Félix, Alvaro del Castillo San**. El servidor de web Apache: Introducción práctica. *Barrapunto.com*. [En línea] 2000. [Citado el: 30 de Marzo de 2008.] <http://acs.barrapunto.org/articulos/trunk/LinuxActual/Apache/html/c20.html>.
24. **Carlos Luis Cuenca**. Arquitectura del Servidor Apache. *Desarrollo Web*. [En línea] [Citado el: 9 de Febrero de 2008.] <http://www.desarrolloweb.com/articulos/1112.php>.
25. ¿Qué son los sistemas gestores de Base de datos? *Garbage Collector*. [En línea] 1 de Noviembre de 2004. [Citado el: 22 de Febrero de 2008.]
26. Introducción a PostgreSQL - Instalación e inicialización. *El rincón de Linux*. [En línea] 8 de Junio de 2007. [Citado el: 19 de Junio de 2008.] <http://www.linux-es.org/node/536>.
27. **Germán Arroyo Moreno** . Modelado del Negocio. *Guía virtual del proceso de desarrollo de Software*. [En línea] [Citado el: 1 de Abril de 2008.] http://lsi.ugr.es/~arroyo/inndoc/doc/negocio/negocio_d.php.
28. **Sergio Sánchez Rios**. Metodologías de Análisis y Diseño. [En línea] 1 de Marzo de 2008. [Citado el: 23 de Marzo de 2008.] http://www.uvmsf.d/~ssanchez/images/Metodologias/Unidad2_MAD.pdf.
29. **Vilas, Ana Fernandez**. Grupo de Redes e Ingeniería del Software. *GRIS*. [En línea] 20 de Marzo de 2001. [Citado el: 23 de Febrero de 2008.] <http://www.gris.det.uvigo.es/~avilas/UML/node31.html>.

Bibliografía

30. **Legarda, David Maniega.** Aplicación de criterios de usabilidad en sitios web: consejos y pautas para una correcta interpretación . *E-LIS*. [En línea] 2003. [Citado el: 1 de Abril de 2008.] Aplicación de criterios de usabilidad en sitios web: consejos y pautas para una correcta interpretación .
31. **Eduardo Manchón.** Principios generales de usabilidad en diseño de sitios web . *Alzado.org*. [En línea] 9 de Febrero de 2003. [Citado el: 10 de Marzo de 2008.] http://www.alzado.org/articulo.php?id_art=56.
32. **Leopoldo Sebastián M. Gómez.** Diseño de Interfaces de Usuario. Principios, Prototipos y Heurísticas para Evaluación. *Monografias.com*. [En línea] 2006. [Citado el: 23 de Abril de 2008.] <http://www.monografias.com/trabajos10/diusuar/diusuar.shtml>.
33. **Chavez, Antonia y Sancho, Fernando.** El modelo Entidad-Relación. *Bases de Datos*. [En línea] 2002. [Citado el: 25 de Abril de 2008.] <http://www.cs.us.es/cursos/bd-2001/temas/disenio.html>.
34. **Dr. Rodolfo Porro;** Caso INTESA - SAIC. *Aporrea.org*. [En línea] 21 de Julio de 2004. [Citado el: 24 de Febrero de 2008.] <http://www.aporrea.org/actualidad/a8991.html>.
35. **MÉNDEZ, DUQUE y D, Néstor.** Conceptos de arquitectura Cliente/Servidor . *Sistemas de Información*. [En línea] [Citado el: 5 de Febrero de 2008.] http://www.it.uc3m.es/mcftp/docencia/si/material/1_cliente_mcfp.pdf.
36. *Sitio Oficial de PDVSA*. [En línea] [Citado el: 2 de Diciembre de 2007.] <http://www.pdvsa.com>.
37. **Stanley.** Principles, Architectures and Techniques. *Database Computers*. Nueva York : McGraw-Hill, 1988.
38. **Jesus Tramullas y Ed. Kronos.** Los sistemas de gestión de bases de datos. *Documática*. [En línea] 11 de Enero de 2004. [Citado el: 10 de Enero de 2008.] <http://tramullas.com/documatica/2-4.html>.
39. **Booch, Grady, Rumbaugh, James y Jacobson, Ivar.** *El lenguaje unificado de modelado. pág. 464*. Madrid : Pearson Addison-Wesley, 1999.
40. **Molpeceres, Alberto.** Procesos de desarrollo: RUP, XP y FDD. *Javahispano*. [En línea] 15 de Diciembre de 2002. [Citado el: 30 de Enero de 2008.] <http://javahispano.org/licencias>.
41. Modelado de Sistemas con UML. [En línea] 11 de Abril de 2007. [Citado el: 21 de Abril de 2008.] <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>.
42. **Larman, Craig.** *UML y Patrones. Introducción al Análisis y Diseño Orientado a Objeto*.

ANEXOS

Adjunto a este documento de tesis se entregará otro documento en el cual se puede encontrar todos los Anexos que se necesitan para una mejor comprensión del trabajo. El mismo está distribuido en 5 epígrafes los cuales son:

- Anexo 1. Diagramas de Actividades del Negocio.
- Anexo 2. Descripción de los CUS Secundarios.
- Anexo 3. Diagramas de Colaboración y Clases del Análisis.
- Anexo 4. Diagramas de Clases del Diseño.
- Anexo 5. Descripción de las entidades del DER (Diagrama Entidad Relación) del Módulo de Administración de SACGIR.

GLOSARIO DE TÉRMINOS

Apache: Servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.).

Aplicación: Un sistema comercial de computadora que procesa una parte de una base de datos para cumplir con las necesidades de información de un usuario. Se compone de menús, formas, reportes y programas de aplicaciones.

Base de datos: Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar.

Clase(s): Abstracciones de objetos. Una clase es una definición de un objeto. Un objeto es una instancia de una clase.

Cliente/Servidor: Método de distribución de información o de archivos en el cual la agrupación central, servidor, almacena los archivos y los hace disponibles para solicitudes de aplicaciones cliente.

Consultas: Una consulta SQL es tipo de consulta a una base de datos empleando lenguaje SQL.

CSS (*Cascading Style Sheets*): Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una Web.

Diagrama entidad-relación: Una representación gráfica de las entidades y sus relaciones.

Por lo general, las entidades se muestran en cuadrados o rectángulos y las relaciones en diamantes. La cardinalidad de la relación se muestra dentro del diamante.

Entidad: Algo de importancia para un usuario que necesita representarse en una base de datos. En un modelo entidad-relación, las entidades están limitadas a cosas que pueden representarse por medio de una tabla única.

Extreme Programming: Metodología heterodoxa de programación. Es la más popular de las denominadas metodologías ágiles.

Feature Driven Development (FDD): Metodología ágil de desarrollo. No requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso.

Frameworks: Plataforma, entorno, marco de trabajo.

Herramientas CASE (Computer Aided Software Engineering): Se incluyen una serie de herramientas, lenguajes y técnicas de programación que permiten la generación de aplicaciones de manera semiautomática.

Indicador: Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos, en la ejecución de un proyecto, programa o actividad. Resultado cuantitativo de comparar dos variables.

Glosario de Términos

Medida sustitutiva de información que permite calificar un concepto abstracto. Se mide en porcentajes, tasas y razones para permitir comparaciones.

Interfaz: En software, parte de un programa que permite el flujo de información entre un usuario y la aplicación.

JavaScript: Lenguaje de programación interpretado, o sea, no requiere compilación. Es utilizado especialmente en páginas web.

Microsoft Solutions Framework (MSF): Marco para desarrollo de sistemas de software basado en principios, modelos, disciplinas, conceptos, prácticas y recomendaciones propias, derivadas de la experiencia de Microsoft.

Nomenclador: Catálogo que tiene la nomenclatura de una ciencia.

Nomenclatura: Conjunto de principios y reglas que se aplican para la denominación inequívoca, única y distintiva.

Open Source: Código abierto o código libre.

RAM (Random Access Memory): Un tipo de memoria de ordenador a la que se puede acceder aleatoriamente.

Servlet: Los servlets son programas que funcionan como los CGI's convencionales atendiendo peticiones de un cliente teniendo al servidor como el encargado.

Subclases: Clase hijo de una clase padre. Algunos lenguajes de programación permiten que una clase tenga múltiples padres.

Tabla: Estructura que permite almacenar una entidad.

TCP/IP: Son las siglas de Protocolo de Control de Transmisión/Protocolo de Internet (en inglés Transmission Control Protocol/Internet Protocol).

Triggers (Trigger o disparador): Se define así a una subrutina que es ejecutada de manera automática cuando se produce algún tipo de transacción (inserción, borrado o actualización) en la tabla de una base de datos.

XAMPP: Es un paquete que combina el servidor Apache, base de datos y el lenguaje de programación PHP.

XML (eXtensible Markup Language): Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).