

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



FACULTAD 9

TÍTULO

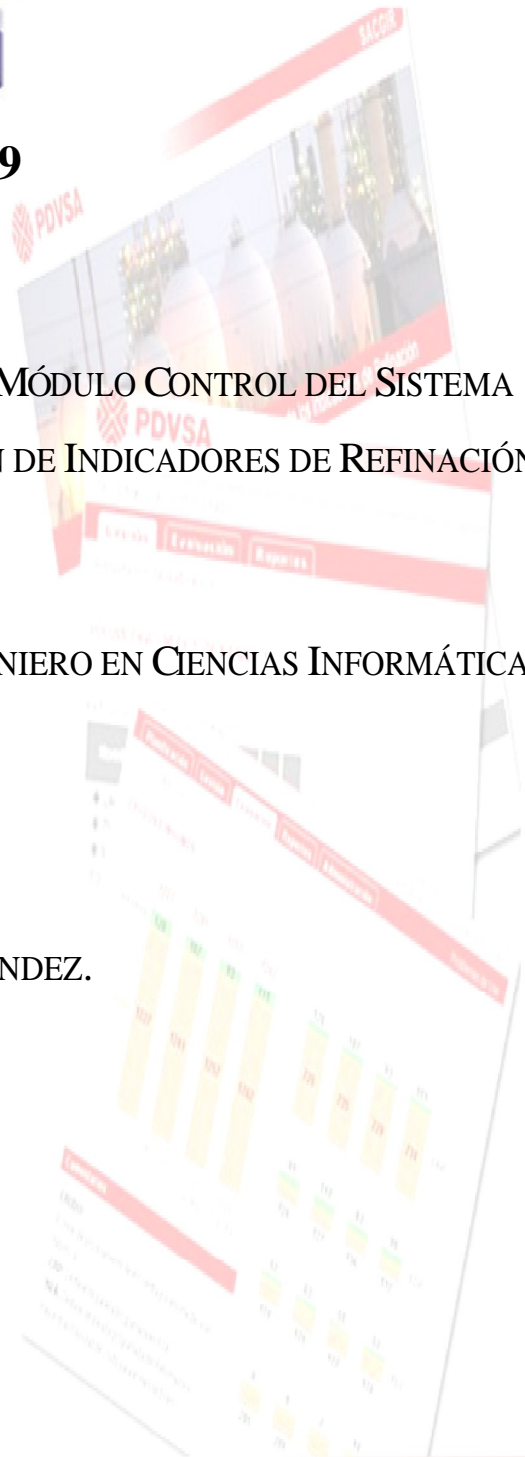
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO CONTROL DEL SISTEMA AUTOMATIZADO PARA EL CONTROL Y GESTIÓN DE INDICADORES DE REFINACIÓN.

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

AUTOR(ES): MAIDELINES ENNI MARTÍNEZ.

TUTOR: ING. YOENIS PANTOJA ZALDÍVAR.

CO-TUTOR: LIC. GRETCHEN GUILLERMO HERNÁNDEZ.



RESUMEN

Este trabajo consiste en el desarrollo del Módulo Control del Sistema Automatizado para el Control y la Gestión de Indicadores de Refinación (SACGIR). El cual se está realizando gracias al Convenio ALBET, S.A-PDVSA, S.A, debido a las deficiencias que tienen los sistemas informáticos en PDVSA Refinación y la necesidad de migración hacia el software libre.

En el documento se expone los resultados de todo el trabajo investigativo realizado. Se identifican y describen los procesos que desarrolla la empresa venezolana PDVSA Refinación, específicamente aquellos que corresponden al Módulo Control, que es el encargado de realizar la evaluación de la información referente a la empresa para conocer los resultados de la misma en un período determinado. Además se dan argumentos para demostrar que la situación problemática requiere de un sistema que cumple con los requisitos propuestos para satisfacer las necesidades identificadas. Para su desarrollo se siguieron los pasos que proponen James Rumbaugh, Ivar Jacobson y Grady Booch autores del Proceso Unificado de Desarrollo de Software. Se muestran los resultados del diseño de la propuesta del sistema, se desarrolla bajo el concepto de software libre, en PHP como lenguaje de programación apoyándose en PostgreSQL como gestor de Base de Datos.

El documento está estructurado en cinco capítulos que abordan un estudio de los principales conceptos asociados al dominio del problema, las tecnologías, herramientas existentes y las seleccionadas para el desarrollo de la aplicación, descripción de la propuesta de solución explicada en detalles y la construcción de dicha propuesta.

PALABRAS CLAVES

- Sistema.
- Indicadores.
- Evaluación de Indicadores.

TABLAS

TABLA # 1 DESCRIPCIÓN DEL CASO DE USO CONSOLIDAR RESULTADO MENSUAL DE REFINACIÓN.....	48
TABLA # 2 DESCRIPCIÓN DEL CASO DE USO EMITIR RESULTADO MENSUAL DE REFINACIÓN CONSOLIDADO	50
TABLA # 3 DESCRIPCIÓN DEL CASOS DE USO DEL NEGOCIO EMITIR RESULTADOS DE UN PERÍODO	51
TABLA # 4 DESCRIPCIÓN DEL CASO DE USO EVALUAR LA VARIACIÓN DE INDICADORES	52
TABLA# 5 REQUERIMIENTOS FUNCIONALES	54
TABLA # 6 REQUERIMIENTOS DE USABILIDAD	58
TABLA # 7 REQUERIMIENTOS DE CONFIABILIDAD.	59
TABLA # 8 REQUERIMIENTOS DE RENDIMIENTO	59
TABLA# 9 REQUERIMIENTOS DE SOPORTE.....	60
TABLA # 10 REQUERIMIENTOS DE RESTRICCIONES DEL DISEÑO	61
TABLA # 11 REQUERIMIENTOS DE ADQUISICIÓN DE COMPONENTES	61
TABLA # 12 REQUERIMIENTOS DE INTERFACES DE USUARIOS	61
TABLA # 13 REQUERIMIENTOS DE INTERFACES CON OTROS SOFTWARE.....	62
TABLA# 14 REQUERIMIENTOS DE HARDWARE. SERVIDOR WEB.....	62
TABLA# 15 REQUERIMIENTOS DE HARDWARE. SERVIDOR DE BASE DE DATOS.....	62
TABLA# 16 REQUERIMIENTOS DE HARDWARE. CLIENTE WEB	63
TABLA # 17 REQUERIMIENTOS DE SOFTWARE. PC CLIENTE	63
TABLA # 18 REQUERIMIENTOS DE SOFTWARE. PC SERVIDOR WEB	63
TABLA# 19 REQUERIMIENTOS DE SOFTWARE. SERVIDOR DE BASE DE DATOS.....	64
TABLA# 20 REQUERIMIENTOS DE SEGURIDAD	64
TABLA # 21 ACTORES DEL SISTEMA	65
TABLA # 22 DESCRIPCIÓN DEL CUS EVALUAR COMPORTAMIENTO DEL INDICADOR.....	65
TABLA # 23 DESCRIPCIÓN DEL CUS MOSTRAR INFORMACIÓN FUENTE DEL INDICADOR	71
TABLA # 24 DESCRIPCIÓN DEL CUS REALIZAR GRÁFICOS DE INFORMACIÓN	74
TABLA # 27 ESTÁNDAR DE CODIFICACIÓN PARA TIPOS DE VARIABLES.	86

FIGURAS

FIGURA # 1 REPRESENTACIÓN DE SISTEMA.....	10
FIGURA # 2 RUP EN DOS DIMENSIONES	22
FIGURA # 3 VISTA DEL MODELO DE ARQUITECTURA.	24
FIGURA # 4 PROCESO ITERATIVO E INCREMENTAL	25
FIGURA # 5 METODOLOGÍA EXTREME PROGRAMMING	26
FIGURA # 6 VISTA GENERAL DE FDD	29
FIGURA # 8 DIAGRAMA DE CASOS DE USOS DEL NEGOCIO DEL PROCESO GESTIÓN DE RESULTADOS MENSUALES Y EVALUACIÓN DE INDICADORES	48
FIGURA # 9 DIAGRAMA DE CASOS DE USO DEL SISTEMA. MÓDULO EVALUACIÓN.....	65
FIGURA # 11 DIAGRAMA DE CLASES DEL ANÁLISIS	78
FIGURA # 13 DIAGRAMA DE CLASES DEL DISEÑO DEL CUS EVALUACIÓN DEL COMPORTAMIENTO DE INDICADORES .MÓDULO EVALUACIÓN.....	81
FIGURA # 15 PÁGINA PRINCIPAL DE LA APLICACIÓN.....	82
FIGURA # 16 PÁGINA DE LA APLICACIÓN.	83
FIGURA # 17 PÁGINA EVALUACIÓN DE INDICADORES	83
FIGURA # 18 MODELO ENTIDAD RELACIÓN. MÓDULO INDICADORES.....	85
FIGURA # 19 DIAGRAMA DE CLASES PERSISTENTES. MÓDULO INDICADORES	86
FIGURA # 20 DIAGRAMA DE DESPLIEGUE	91
FIGURA 20.1 SERVIDOR WEB.	91
FIGURA 20.2 SERVIDOR DE BASES DE DATOS	92
FIGURA 20.3 SERVIDOR DE SERVICIOS WEB.....	92
FIGURA 20.4 CLIENTES WEB.	93
FIGURA 20.5 CLIENTE DE SERVICIOS WEB.....	93
FIGURA # 21 DIAGRAMA DE COMPONENTES.....	94
FIGURA # 22 DIAGRAMA DE ACTIVIDADES CONSOLIDAR RESULTADO MENSUAL DE REFINACIÓN.....	102
FIGURA # 23 DIAGRAMA DE ACTIVIDADES EMITIR RESULTADO MENSUAL DE REFINACIÓN CONSOLIDADO	103
FIGURA # 24 DIAGRAMA DE ACTIVIDADES EMITIR RESULTADOS DE REFINACIÓN DE UN PERÍODO	104
FIGURA # 25 DIAGRAMA DE ACTIVIDADES EVALUAR LA VARIACIÓN DE INDICADORES	105
FIGURA # 26 DIAGRAMA DE ACTIVIDADES EVALUAR LA VARIACIÓN DE INDICADORES	106
FIGURA # 27 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CUS MOSTRAR INFORMACIÓN FUENTE DEL INDICADOR.....	106
FIGURA # 28 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CASO DE USO GRAFICAR INFORMACIÓN.....	106
FIGURA # 30 DIAGRAMA DE COLABORACIÓN DEL CUS GRAFICAR INFORMACIÓN DEL INDICADOR.	107
FIGURA # 31 DIAGRAMA DE COLABORACIÓN DEL CUS EVALUAR COMPORTAMIENTO DEL INDICADOR. SECCIÓN MODIFICAR COMENTARIOS....	108
FIGURA # 32 DIAGRAMA DE COLABORACIÓN DEL CUS EVALUAR COMPORTAMIENTO DEL INDICADOR. SECCIÓN AÑADIR COMENTARIOS.	108
FIGURA # 33 DIAGRAMA DE COLABORACIÓN DEL CUS MOSTRAR INFORMACIÓN FUENTE.	108
FIGURA # 34 DIAGRAMA DE CLASES DEL DISEÑO MOSTRAR INFORMACIÓN FUENTE.	108

ÍNDICE

INTRODUCCIÓN	4
CAPÍTULO 1	9
CAPÍTULO # 1" FUNDAMENTACIÓN TEÓRICA"	9
1.1 <i>Introducción</i>	9
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA	9
1.2.1 <i>Sistema</i>	9
1.2.2 <i>Evaluación</i>	10
1.2.3 <i>Indicadores</i>	11
1.2.4 <i>Evaluación de Indicadores</i>	12
1.3 DESCRIPCIÓN GENERAL Y ACTUAL DEL PROCESO EVALUACIÓN DE INDICADORES	13
1.3.1 <i>Proceso de Evaluación de Indicadores</i>	13
1.3.2 <i>Situación Problemática</i>	14
1.4 DESCRIPCIÓN DEL SISTEMA ACTUAL EXISTENTE EN PDVSA	14
1.4.1 <i>Objetivos</i>	14
1.4.2 <i>Beneficios</i>	15
1.4.3 <i>Propiedades</i>	15
1.4.4 <i>Principales Problemas</i>	16
1.5 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES	16
1.5.1 <i>Petróleos Mexicanos (PEMEX)</i>	16
1.5.2 <i>Petróleos Colombianos (Ecopetrol)</i>	17
1.5.3 <i>Petróleos Brasileños (Petrobras)</i>	17
1.6 CONCLUSIONES PARCIALES	18
CAPÍTULO 2	19
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES	19
2.1 INTRODUCCIÓN	19
2.2 LENGUAJE UNIFICADO DE MODELADO (UML)	19
2.2.1 <i>Propiedades de UML</i>	20
2.2.2 <i>Ventajas de UML</i>	20
2.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	21
2.3.1 <i>Proceso Unificado de Desarrollo de Software (RUP)</i>	21
2.3.2 <i>Programación Extrema (Extreme Programming (XP))</i>	26
2.3.3 <i>Desarrollo Guiado por la Funcionalidad (FDD)</i>	28
2.3.4 <i>Selección de la Metodología a usar</i>	30
2.4 HERRAMIENTAS CASE DE DESARROLLO DE SOFTWARE	30
2.4.1 <i>Visual Paradigm</i>	31
2.4.2 <i>Rational Rose Enterprise</i>	31
2.4.3 <i>Selección de la herramienta a usar</i>	32
2.5 APLICACIÓN WEB	32
2.6 <i>Arquitectura Cliente Servidor</i>	33
2.6.1 <i>Características de la Arquitectura Cliente/Servidor</i>	33
2.6.2 <i>Ventajas de la Arquitectura Cliente Servidor</i>	34
2.6.3 <i>Características del Patrón Modelo Vista Controlador</i>	35
2.7 LENGUAJES DE PROGRAMACIÓN	36
2.7.1 HTML	36

2.7.1.1 Ventajas	37
2.7.2 JavaScript.....	37
2.7.3 PHP.....	38
2.7.3.1 Ventajas.....	39
2.7.3.2 Seguridad.....	39
2.7.3.3 Cake PHP.....	39
2.7.3.3.1 Características y Ventajas de CakePHP	40
2.8 SERVIDORES WEB	40
2.8.1 Apache.....	41
2.8.1.1 Características	41
2.8.1.2 Ventajas	41
2.9 SISTEMA GESTOR DE BASE DE DATOS.....	42
2.9.1 Sistema Gestor de Base PostgreSQL.....	42
2.9.2.1 Características	42
2.10 CONCLUSIONES PARCIALES.....	43
CAPÍTULO 3.....	45
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	45
3.1 INTRODUCCIÓN.....	45
3.2 MODELO DE NEGOCIO	45
3.2.1 Actores del Negocio.....	46
3.2.2 Trabajadores del Negocio.....	47
3.2.3 Diagrama de Casos de Uso del Negocio	48
3.2.4 Descripción textual de los casos de uso del Negocio	48
3.2.4.5 Proceso Gestión de Resultados Mensuales y Evaluación de Indicadores.....	48
3.3 REQUERIMIENTOS FUNCIONALES.....	54
3.4 REQUERIMIENTOS NO FUNCIONALES.....	58
3.4.1 Requerimientos de Usabilidad.....	58
3.4.2 Requerimientos de Confiabilidad.....	59
3.4.3 Requerimientos Rendimiento	59
3.4.4 Requerimientos de Soporte	60
3.4.5 Requerimientos de Restricciones de diseño.....	61
3.4.6 Requerimientos de Adquisición de Componentes.....	61
3.4.7 Requerimientos de Interfaz	61
3.4.7.1 Requerimientos de Interfaces de Usuarios.....	61
3.4.7.2 Requerimientos de Interfaces con otros Software	62
3.4.8 Requerimientos de Hardware.....	62
3.4.8.1 Servidor Web.....	62
3.4.8.2 Servidor de Base de Datos.....	62
3.4.8.3 Cliente Web	63
3.4.9 Requerimientos de Software	63
3.4.9.1 PC Cliente	63
3.4.9.2 PC Servidor Web.....	63
3.4.9.3 Servidor de Bases de Datos	64
3.4.10 Requerimientos de Seguridad.....	64
3.5 DESCRIPCIÓN DEL SISTEMA PROPUESTO	65
3.5.1 Descripción de los actores sistema	65
3.5.2 Diagrama de Caso de Uso del Sistema (CUS)	65
3.5.2.1 Descripción del CUS Evaluar Comportamiento del Indicador.....	65
3.5.2.2 Descripción del CUS Mostrar Información fuente del indicador	71
3.5.2.3 Descripción del CUS Realizar Gráficos de Información	74

3.6 CONCLUSIONES PARCIALES.....	75
CAPÍTULO 4.....	76
CAPÍTULO 4: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	76
4.1 INTRODUCCIÓN.....	76
4.2 MODELO DE ANÁLISIS.....	76
4.2.1 <i>Diagrama de Clases del Análisis</i>	76
4.2.1.1 Diagrama de Clases del Análisis del CUS Evaluación del Comportamiento de Indicadores.....	78
4.2.2 <i>Diagramas de Interacción</i>	78
4.3 MODELO DE DISEÑO.....	79
4.3.1 <i>Diagrama de Clases del Diseño</i>	79
4.3.1.1 Diagrama de Clases del Diseño del CUS Evaluación del Comportamiento de Indicadores .Módulo Evaluación.....	81
4.3.2 <i>Principios del Diseño</i>	81
4.3.2.1 Estándares en la interfaz de la aplicación.....	82
4.4.1 <i>Diseño de la Base de Datos</i>	84
4.4.1.1 Modelo Entidad Relación.....	84
4.4.1.2 Diagrama de Clases Persistentes.....	85
4.5 ESTÁNDAR EN LA CODIFICACIÓN.....	86
4.6 MODELO DE DESPLIEGUE.....	90
4.6.1 <i>Diagrama de despliegue</i>	91
4.6.1.1 Descripción de los Nodos.....	91
4.7 MODELO DE IMPLEMENTACIÓN.....	93
4.7.1 <i>Diagrama de Componentes</i>	94
4.8 CONCLUSIONES PARCIALES.....	94
CONCLUSIONES GENERALES.....	96
RECOMENDACIONES.....	97
REFERENCIAS BIBLIOGRÁFICAS.....	98
BIBLIOGRAFÍA.....	100
ANEXOS.....	102
GLOSARIO.....	109

INTRODUCCIÓN

Petróleos de Venezuela, S.A (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, dicho proceso se encarga de la transformación de los hidrocarburos en productos derivados.

PDVSA esta estructurado de la siguiente manera en el área de Refinación:

A nivel corporativo existe la Gerencia de Planificación y Gestión de Refinación que tiene subordinada a ella las Gerencias de Evaluación de Gestión de Refinación, Planificación, Nuevos Negocios y Finanzas.

A nivel de refinería existe la Gerencia Técnica que tiene subordinado la Gerencia de Programación y Gestión (en algunas refinerías se le llama Gerencia de Programación y Economía).

En las gerencias antes mencionadas se manejan un gran número de datos y de información, de importancia para la empresa, algunos de estos datos son lo indicadores de refinación, que estos no son más que valores en los que los directivos se basan para realizar la evaluación de las refinerías en un período determinado.

En la historia de PDVSA, desde 1984, el manejo de la información se ha llevado a cabo de forma manual. En 1996 los directivos de PDVSA y la empresa estadounidense SAIC¹, fundaron la sociedad Intesa, con el objetivo de actualizar y gestionar en el futuro el sistema de computación de la empresa. En este "negocio", justificado supuestamente por la reducción de costos, PDVSA aportaba la totalidad del capital, pero controlaba sólo el 40% de las acciones de Intesa. Sin embargo SAIC, sobre la base solamente de sus conocimientos técnicos, se quedaba con el 60% de las acciones.

A través de Intesa, SAIC terminó controlando desde sus servidores todos los datos financieros, presupuestarios, de instalaciones físicas operativas y de negocios de PDVSA. De esa manera, información que se calificaba como confidencial, y que era de importancia fundamental para la seguridad y defensa del país, quedó en manos de una empresa transnacional.

En Diciembre del 2002, comenzó el paro-sabotaje en contra de la industria petrolera venezolana. La directiva de la empresa utilizó el Control de los sistemas informáticos de PDVSA para sabotear su funcionamiento. Intesa prácticamente se unió al paro puesto que bajo la dirección de SAIC alentó a sus empleados a abandonar sus puestos de trabajo, lo cual interrumpió el funcionamiento de los sistemas informáticos de PDVSA y ocasionó la pérdida de capacidad operacional de la corporación. Cuando los trabajadores de Intesa abandonaron sus puestos de trabajo, PDVSA se vio obligada a

¹ Science Applications International Corporation. Es una empresa que ha montado los sistemas de defensa y ataque aeroespacial mundial del Departamento de Defensa de Estados Unidos

tratar de restablecer sus operaciones de informática con un equipo de voluntarios. Además PDVSA suspendió el acceso remoto a los sistemas. Esta fue una medida de seguridad indispensable por la magnitud de la crisis y de los riesgos inherentes a ella, pues mientras PDVSA trataba de restaurar sus operaciones, manualmente, ocurrían interrupciones vía acceso remoto.

La revolución del software libre esta encontrando un eco enorme en las empresas de muy diversas industrias, por las facilidades que este ofrece que una vez obtenido puede ser usado, estudiado, copiado, modificado y redistribuido libremente, además que los costos son mil veces mas bajos que el software propietario, por lo que el Ministro del Poder Popular para las Telecomunicaciones e Informática (MPPTI), Jesse Chacón, expresó que el Estado venezolano no invertirá más en la compra de software propietario, es así como el 13 de julio del 2005 se aprueba en PDVSA el plan de migración a software libre, teniendo en cuenta además los acontecimientos referidos al paro petrolero del 2002 y la posición que tomó la sociedad Intesa en estos sucesos, al mismo tiempo ya existían dificultades con los servicios informáticos existente en PDVSA.

El Sistema de Refinación de PDVSA cuenta con una variedad de sistemas informáticos que apoyan el proceso, y a su vez alimentan la gestión de datos para los entes interesados. Entre esos sistemas se encuentra el Sistema de Información de Manufactura (SIM), orientado a la gestión de datos de Refinación, desarrollado en plataforma Web con el lenguaje propietario ASP y con soporte de datos sobre Oracle. Desde el paro petrolero en muchas refinerías no se ha podido reactivar el sistema, debido a la falta de documentación que existe, dando espacio al procesamiento de datos de forma manual y a los inconvenientes que esto conlleva. Además la herramienta tiene aspectos, como la eliminación de transferencias de productos entre refinerías para la Consolidación de los Balances², que limitan su empleo y obligan al personal a realizar operaciones secundarias con sistemas en Excel, aumentando el margen de error.

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. Por lo que el **Problema** es: ¿Cómo perfeccionar un Sistema de Gestión de evaluación de los indicadores de refinación a partir de la informatización del proceso Resultados de Refinación en un Período? Este problema se enmarca en el **Objeto de Estudio**: Procesos de evaluación de los indicadores.de refinación

² La consolidación de los balances no es más que la información en un solo dato.

Se propone como **Objetivo General**: Desarrollar una aplicación Web que permita informatizar el proceso Resultados de Refinación en un Período.

Como **Objetivos Específicos**:

1. Estudiar los fundamentos teóricos de los Resultados de Refinación en un Período.
2. Describir el sistema actual, sus deficiencias, funcionalidades y las posibles características a implementar.
3. Realizar un estudio sobre las tecnologías a usar.
4. Diseñar la aplicación de forma modular utilizando la metodología RUP como proceso iterativo e incremental y el modelado a través del Lenguaje Unificado de Modelado (UML).
5. Implementar una aplicación Web para la gestión de evaluación de los indicadores.

El **Campo de Acción** abarcado es: Procesos de evaluación de los indicadores a partir de la informatización del proceso Resultados de Refinación en un Período.

Para guiar la investigación se plantea la siguiente **hipótesis**: Si se perfecciona la aplicación Web que permita automatizar los procesos de evaluación de los indicadores, se logrará optimizar el comportamiento de los indicadores en un período y además la comparación con períodos anteriores.

Para lograr los objetivos planteados anteriormente se trazaron las siguientes **tareas de investigación**:

1. Revisión Bibliográfica de la documentación relacionada con el Módulo Control de SACGIR³.
2. Investigar como se lleva a cabo la evaluación de los indicadores y que tipo de evaluación se realiza.
3. Definir los procesos de evaluación de indicadores: Resultados de Refinación en un Período
4. Realizar el diseño de la aplicación Web utilizando la metodología seleccionada.
5. Implementar una aplicación Web que facilite el trabajo con la evaluación de los indicadores de refinación utilizando el lenguaje seleccionado.

PDVSA cuenta con 21 refinerías, de las cuales 3 se encuentran en territorio nacional estas son las que formaran parte de **la población** de la investigación las cuales son:

1. Complejo Refinador Paraguaná (Esta compuesto por las Refinerías Amuay, Cardón Y Bajo Grande)
2. Refinería El Palito.

³ Sistema Automatizado para el Control y Gestión de los Indicadores de Refinación

3. Refinería Puerto La Cruz.

Además se incluye en la población la Gerencia de Planificación y Gestión de Refinación de PDVSA y la Refinería Isla que se encuentra fuera de territorio venezolano.

Toda la investigación llevada a cabo se realizó en la Gerencia de Planificación y Gestión de Refinación de PDVSA por lo que esta constituirá **la muestra** de la investigación.

Técnica de Muestreo: La técnica de muestreo utilizada fue la No probabilística porque en la misma solo se muestran determinados elementos de la población, dentro de esta técnica se incluye el muestreo intencional pues se escogió como muestra la Gerencia de Planificación y Gestión de Refinación de PDVSA teniendo en cuenta que la misma es uno de los elementos que mayor información puede brindar para realizar la investigación.

Se utilizó **el método de investigación teórico** para que permitiera estudiar las características del proceso de evaluación de indicadores, las cuales no son observables directamente, específicamente se utilizó **el método Análisis histórico lógico** para estudiar si existió o existe alguna aplicación informática en PDVSA y sus funcionalidades.

El método modelación para representar todos los diagramas que ayudarán en la implementación de la aplicación web.

El método análisis síntesis permitirá analizar los documentos de importancia en PDVSA donde se puede extraer la información relacionada con el proceso de evaluación de indicadores.

El presente documento está estructurado por 5 capítulos a continuación se describen brevemente cada uno de ellos:

Capítulo 1: Se analizan conceptos fundamentales del dominio del problema que originan el estudio del estado del arte, así como la descripción de los procesos actuales de la Gerencia de Planificación y Gestión de Refinación de PDVSA. Además de otros sistemas existentes en el mundo con soluciones similares.

Capítulo 2: Se realiza un estudio de las tecnologías, lenguajes, metodologías y herramientas utilizadas, en la realización de SACGIR.

Capítulo 3: Descripción de la solución propuesta. Modelado del negocio y sistema. Descripción de la información que se maneja, de la aplicación que se propone así como de todos los requisitos funcionales y no funcionales con los cuales debe cumplir el sistema.

Capítulo 4: Construcción de la solución propuesta. Incluye la definición del modelo de análisis del sistema y del modelo de clases. Describe los diagramas de colaboración del modelo de análisis para

cada realización de los casos de uso. Muestra el diagrama de clases del diseño, modelo de datos y diagramas de implementación (Componentes y Despliegue).

CAPÍTULO 1

Capítulo # 1 "Fundamentación Teórica"

1.1 Introducción

En el presente capítulo se analizan los conceptos necesarios para lograr una mejor investigación del tema, se describe los procesos actuales de la Gerencia de Planificación y Gestión de Refinación en PDVSA que evalúan los indicadores, además se identifican los principales problemas que motivan la investigación y se analizan otras soluciones existentes a nivel mundial, que pudieran ayudar en la solución de la problemática.

1.2 Conceptos asociados al dominio del problema

1.2.1 Sistema

Sistema es el conjunto de medios interconectados (objetos, seres humanos, informaciones), utilizados según un proceso dinámico con el fin de alcanzar los objetivos señalados. Se considera a un todo formado por partes, estas partes son afectadas por factores internos y externos a ese todo. Además, cada parte debe interrelacionarse e interactuar entre sí y con otras provenientes del medio externo.

También, éste todo debe, a su vez, interactuar como un conjunto lógico para dar respuestas a sus propios requerimientos y a los que el medio externo espera de él. Si cada parte de este todo, actúa e interrelaciona interna y externamente mediante plan, método y orden, se estará entonces, en presencia de un sistema integrado totalmente. La ausencia de plan, método y orden para actuar e interrelacionarse provoca que se coloque frente a un caos. Por otra parte, sistema indica orden; lo opuesto a sistema es el caos. (1)

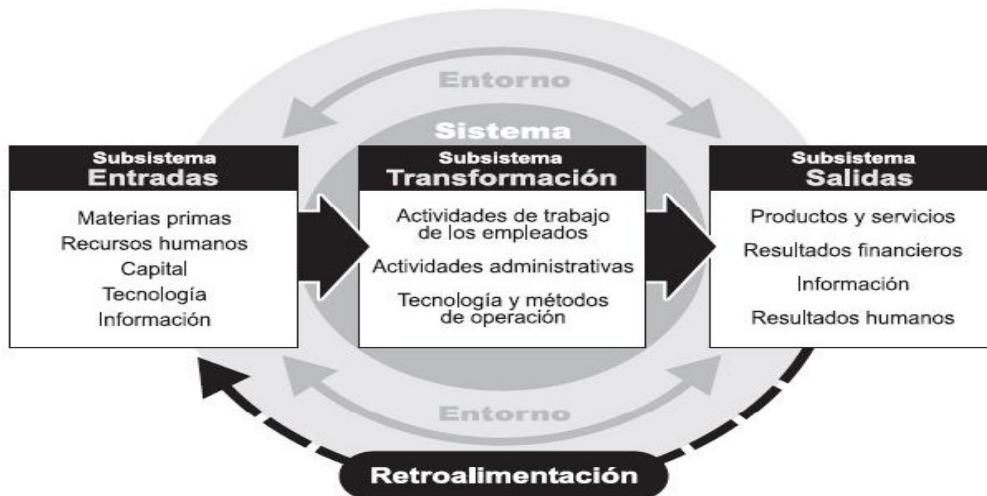


Figura # 1 Representación de Sistema

Sistema es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por cualquiera de las partes individuales. La complejidad de la combinación está implícita. (2)

“Tienen tres características estructurales básicas: Los elementos que lo componen, las relaciones entre los mismos y los límites que determinan los elementos que pertenecen o no al sistema.” (3)

Dado estos planteamientos se puede definir sistema como el conjunto de procesos o elementos interrelacionados (definiciones, nombres, símbolos) con un medio para formar una totalidad encauzada hacia un objetivo común.

1.2.2 Evaluación

Considerada la evaluación en su acepción más amplia, se encuentran con definiciones como la de la Real Academia Española: evaluar es “señalar el valor de una cosa”. Para el *Diccionario del Español Actual*, evaluar significa “valorar (determinar el valor de alguien o de algo)”. Y, en cuanto al término evaluativo/va, en el diccionario mencionado se distingue “un uso evaluativo y un uso descriptor”

Como una primera aproximación a la precisión conceptual del término, podemos decir que la palabra evaluación designa el conjunto de actividades que sirven para dar un juicio, hacer una valoración, medir “algo” (objeto, situación, proceso) de acuerdo con determinados criterios de valor con que se emite dicho juicio. En la vida cotidiana permanentemente estamos valorando sobre todo cuando ponderamos las acciones y decisiones que tomamos. Son formas de evaluación informal, las que no necesariamente se basan en una información suficiente y adecuada, ni pretenden ser objetivas y válidas. Pero cuando queremos evaluar servicios o actividades profesionales no basta la evaluación

informal. Debemos recurrir a formas de evaluación sistemática que, utilizando un procedimiento científico, tienen garantía de validez y fiabilidad. (4)

“La evaluación es una forma de investigación social aplicada, sistemática, planificada y dirigida; encaminada a identificar, obtener y proporcionar de manera válida y fiable, datos e información suficiente y relevante en que apoyar un juicio acerca del mérito y el valor de los diferentes componentes de un programa (tanto en la fase de diagnóstico, programación o ejecución), o de un conjunto de actividades específicas que se realizan, han realizado o realizarán, con el propósito de producir efectos y resultados concretos; comprobando la extensión y el grado en que dichos logros se han dado, de forma tal, que sirva de base o guía para una toma de decisiones racional e inteligente entre cursos de acción, o para solucionar problemas y promover el conocimiento y la comprensión de los factores asociados al éxito o al fracaso de sus resultados.” (5)

1.2.3 Indicadores

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. (6)

Existen tres tipos fundamentales: (7)

Los indicadores de proceso: Se definen como el conjunto de datos obtenidos durante la ejecución del proceso, y referidos a ésta, que permiten conocer el comportamiento del mismo y, por tanto, predecir su comportamiento futuro en circunstancias similares.

Los indicadores de producto: Son el conjunto de datos referidos al producto en sí (medidas obtenidas respecto a medidas previstas, por ejemplo) cuyo análisis indica hasta qué punto se ha conseguido el producto que se deseaba.

Los indicadores de servicio: Igual que los indicadores de producto, son el conjunto de datos referidos al servicio cuyo análisis indica el grado de cumplimiento de los niveles de servicio previamente establecidos.

Los indicadores permiten obtener información rápida del desempeño o evolución de actividades correlacionadas, además hacer comparaciones y estimaciones basándose en información histórica y la evolución del indicador. Son fácil de visualizar, sea en su presentación numérica o gráfica, con un vistazo se puede observar el comportamiento de lo medido.

Según lo planteado los indicadores son valores que permiten conocer el comportamiento de la organización ante un determinado factor, estos sirven para controlar y plantear la futura evolución de la organización. Para cada uno de ellos se deberá establecer la métrica o sistema de medida que se va

a utilizar. Además de las métricas, se debe determinar cada cuánto tiempo se va a analizar el resultado de los mismos.

Entre los indicadores definidos en PDVSA Refinación se encuentran:

Accidentes

Fuerza Laboral

Exportaciones

Salidas

Paradas de plantas

Costo de Refinación

Crudos e Insumos

1.2.4 Evaluación de Indicadores

Uno de los aspectos centrales del uso de indicadores es la evaluación del impacto de los programas. Es decir, poder conocer los efectos que ha tenido determinada política o programa. Mediante los indicadores se miden los cambios en el bienestar de los individuos atribuidos a un programa.

Con base en estos cambios se toman decisiones para ampliar, modificar, eliminar o asignar prioridad a las acciones públicas. Finalmente con la evaluación se aprovecha el potencial de aprendizaje que se deriva de los resultados de los programas, por lo cual es importante seleccionar las acciones públicas que se evaluarán y por lo tanto definir muy bien los indicadores que proporcionarán la información.

Teniendo en cuenta que evaluación tiene que ver con el rendimiento que se obtiene de una tarea, trabajo o proceso. Los indicadores de evaluación están relacionados con los métodos que ayudan a identificar fortalezas, debilidades y oportunidades de mejora. (8)

En PDVSA Refinación se realiza la evaluación de indicadores mensual obteniendo los Resultados Mensuales de cada Refinería que en conjunto con los planes permitirán el cálculo de los indicadores operacionales y financieros. Además se obtienen las Operaciones Diarias, para de esta manera tener una vista en tiempo casi real del proceso de Refinación en la base y por tanto mejor control del mismo. Igualmente se obtiene La Consolidación de los Balances que es el resumen de información variada en un solo dato y representa los Resultados Mensuales así como las Operaciones Diarias, a cualquier nivel del circuito de Refinación.

1.3 Descripción General y Actual del Proceso Evaluación de Indicadores

1.3.1 Proceso de Evaluación de Indicadores

Es preciso para la Gerencia de Planificación y Gestión de Refinación en PDVSA, informatizar el proceso Resultados de Refinación en un Período para mejorar la gestión de evaluación de indicadores y lograr una mejoría en el procesamiento de datos, ajustes de tiempo así como seguridad, precisión y oportunidad de la información.

Para alcanzar la informatización de este proceso hay que tener en cuenta que la Gerencia de Evaluación de Gestión de Refinación son los que llevan los resultados de cada refinería a PDVSA, tienen un departamento de programación, gestión o programación y economía donde dicha Gerencia depende de la Gerencia Técnica, los cuales son sus homólogos a nivel de refinería.

Al final del mes cada refinería emite sus resultados a través del balance volumétrico que es aprobado primeramente por el gerente de refinería y luego el ministerio para después emitir un balance volumétrico consolidado a PDVSA.

En el archivo máster se presenta para todas las refinerías el crudo procesado, el margen bruto unitario (US/barril) y el margen bruto total expresado en valor. Además el margen del circuito como el resultado de dividir el margen bruto total de todas las refinerías entre el crudo total de todas ellas. Aparece un resumen con un histórico de 5 años y el acumulado del período que se trate con el margen bruto de cada refinería y el margen bruto global neto. Se presenta también el acumulado del margen bruto global del sistema.

El Analista de Economía de Refinación de la Gerencia de Refinación revisa los márgenes de contribución por producto por cada refinería y determina aquellos cuyos valores no se correspondan con el esperado, se comunica con el Gerente Técnico y/o de Gestión de cada Refinería para solicitar explicación al respecto o rectificación.

Una vez que han sido revisados todos los márgenes de las refinerías se calcula el del Sistema de Refinación a partir de éstos. Una vez revisados los márgenes de refinerías estos permiten conocer el comportamiento de este indicador en el período y compararlo con períodos anteriores para evaluar su similar comportamiento y de ser necesario investigar las causas. De esta forma se realiza la evaluación para cada unos de los indicadores teniendo en cuenta sus parámetros de obtención. En el caso del indicador parada de plantas se realiza el cálculo de pérdida diaria, porque no solo impactan en otras refinerías sino en todo el sistema.

1.3.2 Situación Problemática

PDVSA al igual que muchas empresas en el mundo se preocupa por llevar a cabo el proceso de evaluación de indicadores para llegar a conclusiones claras, sobre el comportamiento de la empresa, por lo que para esto cuentan con variados sistemas informáticos que apoyan el proceso y a su vez alimentan la gestión de datos para los entes interesados. Específicamente 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. Como se explica anteriormente el apoyo a estos procesos con herramientas automatizadas es decisivo, pues implica ajustes de tiempo así como seguridad, precisión y oportunidad de la información. Además se realiza el proceso de planificación, como apoyo a la producción.

En la Gerencia Corporativa el mayor respaldo de herramientas informáticas profesionales lo tiene el proceso de gestión, con el Sistema de Información de Manufactura (SIM), aunque todavía en más de una Refinería la información se transcribe de forma manual. Además la herramienta tiene aspectos, como la eliminación de transferencias de productos entre Refinerías para la consolidación, que limitan su empleo y obligan al personal a realizar operaciones secundarias con sistemas en Excel, aumentando el margen de error.

Los resultados de la planificación todavía no se gestionan de forma automatizada, aumentando la carga de trabajo de los analistas e incrementando el nivel de posibles errores y de los tiempos de respuesta.

1.4 Descripción del Sistema Actual Existente en PDVSA

El Sistema de Información de Manufactura (SIM) es una aplicación empleada para controlar y apoyar la Gestión de Refinación y Financiera entre las diversas refinerías de PDVSA.

1.4.1 Objetivos

- Apoyar y Controlar la gestión de refinación y financiera entre las diversas refinerías de PDVSA.
- Consolidar la información relevante para la gestión de las refinerías de PDVSA, 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 el proceso de rendición de cuentas ante la Junta Directiva, los entes gubernamentales, así como también para la toma de decisiones en la corporación.

1.4.2 Beneficios

- Permite mantener uniformidad, confiabilidad, consistencia y disponibilidad inmediata de la información de gestión operacional y financiera de las refinerías del Sistema Venezuela + Isla.
- Minimiza el tiempo de procesamiento de la información para el análisis, evaluación y toma de decisiones.
- Facilita la disponibilidad de información a otras organizaciones en PDVSA y entes gubernamentales relacionados.
- Permite la consolidación de la información de las refinerías del Sistema Venezuela + Isla
- Disponibilidad de data histórica.
- Sirve de plataforma a potenciales sistemas gerenciales /corporativos para consulta y toma de decisiones

Por ser una aplicación en tecnología Web permite a sus usuarios conectarse desde todas las áreas a nivel nacional y refinería Isla.

1.4.3 Propiedades

- El usuario posee la facilidad de cambiar su contraseña de ingreso a la aplicación.
- Al cambiar del sitio web Inglés a español o viceversa, automáticamente el usuario queda registrado por el último idioma seleccionado.
- En el Módulo de Carga cuando no se tienen datos registrados para la fecha seleccionada en cierta Refinería, el sistema coloca por defecto los datos de la plantilla de la fecha anterior.
- Las consultas no serán visualizadas hasta los datos cargados para el período y Complejos o Refinerías seleccionadas estén confirmados, a su vez el sistema debe indicar los Complejos o Refinerías que no han sido confirmados en el período seleccionado.
- Para la Consulta del Plan Volumétrico Anual, serán tomados solo los planes confirmados, ya que una refinería en un año puede tener varios planes, pero el oficial es aquel que haya sido confirmado.
- En la opción de Carga de Plan Parada de Planta la información cargada debe ser confirmada inicialmente por el analista de la refinería y luego debe ser confirmada por el analista de Caracas.

1.4.4 Principales Problemas

- En mas de una refinería la información se transcribe de forma manual, el único que la utiliza es el Complejo Paraguaná.
- La herramienta tiene aspectos como la eliminación de transferencias de productos entre refinerías para la consolidación que limitan su empleo. Y obligan al personal a realizar operaciones secundarias con sistemas en Excel, aumentando el margen de error.
- No realiza los procesos de evaluación de indicadores.
- Está desarrollado con lenguaje propietario ASP y como gestor de base de datos Oracle que también es software propietario.

1.5 Análisis de otras Soluciones Existentes

1.5.1 Petróleos Mexicanos (PEMEX)

Petróleos mexicanos es una empresa estatal que realiza integralmente la exploración y explotación del petróleo crudo y gas, el procesamiento de gas natural, la producción de petroquímicos y refinados, así como su comercialización en los mercados nacional y del exterior. PEMEX se integra por un área Corporativa y cuatro organismos subsidiarios dentro de estos se encuentran Pemex Refinación.

Cuentan con el Sistema de de Administración de Mantenimiento (SAM) el cual tiene como objetivo mejorar la disponibilidad, confiabilidad y productividad de los activos de Petróleos Mexicanos y actualizar el censo de los derechos de vía compartidos de los organismos subsidiarios. Tienen implantado un sistema informático para dar seguimiento en tiempo real a la cadena productiva de gas natural-etano en Pemex y para validar los Indicadores del Sistema de Control del Programa de Transparencia y Combate a la Corrupción obteniendo el Reporte de Evaluación Trianual para el período deseado donde se analizan los principales logros y problemáticas que enfrenta la empresa.

La información es vital para evaluar el desempeño de PEMEX en materia de seguridad, medio ambiente y uso de la energía en sus instalaciones industriales para esto tienen implementado el Subsistema de Información de Seguridad Industrial y Protección Ambiental (SISPA). El objetivo del SISPA es fortalecer el proceso de la información en las áreas señaladas y de esta forma, contar con datos oportunos y confiables sobre el estado operacional, incidentes y accidentes, desempeño ambiental y la atención de emergencias en los centros de trabajo de la empresa.

En la actualidad el SISPA cuenta con un conjunto de indicadores ambientales adecuados a cada línea de negocio, característica que permite dar seguimiento al cumplimiento de las metas establecidas y hacer evaluaciones comparativas entre instalaciones similares. (9)

1.5.2 Petróleos Colombianos (Ecopetrol)

Empresa petrolera integrada de crudo y gas, creada en 1951, es la compañía más grande de Colombia generadora de aproximadamente el 20% de los ingresos de la nación.

Ecopetrol cuenta con servicios telemáticos necesarios para soportar los procesos de las áreas de exploración y producción (manejo de información geofísica, geológica, de yacimientos, de interpretación, de producción de crudos a nivel de país, de perforación), transporte (para coordinar el abastecimiento de combustibles a nivel nacional), refinación (desempeño de los procesos productivos en la refinación de hidrocarburos), financiero-contable, gerencial y de apoyo (Proyecto SENSOR el cual es un programa de mejoramiento integral de procesos, organización, tecnología e información en las áreas financieras, de apoyo y gerencial de la empresa). (10)

1.5.3 Petróleos Brasileños (Petrobras)

Petrobras es una compañía de energía que opera en el mercado internacional. A través de sus más de cincuenta años de historia, se ha convertido en una estructura consolidada que le permite abarcar actividades que van desde la exploración de petróleo y gas, producción, transporte, refinación, comercialización y distribución de petróleo gas y sus derivados, hasta la generación, transporte y distribución de energía eléctrica.

Petrobras avanza en la iniciativa denominada "Excelencia en la Gestión" con el fin de alcanzar los más altos estándares en la gestión empresarial. Esta iniciativa se desarrolla a través de un proceso continuo de evaluación e implementación de planes de mejora de la gestión, basado en la Guía Petrobras de Gestión para la Excelencia Cuenta con un sistema informático que su principal objetivo es mantener la información actualizada y certera con evaluación de indicadores importantes para la empresa. Además de un sistema de gestión aplicada a las emisiones atmosféricas que permite: (11)

- Definir las principales fuentes de emisiones atmosféricas.
- Ofrecer un panorama de las tendencias en el tiempo.
- Evaluar sistemáticamente el desempeño ambiental de Petrobras Energía
- S.A. con relación a sus emisiones atmosféricas.

- Evaluar planes de manejo de Calidad de Aire, técnicas disponibles para la reducción de emisiones y aumento de eficiencia.

1.6 Conclusiones Parciales

El estudio de los fundamentos teóricos de la evaluación de indicadores propició desarrollar un marco conceptual de la investigación, además permitió conocer que El Sistema de Información de Manufactura (SIM) no es suficiente para la gestión de indicadores de refinación. Demostrando así que el desarrollo de un sistema de Evaluación de Indicadores resulta beneficioso para la gestión de los mismos.

CAPÍTULO 2

Capítulo 2: Tendencias y Tecnologías Actuales.

2.1 Introducción

Las Tecnologías de la Información y las Comunicaciones (TICs) han permitido llevar la globalidad al mundo de la comunicación, facilitando la interconexión entre las personas e instituciones a nivel mundial, y eliminando barreras espaciales y temporales. Prácticamente todas las empresas e instituciones del mundo están conectadas a la World Wide Web (WWW)⁴ tanto para obtener algún servicio o para prestarlo. En todos los sistemas operativos hay agujeros de seguridad, por lo que estos no son confiables ya que personas ajenas pueden acceder a informaciones que son confiables para la empresa. Debido a esto, los servicios web se modifican diariamente, además de que las tecnologías cambian y aparecen otras con muchas más ventajas para el usuario. Por esta razón en este capítulo se analizarán las tecnologías que se utilizarán en la realización de la aplicación web.

2.2 Lenguaje Unificado de Modelado (UML)

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. El UML (*Lenguaje Unificado de Modelado*) es una herramienta que cumple con esta función, ya que ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien este involucrado en un proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo.

El UML es la creación de Grady Booch, James Rumbaugh e Ivar Jacobson. Estos caballeros, apodados recientemente "Los tres amigos", trabajaban en empresas distintas durante la década de los años ochenta y principios de los noventa y cada uno diseñó su propia metodología para el análisis y diseño orientado a objetos. Sus metodologías predominaron sobre las de sus competidores. A mediados de los años noventa empezaron a intercambiar ideas entre si y decidieron desarrollar su trabajo en conjunto.

Los anteproyectos del UML empezaron a circular en la industria del software y las reacciones resultantes trajeron consigo considerables modificaciones. Conforme diversos corporativos vieron que el UML era útil a sus propósitos, se conformó un consorcio del UML. En 1997 el consorcio produjo la

⁴ World Wide Web, literalmente "malla que cubre el mundo". Servicio de información distribuido, basado en hipertexto, cuya información puede ser de cualquier formato (texto, gráfico, audio, imagen fija o en movimiento) y fácilmente accesible a los usuarios mediante los programas navegadores.

versión 1.0 del UML y lo puso a consideración del OMG⁵ como respuesta a su propuesta para un lenguaje de modelado estándar.

El consorcio aumentó y generó la versión 1.1, que se puso nuevamente a consideración del OMG. El grupo adoptó esta versión a finales de 1997. El OMG se encargó de la conservación del UML y produjo otras dos revisiones en 1998. El UML ha llegado a ser el estándar de facto en la industria del software, y su evolución continua. (12)

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. UML ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de las piezas que constituirán el modelo. (13)

2.2.1 Propiedades de UML

Algunas de las propiedades de UML como lenguaje de modelado estándar son: (13)

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por OMG.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

2.2.2 Ventajas de UML

Una de las ventajas que ofrece UML es referente al desarrollo de aplicaciones globales para la Web, no sólo para comercio electrónico. UML está siendo utilizado por los gerentes de proyectos,

⁵ Grupo de Administración de Objetos que en inglés es Object Management Group (OMG) es una asociación sin fines de lucro, formada por grandes corporaciones, muchas de ellas de la industria del software, como IBM, Apple, Sun Microsystems y HP.

desarrolladores y arquitectos de la Web que aplican técnicas orientadas a objetos para construir aplicaciones Web robustas, escalables y eficientes. UML permite a los desarrolladores modelar sus aplicaciones Web como parte de un sistema completo y la lógica de negocios que se debe reflejar en las aplicaciones.

2.3 Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos de software.

Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

El objetivo de un proceso de desarrollo es subir la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

(14)

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, los resultados finales son impredecibles, además que no hay forma de controlar lo que está sucediendo en el proyecto y se obtienen clientes insatisfechos con el resultado y desarrolladores aún mas insatisfechos.

2.3.1 Proceso Unificado de Desarrollo de Software (RUP)

Rational Unified Process (RUP) es un proceso de desarrollo de software, un marco de trabajo genérico, un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. Utiliza UML para preparar todos los esquemas de un sistema de software. No obstante los verdaderos aspectos definitorios del proceso unificado se resumen en tres frases claves: dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental.

Contiene muchas de las mejores prácticas en el desarrollo de software. Constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. Le proporciona a cada miembro del equipo las pautas, plantillas y herramientas, ayudándolos a que produzcan, dentro de un horario predecible, con un presupuesto razonable y con alta calidad para satisfacer las necesidades de los usuarios. (15)

Como RUP es un proceso, en su modelación define como sus principales elementos: (16)

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de los elementos.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades (“Cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. En la Figura 2: RUP en Dos Dimensiones se representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

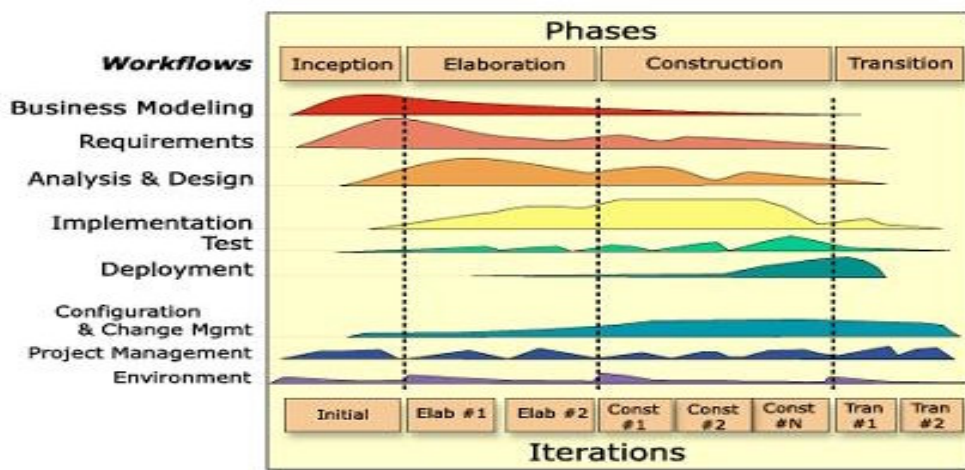


Figura # 2 RUP en Dos Dimensiones

Los Flujos de trabajo definidos por RUP son: (16)

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce el release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios) para entregar el software a los usuarios finales.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

La metodología RUP se divide en 4 fases, las cuales son:_(16)

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un subconjunto de usuarios.

• **Transición:** El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de RUP se caracteriza por: (16)

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. Tal como se aprecia en la Figura 2, el modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

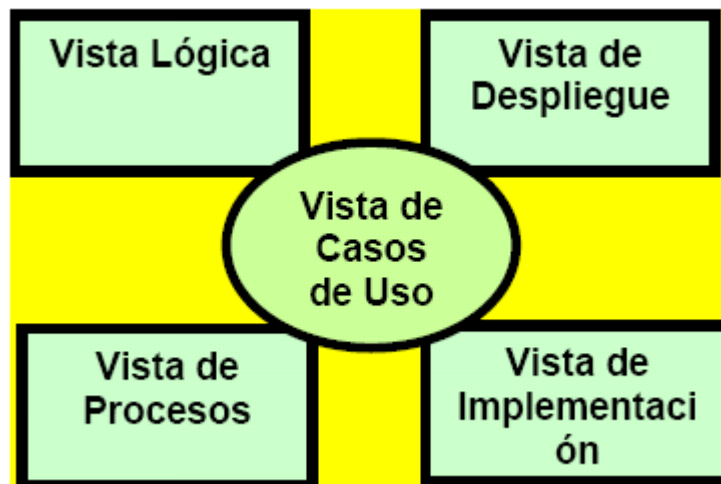


Figura # 3 Vista del modelo de arquitectura.

3. **Iterativo e Incremental:** Aunque la figura 1 puede sugerir que los flujos de trabajo se desarrollan en cascada, la "lectura" de este gráfico tiene que ser vertical y horizontal. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo

incrementalmente en cada iteración. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.



Figura # 4 Proceso Iterativo e Incremental

RUP es un proceso muy general y muy grande, por lo que antes de usarlo habrá que adaptarlo a las características de la empresa.

La vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el proceso unificado de desarrollo, Cada ciclo consta de cuatro fases: Inicio, elaboración, construcción y transición. Cada ciclo concluye con una versión del producto.

Cada fase se subdivide en iteraciones. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software

A continuación se enuncian la organización de modelos propuesta por Rational Unified Process (RUP) (16)

- M. de Casos de Uso del Negocio
- M. de Objetos del Negocio
- M. de Casos de Uso
- M. de Análisis
- M. de Diseño
- M. de Despliegue
- M. de Datos
- M. de Implementación
- M. de Pruebas

Al finalizar un ciclo se obtiene una nueva versión del sistema, cada versión es producto terminado que incluye los requisitos, casos de uso, especificaciones no funcionales y casos de prueba. Incluye el modelo de la arquitectura y el modelo visual -artefactos modelados con el Lenguaje Unificado de Modelado. Todos estos artefactos constituyen una representación del producto que es necesaria para que los desarrolladores puedan llevar a cabo el siguiente ciclo de vida del producto.

2.3.2 Programación Extrema (Extreme Programming (XP))

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, equipo pequeño y cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

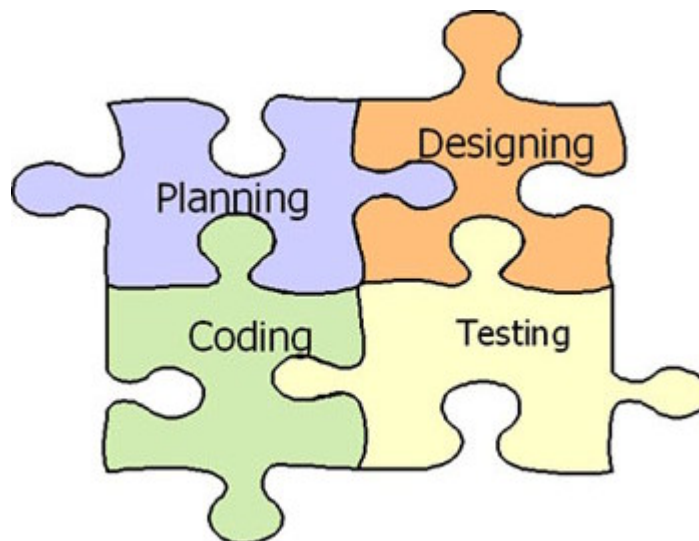


Figura # 5 Metodología Extreme Programming

Características de XP, la metodología se basa en: (15)

Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que si se adelanta en algo hacia el futuro, se puede hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantaran a obtener los posibles errores.

Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto

en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

Empieza en pequeño y añade funcionalidad con retroalimentación continua.

El manejo del cambio se convierte en parte sustantiva del proceso.

El costo del cambio no depende de la fase o etapa.

No introduce funcionalidades antes que sean necesarias.

El cliente o el usuario se convierten en miembro del equipo.

Derechos del Cliente

Decidir que se implementa.

Saber el estado real y el progreso del proyecto.

Añadir, cambiar o quitar requerimientos en cualquier momento.

Obtener lo máximo de cada semana de trabajo.

Obtener un sistema funcionando cada 3 o 4 meses.

Derechos del Desarrollador

Decidir como se implementan los procesos.

Crear el sistema con la mejor calidad posible.

Pedir al cliente en cualquier momento aclaraciones de los requerimientos.

Estimar el esfuerzo para implementar el sistema.

Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

La comunicación, entre los usuarios y los desarrolladores.

La simplicidad, al desarrollar y codificar los módulos del sistema.

La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

2.3.3 Desarrollo Guiado por la Funcionalidad (FDD)

FDD esta pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Las iteraciones se deciden en base a funcionalidades, que son pequeñas partes del software con significado para el cliente.

Un proyecto que sigue FDD se divide en 5 fases: (17)

1. Desarrollo de un modelo general.
2. Construcción de la lista de funcionalidades.
3. Plan de releases en base a las funcionalidades a implementar.
4. Diseñar en base a las funcionalidades.
5. Implementar en base a las funcionalidades.

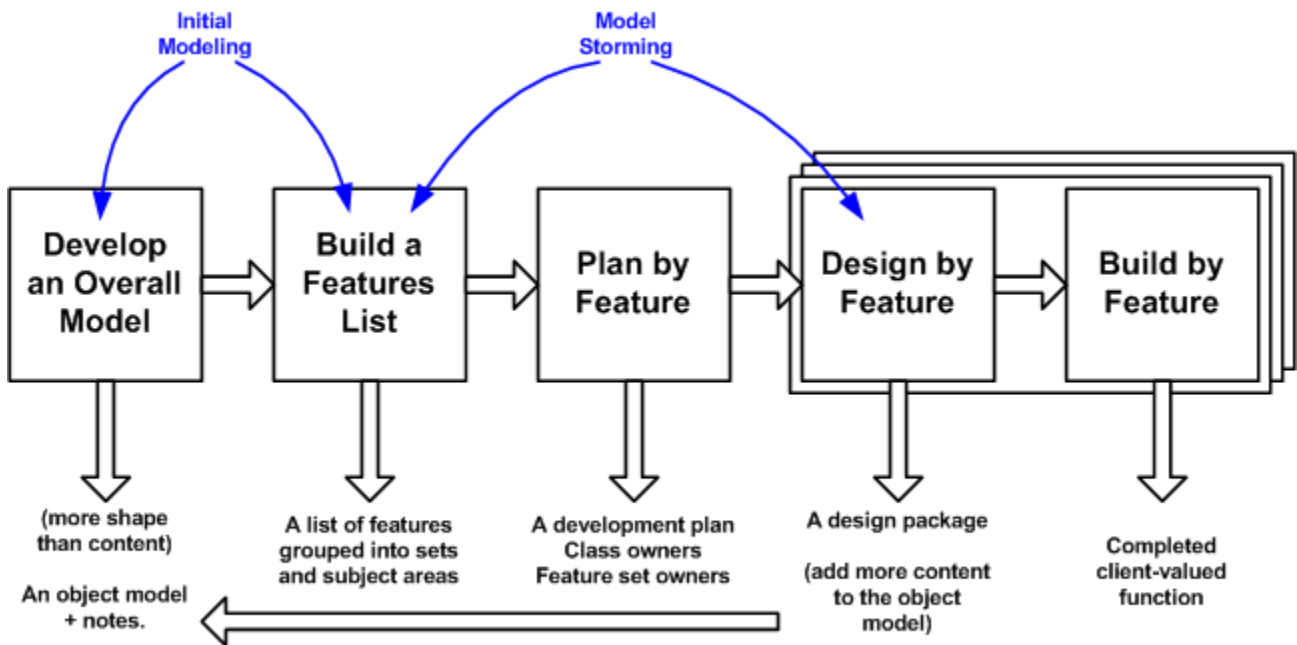


Figura # 6 Vista general de FDD

Las primeras tres fases ocupan parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento.

El trabajo (tanto de modelado como de desarrollo) se realiza en grupo, aunque siempre habrá un responsable último (arquitecto jefe o jefe de programadores en función de la fase en que se encuentre), con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo.

Las funcionalidades a implementar en una release se dividen entre los distintos subgrupos del equipo, y se procede a implementarlas. Las clases escritas tienen propietario (es decir, solo quién las crea puede cambiarlas), es por ello que en el equipo que implementa una funcionalidad dada deberán estar todos los dueños de las clases implicadas, pudiendo encontrarse un programador en varios grupos, implementando distintas funcionalidades. Habrá también un programador jefe (normalmente el más experimentado) que hará las funciones de líder del grupo que implementa esa funcionalidad.

En el proceso de implementar la funcionalidad también se contemplan como partes del mismo (en otros métodos se describen como actividades independientes) la preparación y ejecución de pruebas, así como revisiones del código (para distribuir el conocimiento y aumentar la calidad) e integración de las partes que componen el software.

FDD también define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y que pueden ayudar, además de conocer el estado actual del desarrollo, a realizar mejores estimaciones en proyectos futuros. (17)

2.3.4 Selección de la Metodología a usar

La metodología a usar es RUP puesto que es uno de los procesos más generales de los existentes actualmente ya que en realidad esta pensado para adaptarse a cualquier proyecto, y no solo de software.

Teniendo en cuenta que el proyecto SACGIR es de larga duración y con un equipo de trabajo relativamente grande, para este es más adaptable la metodología de desarrollo RUP, puesto que está considerado para proyectos con estas características. Sin embargo XP y FDD se implementan mejor para proyectos de corta duración y equipos pequeños.

Además una particularidad de RUP es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

2.4 Herramientas CASE de Desarrollo de Software

Las Herramientas CASE⁶ son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Estas herramientas tienen como objetivos: (18)

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Mejorar el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.

⁶ Computer Aided Software Engineering en español Ingeniería de Software Asistida por Ordenador

- Automatizar, desarrollo del software, documentación, generación de código, pruebas de errores y gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la Ingeniería del Software.

2.4.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El lenguaje de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Visual Paradigm ofrece: (18)

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad de integrarse a los principales IDEs⁷.
- Disponibilidad en múltiples plataformas.

2.4.2 Rational Rose Enterprise

Rational Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software (UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Otra ventaja de Rose es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. (19)

⁷ Integrated Development Environment en español Entorno integrado de desarrollo Es una aplicación compuesta por un conjunto de herramientas útiles para los programadores.

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basado en objetos. Se utiliza para modelar un sistema antes de construirlo.

1. Cubre todo el ciclo de vida de un proyecto.
2. Concepción y formalización del modelo.
3. Construcción de los componentes.
4. Transición a los usuarios.
5. Certificación de las distintas fases.

Específicamente Rational Rose Enterprise es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a UML, pero no son compatibles con las mismas tecnologías de implementación.

Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software. Además proporciona mecanismos para realizar la Ingeniería inversa.

2.4.3 Selección de la herramienta a usar

Después de hacer un estudio de estas herramientas se llegó a la conclusión que la mas adecuada para el desarrollo ingenieril es el Visual Paradigm, puesto que la misma permite realizar la transformación de diagramas de Entidad-Relación en tablas de base de datos. Además de contar con una versión libre, siendo esto de gran importancia puesto que SACGIR se está desarrollando bajo el paradigma de Software Libre.

2.5 Aplicación Web

Una Aplicación Web es un Sistema Web que permite a los usuarios ejecutar lógica de negocio a través de un Navegador o lo que es lo mismo, modificar el estado del negocio. Las aplicaciones Web hacen uso de las tecnologías que existen para permitir a los usuarios del sistema modificar la lógica del negocio en el servidor y para generar contenidos dinámicos, de no existir lógica de negocio en el servidor el sistema es considerado como sitio, no como una aplicación Web.

Las aplicaciones Web generalmente presentan una arquitectura simple, como componentes principales usan el servidor Web, la red y el navegador. El servidor es el encargado de distribuir las páginas por cada petición de los clientes. Las peticiones se hacen a través de las conexiones de redes y para

hacerlo utilizan el protocolo de comunicación HTTP⁸. Para mostrar la información a los usuarios y hacer las validaciones en la entrada de los datos se usa siempre un browser o navegador. (20)

2.6 Arquitectura Cliente Servidor.

La Arquitectura de Software es la organización fundamental de un sistema. Se representa a través de sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución. Se concentra en requerimientos no funcionales, que son satisfechos mediante los modelos y diseños de la aplicación. Un diseño correcto de la Arquitectura del sistema es esencial para el éxito o fracaso del proyecto.

La Arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.

Por su parte los servidores realizan, entre otras, las siguientes funciones:

- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

2.6.1 Características de la Arquitectura Cliente/ Servidor

Entre las principales características de la Arquitectura Cliente/Servidor se pueden destacar las siguientes: (21)

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, y otros.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad, capacidades del disco y dispositivos de entrada y salida.

⁸ Hypertext Transfer Protocol en español Protocolo de Transferencia de Hipertexto

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

2.6.2 Ventajas de la Arquitectura Cliente Servidor

Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces más amigables al usuario. De esta manera, se puede integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.

Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario. En el uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la superioridad, con respecto a uno centralizado, de que no es

siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.

Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL⁹ o las herramientas de más bajo nivel como los sockets¹⁰ o el RPC¹¹).

La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.

El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global. (21)

La base arquitectónica del Sistema Automatizado para el Control de Gestión de Indicadores de Refinación (SACGIR) se modelará haciendo uso del Patrón de diseño Modelo Vista Controlador (MVC) 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.

El Patrón Modelo Vista Controlador fue introducido inicialmente en la comunidad de desarrolladores de Smalltalk-80. Según Patrones Orientados a la Arquitectura del Software, es uno de los Sistemas de Patrones de Arquitectura más extendido en el mundo.

El patrón MVC se encuentra en muchos de los diseños de aplicaciones reconocidas por sus interfaces sofisticadas, dando razones para apostar por su aplicación. 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 el 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.

2.6.3 Características del Patrón Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) divide la aplicación interactiva en 3 áreas, para ello utiliza las siguientes abstracciones:

Modelo: Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada. El modelo es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento y además de definir las reglas de negocio.

⁹ Structured Query Language en español Lenguaje de Consulta Estructurado es un lenguaje declarativo de acceso a bases de datos relacionales.

¹⁰ socket (enchufe), es un método para la comunicación entre un programa del cliente y un programa del servidor en una red.

¹¹ Remote Procedure Call en español Llamada a Procedimiento Remoto. Protocolo que permite a una aplicación en una computadora ejecutar código en otra remota sin tener que preocuparse por la comunicación entre ambas.

Vista: Intercambia la información con el usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. Las vistas son responsables de recibir datos del modelo a través del Controlador para mostrar al usuario.

Controlador: Recibe las entradas, traducidas a solicitudes de servicio para el modelo.

El **controlador** es responsable de recibir los eventos de entrada y comunicar las acciones que ellos derivan a la capa de Modelo. (22)

2.7 Lenguajes de Programación

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para las web dinámicas, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

2.7.1 HTML

HTML (HyperText Markup Language) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos ya sea en la computadora u otros recursos de Internet, como FTP¹².

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.

HTML permite: (23)

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

¹² File Transfer Protocol en español Protocolo de Transferencia de Archivos

HTML ha sido desarrollado con la premisa de que cualquier tipo de dispositivo debería ser capaz de usar la información de la Web: PCs con pantallas gráficas con distintas resoluciones y colores, teléfonos móviles, dispositivos de mano, dispositivos de salida y entrada por voz, computadoras con anchos de banda grandes o pequeños, y otros.

Las principales características de HTML son: (24)

- Permite crear lenguajes de codificación descriptivos.
- Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporciona una especificación formal completa del documento.
- No tiene un conjunto implícito de convenciones de señalización.
- Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
- Los documentos generados por él son legibles.

2.7.1.1 Ventajas

HTML es un lenguaje sencillo que permite describir hipertexto es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas. HTML no necesita de grandes conocimientos puesto que es un lenguaje fácil de aprender y es más practicable cuando se cuenta con un editor de páginas web o WYSIWYG¹³. Otras de las ventajas que posee este lenguaje es que lo admiten todos los exploradores.

2.7.2 JavaScript

JavaScript es un lenguaje de programación que se interpreta y se ejecuta del lado del cliente o navegador. Permite agregarle una serie de efectos dinámicos a la Web. Las páginas Web incluyen los scripts, que son las instrucciones que el browser ejecutará. Lo más usual es que el código script esté embebido dentro del código HTML de la página Web. Pero resulta más práctico vincular el código a la página mientras está dentro de un fichero JavaScript de extensión .js, de esta forma se evita tener que cambiar página por página el código script cuando se quiera hacer cualquier modificación de una función en JavaScript.

Muchas de las funciones que tienen las aplicaciones Web se basan en JavaScript. Todos los navegadores actuales para la versión 3 o superior lo admiten sin problemas. Las funciones que pueden hacerse con este lenguaje hacen más rica la navegación entre las páginas Web. En JavaScript se puede trabajar con los eventos onclick, onmouseover, entre otros, que permiten interactuar con el menú, cambiar imágenes dentro de la propia página. Se pueden crear menús desplegables,

¹³ “What You See Is What You Get” en español “Lo que ves es lo que obtienes”. Tipo de aplicación o herramienta aplicable especialmente a los editores de texto que permiten hacer un documento viendo directamente el resultado final.

herramientas de selección de color, alarmas, generar mensajes, calendarios, relojes, juegos sencillos para jugar online, efectos de imágenes, audios y muchos más.

2.7.3 PHP

PHP es un lenguaje de programación utilizado para la creación de sitios web. PHP es un acrónimo recursivo que significa “HyperText Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS¹⁴ con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Entre sus características fundamentales están: (25)

Gratuito. Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.

Gran popularidad. Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código, y que en muchos casos estarán encantados de echar una mano cuando alguna persona se enfrente a algún problema.

Enorme eficiencia. Con escaso mantenimiento y un servidor gratuito, puede soportar sin problema millones de visitas

Sencilla integración con múltiples bases de datos. Esencial para una página Web verdaderamente dinámica, es una correcta integración con base de datos, puede conectarse a PostgreSQL, Oracle, dbm, filePro, interbasem o cualquier otra base de datos compatible con ODBC¹⁵.

Versatilidad. PHP puede usarse con la mayoría de sistemas operativos, ya sea basado en UNIX Linux, Solares, FreeBSD y Windows.

Gran número de funciones predefinidas. A diferencia de otros lenguajes de programación, PHP fue diseñado especialmente para el desarrollo de páginas Web dinámicas. Por ello, está dotado de un gran número de funciones que simplifican enormemente tareas habituales como descargar documentos, enviar correos, trabajar con cookies, sesiones y otros

¹⁴ Internet Information Server en español Servidor de Información de Internet, es el servidor Web de Microsoft que corre sobre plataformas Windows

¹⁵ Open Database Connectivity es un estándar de acceso a bases de datos.

2.7.3.1 Ventajas

PHP es un lenguaje de programación muy fácil de aprender puesto que es muy sencillo a la hora de programar, además de ser muy rápido. Este lenguaje soporta en cierta medida la programación orientada a objetos.

Inicialmente PHP fue diseñado para entornos UNIX por lo que ofrece más prestaciones en este sistema operativo pero es perfectamente compatible con Windows por lo que es considerado un lenguaje multiplataforma. Además tiene la capacidad de conexión con la mayoría de los manejadores de bases de datos: MySQL, PostgreSQL, Oracle y otros.

PHP soporta protocolos de comunicación como LDAP, IMAP, SNMP, NNTP, POP3, HTTP(en Windows) y muchos otros para comunicarse con otros servicios. Puede usar objetos java de forma transparente como si fueran de PHP y la extensión CORBA para acceder a los objetos remotos.

En su página oficial se puede encontrar mucha documentación la cual incluye descripción y ejemplos de cada una de las funciones. Es un lenguaje libre y abierto, muchos lo conocen como el hijo del código abierto, por lo que se presenta como una alternativa de fácil acceso para todos también incluye gran cantidad de funciones. (26)

2.7.3.2 Seguridad

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI¹⁶ separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación. (26)

2.7.3.3 Cake PHP

En el año 2005, Michael Tatarowicz escribió una versión muy básica de una plataforma de desarrollo rápido para aplicaciones PHP. La publicó bajo la licencia MIT¹⁷ apodándola Cake y la abrió a la comunidad de desarrolladores que ahora la mantienen bajo el nombre CakePHP.

CakePHP es una plataforma de software para desarrollo rápido de aplicaciones en PHP. Comprende librerías, clases e infraestructuras de ejecución para programadores que crean aplicaciones web y que originalmente estuvo inspirado en la plataforma de desarrollo Ruby On Rails. Su objetivo primordial es facilitar una forma de trabajar estructurada y rápida, pero sin perder flexibilidad. (27)

¹⁶ Common Gateway Interface, en español Interfaz Común de Comunicación, es un método para la transmisión de información hacia un compilador instalado en el servidor.

¹⁷ Massachusetts Institute of Technology

2.7.3.3.1 Características y Ventajas de CakePHP

CakePHP es un lenguaje de programación que se basa en PHP, el más usado en entornos Web. Tiene una licencia muy flexible, ya que, al usar la licencia MIT, permite no solo usar, copiar o modificar el código, sino que también se puede publicar, sublicenciar o vender copias de la aplicación desarrollada con esta plataforma; además tiene compatibilidad con PHP4 y PHP5, que permite aprovechar las nuevas mejoras que incorpora PHP5. Funciona incluso sin reconfigurar Apache esto facilita mucho el uso de CakePHP en el mayor número de entornos posibles. Incorpora el Scaffolding¹⁸ que es muy útil, especialmente en el período de puesta en marcha de una aplicación.

Otra característica que ahorra tiempo y esfuerzo, son las validaciones de datos incorporadas gracias a las cuales, una vez definidas, por ejemplo, que un campo de la base de datos debe de ser una dirección de email valida, tendrá en cuenta esta validación en los formularios en los que se trabaje con este campo, sin tener que preocuparse de comprobar el contenido cada vez que se vaya a insertar en la base de datos.

Usa una arquitectura Modelo Vista Controlador (MVC), que obliga a seguir una estructura determinada, de la que el desarrollador se beneficia a la hora de mantener o ampliar las funcionalidades del código. También incluye una serie de librerías con funciones de AJAX, Javascript, Formularios HTML, y otros, que permite y se encargan, además de desarrollar más rápidamente, de tener un código completamente basado en la nomenclatura de CakePHP sin necesidad de mezclar distintos tipos de lenguajes.

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

2.8 Servidores Web

Los servidores web almacenan un conjunto de páginas Web como sitio Web así como otro tipo de archivos necesarios para presentar las páginas Web como imágenes, sonidos, bases de datos y otros. Permiten dada una solicitud de información de un cliente hacer un procesamiento interno para presentarle la información de la forma más apropiada.

Los servidores web permiten alojar sitios web y brindar esta información de forma pública o restringida a los clientes. Estas computadoras deben de contar con altos requerimientos de velocidad, memoria y espacio en disco duro para ser capaz de soportar todas las peticiones que se les hace simultáneamente y no colapsar por necesidad de recursos.

¹⁸ Scaffolding es un método para construir aplicaciones basadas en bases de datos y está soportada por algunos frameworks del tipo Modelo Vista Controlador.

2.8.1 Apache

Apache es un servidor Web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación.

2.8.1.1 Características

Este servidor web corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Además es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se esta instalando como servidor, se puede saber, sin ningún secreto, sin ninguna puerta trasera.

También trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod_perl. Igualmente trabaja con Java y páginas JSP¹⁹ y tiene todo el soporte que se necesita para tener páginas dinámicas.

Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.

Es muy configurable en la creación y gestión de Logs²⁰. Apache permite la creación de ficheros de log, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. (28)

2.8.1.2 Ventajas

Entre las ventajas de este servidor se destaca su licencia que es de código abierto del tipo BSD²¹ Desarrollo de Unix llevado a cabo en la Universidad de California. Con estas tres letras se denota este estándar que permite el uso comercial y no comercial de Apache. También se destaca la talentosa comunidad de desarrolladores que sigue un proceso abierto de desarrollo. Además posee una arquitectura modular. Los usuarios de Apache pueden adicionar fácilmente funcionalidades a sus ambientes específicos. Trabaja sobre todas las versiones recientes de UNIX, Linux, Windows y otras.

¹⁹ JavaServer Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

²⁰ En español registro que es un archivo donde se guarda la información de las actividades desarrolladas por los usuarios

²¹ Berkeley Software Distribution

2.9 Sistema Gestor de base de datos

Los Sistemas Gestores de Base de Datos (SGBD) también llamados DBMS por sus siglas en inglés: Database Management Systems, no manipulan documentos, sino que manipulan registros.

Un Sistema de Gestión de Bases de Datos es un programa o aplicación capaz de gestionar adecuadamente las bases de datos. Actualmente casi todos los SGBD implementan los conceptos descritos en la teoría relacional. Un Sistema Gestor de Base de datos Relacional (SGBDR) almacena la información en tablas organizadas lógicamente que se enlazan definiendo relaciones y contienen datos. El lenguaje de consulta SQL (Structured Query Language), que ha sido estandarizado por la ISO²², proporciona la recuperación y gestión de estos datos.

Generalmente las bases de datos manejan transacciones que deben cumplir una serie de propiedades, a las que, comúnmente, se les suele denominar como propiedades ACID (Atomicity, Consistency, Isolation, Durability):

- **Atomicidad:** garantiza que o se ejecutan todas las acciones, o no lo hace ninguna.
- **Consistencia:** garantiza que, aunque haya muchos usuarios accediendo a la base de datos de manera concurrente, se mantenga la integridad de la información.
- **Aislamiento:** garantiza que las transacciones que se están realizando concurrentemente en el sistema no interfieran entre ellas.
- **Durabilidad:** garantiza que una transacción que finaliza correctamente queda adecuadamente reflejada. Además, el sistema será capaz de recordar todas las transacciones que han sido realizadas.

2.9.1 Sistema Gestor de Base PostgreSQL

PostgreSQL es un gestor de bases de datos Objeto-Relacional. Es uno de los sistemas de gestión de bases de datos relacional de código abierto más antiguos, pues la primera versión data de 1985. Está muy extendido en el mundo Unix/Linux ya que muchas distribuciones Linux, como Red Hat lo instalan por defecto, aunque existen versiones para plataformas Windows. Soporta casi todas las construcciones SQL, tiene una amplia conectividad y una gran diversidad de herramientas disponibles. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.

2.9.2.1 Características

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de

²² International Organization for Standardization en español La Organización Internacional para la Estandarización.

conurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos. Es Altamente Extensible puesto que soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario. Además soporta la especificación del estándar SQL99 e incluye características avanzadas tales como las uniones. Tolera además la integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

La flexibilidad del API²³ de PostgreSQL ha permitido a los desarrolladores proporcionar soporte al desarrollo fácilmente para el Sistema Gestor de Base de Datos Relacional (SGBDR) PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP y otras.

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de este gestor de base de datos es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido. Usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

La tecnología que PostgreSQL usa para evitar bloqueos innecesarios es el MVCC²⁴. En otros gestores de base de datos como MySQL hay ocasiones en las que una lectura tiene que esperar para acceder a información de la base de datos. La espera está provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros. Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles. (29)

La característica de PostgreSQL conocida como WAL²⁵ incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual se puede restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos. (30)

2.10 Conclusiones Parciales

Después de haber realizado un profundo análisis de las tecnologías se ha demostrado que las tendencias hacia la utilización de herramientas no propietarias ha ido creciendo cada día más,

²³ Application Programming Interface en español Interfaz de Programación de Aplicaciones.

²⁴ Multi-Version Concurrency Control en español Control de Concurrencia Multi-Versión.

²⁵ Write Ahead Logging.

prácticamente todas las empresas y principalmente las de los países subdesarrollados están emigrando sus sistemas hacia el software libre y por supuesto PDVSA no esta exenta de esos cambios. Se han seleccionado los lenguajes de programación, la metodología de desarrollo y las diferentes herramientas teniendo en cuenta su superioridad.

CAPÍTULO 3

Capítulo 3: Presentación de la Solución Propuesta

3.1 Introducción

En este Capítulo se abordaran temas específicamente de la Ingeniería de Software, es decir artefactos generados en la Fase Inicio de SACGIR, que permitirá entender el entorno de trabajo de PDVSA Refinación , para esto se utiliza el modelado del negocio. Además la especificación de los requisitos funcionales y no funcionales así como la definición de los casos de uso del sistema.

3.2 Modelo de Negocio

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

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 lo que determine los requisitos.

Los objetivos del modelado del negocio son: (31)

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.

3.2.1 Actores 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. Por lo cual los actores del negocio son:

➤ **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, El Balance Volumétrico Consolidado, además es la manera de 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 cada una de las plantas que permiten confirmar la información sobre parada de planta informada, es decir si hay Parada Planta la carga debe bajar.

➤ **Comité de Evaluación de Parada de Planta**

El equipo evaluador (Comité de Parada de Plantas) está conformado por la Gerencia de Planificación y Gestión de Refinación, la Gerencia de Estrategias (Suministro y Logística) y un representante del Comité de Parada de Planta de Refinería, 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.

➤ **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 cada una de las plantas que permiten confirmar la información sobre parada de planta informada, es decir si hay Parada Planta la carga debe bajar.

➤ **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
- 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.

3.2.2 Trabajadores del Negocio

Un trabajador del negocio representa a personas o sistemas dentro del negocio que son los que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio, son los candidatos a convertirse en un futuro en usuarios del sistema que se quiere construir. Por lo cual lo trabajadores del negocio son:

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

Es la persona que se encarga de realizar el cálculo del Margen Bruto y la evaluación de este indicador para conocer su comportamiento, además es quien vela por que se cumpla con los Precios de Ventas en las refinerías a la hora de vender un producto determinado, se encarga de revisar los márgenes recibidos de las refinerías, es quien grafica y evalúa el Margen Bruto y el Margen Neto.

➤ **Superintendente de Economía de refinería**

El la persona que se encarga de calcular el Margen Bruto de la Refinería y de cumplir con que se venda los productos al precio recibido del Analista de Economía de Refinación de la Gerencia de Planificación y Gestión de Refinación se encarga 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 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 período los Indicadores de Seguridad.

3.2.3 Diagrama de Casos de Uso del Negocio

En PDVSA Refinación se llevan a cabo 5 Procesos del Negocio, de los cuales la Gestión de Resultados Mensuales y Evaluación de Indicadores, es el proceso que representa el Módulo Control, a continuación se muestra el diagrama de casos de uso correspondiente a dicho proceso y la descripción de cada caso de uso.

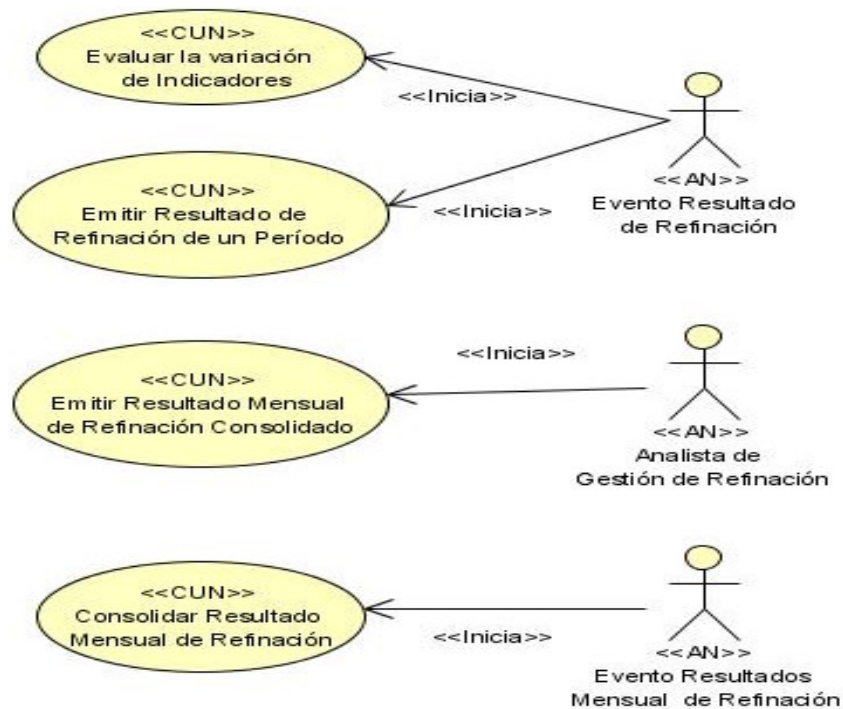


Figura # 8 Diagrama de Casos de Usos del Negocio del Proceso Gestión de Resultados Mensuales y Evaluación de Indicadores

3.2.4 Descripción textual de los casos de uso del Negocio

3.2.4.5 Proceso Gestión de Resultados Mensuales y Evaluación de Indicadores

CUN1- Consolidar Resultado Mensual de Refinación

Tabla # 1 Descripción del Caso de Uso Consolidar Resultado Mensual de Refinación

Caso de Uso:	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.
Precondiciones:	
Referencias:	
Prioridad:	Crítico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Se han obtenido los documentos de Resultado Mensual de Refinación y Recibos y Despachos de las refinerías.	
	2. El Analista de Gestión de Refinación introduce la información recibida de las refinerías en el Documento Máster.
	3. El Documento Máster determina el volumen de transferencia entre refinerías por cada producto.
	4. Calcula la producción de cada producto restando las transferencias.
	5. Calcula las pérdidas o ganancias totales del circuito.
Flujos Alternos	

Acción del Actor	Respuesta del Negocio
Poscondiciones:	
Entidades:	Resultado Mensual de Refinación
Comentarios:	

CUN2- Emitir Resultado Mensual de Refinación Consolidado

Tabla # 2 Descripción del Caso de Uso Emitir Resultado Mensual de Refinación Consolidado

Caso de Uso:	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.
Precondiciones:	
Referencias:	
Prioridad:	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Analista de Gestión de Refinación envía el Resultado Mensual de Refinación Consolidado a los entes relacionados.	
	2. El Gerente General de Planificación y Gestión aprueba el Resultado Mensual de

	Refinación Consolidado.
3. El Analista de Gestión de Refinación envía el Resultado Mensual de Refinación Consolidado, vía correo electrónico, a los entes implicados.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	(En el caso que no lo apruebe) Informa al analista de gestión de los errores.
(Se va al caso de uso rectificar errores)	
Poscondiciones:	
Entidades:	Resultado Mensual de Refinación Consolidado
Comentarios:	

CUN3- Emitir Resultados de Refinación de un Período

Tabla # 3 Descripción del Casos de Uso del Negocio Emitir Resultados de un Período

Caso de Uso:	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.
Precondiciones:	Los Resultados de Refinación de un período deben estar certificados
Referencias:	

Prioridad:	Critico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Los Resultados de Refinación de un Período deben encontrarse certificado.	
	2. El Gerente de Gestión de Refinación presenta los Resultados de Refinación de un Período al Comité y Otras Gerencias.
	3. El Comité Operacional revisa a grandes rasgos los Resultados de Refinación de un Período.
	[Existen Errores] 4. El Gerente de Gestión de Refinación informa los errores al Analista de Gestión de Refinación.
	5. El Analista de Gestión de Refinación envía los Resultados de Refinación de un Período a los entes implicados.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones:	
Entidades:	Presentación de Resultados de Refinación de un Período.
Comentarios:	

CUN4- Evaluar la variación de Indicadores

Tabla # 4 Descripción del Caso de Uso Evaluar la variación de Indicadores

Caso de Uso:	Evaluar la variación de Indicadores
--------------	-------------------------------------

Actores:	Evento Resultados Refinación	
Trabajadores	Analista de Gestión de 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.	
Precondiciones:	Se debe contar con el valor Real y el Plan del indicador.	
Referencias:		
Prioridad:	Critico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Una vez que se tiene el valor Real del indicador y del Plan.		
	2. El Analista de Gestión de Refinación busca el Valor Real del indicador y del Plan.	
	3. Compara el valor Real del Indicador con el Plan.	
	4. Elabora la evaluación cualitativa.	
	5. Registra la evaluación cualitativa.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Poscondiciones:		
Entidades:	Presentación de Resultados de Refinación de un Período.	
Comentarios:		

En el Anexo # 1 se encuentran los diagramas de Actividades correspondientes a cada Caso de Uso del Negocio.

3.3 Requerimientos Funcionales

Los requerimientos funcionales son funcionalidades o capacidades que debe cumplir el sistema por tal motivo a continuación se muestran los requisitos que debe cumplir la aplicación web:

Tabla# 5 Requerimientos Funcionales

R1. Permitir que el “Evaluador” visualice el comportamiento de los indicadores.		
	Descripción	Permitir que la persona encargada de evaluar los indicadores pueda visualizarlos para realizar su evaluación e insertar sus comentarios.
R1.1. Permitir que el Analista de Gestión de Refinación visualice el comportamiento del Volumen de Crudos e Insumos.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Crudos e Insumos.
R1.2. Permitir que el Analista de Gestión de Refinación visualice el comportamiento del Crudo Procesado.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Crudos Procesados
R1.3. Permitir que el Analista de Gestión de Refinación visualice el comportamiento de los Insumos a Procesos.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Insumos a Procesos.
R1.4. Permitir que el Analista de Gestión de Refinación visualice el comportamiento del Costo de Refinación.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Costo de Refinación.
R1.5. Permitir que el Analista de Gestión de Refinación visualice el comportamiento de Exportación de Productos totales y por grupo de productos.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Exportación de Productos.

R1.6. Permitir que el Analista de Economía de Refinación visualice el comportamiento del Margen Bruto Real.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Margen Bruto Real.
R1.7. Permitir que el Analista de Economía de Refinación visualice el comportamiento del Margen Neto Real.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Margen Neto Real.
R1.8. Permitir que el Analista de Economía de Refinación visualice el comportamiento de las Paradas de Planta.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información del indicador Paradas de Plantas.
R1.9. Permitir que el Analista de Economía de Refinación visualice el comportamiento de los Indicadores de Seguridad.		
	Descripción	Permitir que el Analista de Gestión de Refinación visualice la información de los Indicadores de Seguridad.
R2. Calcular el valor acumulado de los indicadores.		
	Descripción	Que el sistema calcule dado un período el comportamiento acumulado del indicador.
R2.1. Calcular el valor acumulado del Volumen de Crudos y Productos		
	Descripción	Que el sistema calcule el acumulado de crudos y productos dato un período.
R3. Permitir que el “Evaluador” pueda visualizar la información fuente de los indicadores.		
	Descripción	Permitir que la persona que se encuentra evaluando el indicador pueda visualizar la información fuente del mismo.
R3.1. Permitir que el “Evaluador” pueda visualizar la información fuente del Volumen de Crudos e Insumos.		
	Descripción	Permitir que la persona que evalúa el indicador Crudo e Insumos pueda visualizar el volumen de los crudos y los insumos por cada

		producto y refinería.
R3.2. Permitir que el “Evaluador” pueda visualizar la información fuente del Crudo Procesado.		
	Descripción	Permitir que la persona que evalúa el indicador Crudo Procesado pueda visualizar el volumen de los crudos procesados por cada crudo y refinería.
R3.3. Permitir que el “Evaluador” pueda visualizar la información fuente de los Insumos a Procesos.		
	Descripción	Permitir que la persona que evalúa el indicador Insumos a Procesos pueda visualizar el volumen de los insumos a procesos por cada producto y refinería.
R3.4. Permitir que el “Evaluador” pueda visualizar la información fuente del Costo de Refinación.		
	Descripción	Permitir que la persona que evalúa el indicador Costo de Refinación pueda visualizar el costo de refinación por cada producto y refinería.
R3.5. Permitir que el “Evaluador” pueda visualizar la información fuente de la Exportación de Productos.		
	Descripción	Permitir que la persona que evalúa el indicador Exportación de Productos pueda visualizar el volumen de las Exportaciones por cada producto, origen y refinería.
R3.6. Permitir que el “Evaluador” pueda visualizar la información fuente del Margen Bruto Real.		
	Descripción	Permitir que la persona que evalúa el indicador Margen Bruto Real pueda visualizar la información de los precios de ventas por productos.
R3.7. Permitir que el “Evaluador” pueda visualizar la información fuente del Margen Neto Real.		
	Descripción	Permitir que la persona que evalúa el indicador Margen Neto Real pueda visualizar la información de los precios de ventas por productos y los costos de refinación.

R3.8. Permitir que el “Evaluador” pueda visualizar la información fuente de las Paradas de Planta.		
	Descripción	Permitir que la persona que evalúa el indicador Parada de Plantas pueda visualizar la información de las Paradas de Plantas por refinerías en un período dado.
R3.9. Permitir que el “Evaluador” pueda visualizar la información fuente de los Indicadores de Seguridad.		
	Descripción	Permitir que la persona que evalúa el indicador Indicadores de Seguridad pueda visualizar la información de los accidentes por refinerías.
R3.10. Permitir que el “Evaluador” pueda visualizar la información fuente de Fuerza Laboral.		
	Descripción	Permitir que la persona que evalúa el indicador Fuerza Laboral pueda visualizar la información de la fuerza laborar de cada una de las refinerías que pertenece al circuito.
R3.11. Permitir que el “Evaluador” pueda visualizar la información fuente del indicador en forma de esquema de grafica.		
	Descripción	Permitir que la persona que evalúa pueda usar gráficos para mostrar la información fuente de algunos indicadores.
R4. Permitir que el “Evaluador” pueda introducir comentarios al comportamiento de los indicadores.		
	Descripción	Permitir que la persona que realiza las evaluaciones pueda introducir comentarios del comportamiento de cada indicador.
R4.1. Permitir que el “Evaluador” pueda introducir comentarios al comportamiento del Volumen de Crudos e Insumos.		
	Descripción	Permitir que el Analista evaluador pueda introducir los comentarios del análisis del comportamiento del indicador Crudo e Insumos.
R4.2. Permitir que el “Evaluador” pueda introducir comentarios al comportamiento del Crudo Procesado.		
	Descripción	Permitir que el Analista evaluador pueda introducir los

		comentarios del análisis del comportamiento del indicador Crudo Procesado.
R4.3. Permitir que el “Evaluador” pueda introducir comentarios al comportamiento de los Insumos a Proceso.		
	Descripción	Permitir que el Analista evaluador pueda introducir los comentarios del análisis del comportamiento del indicador Insumos a Procesos
R4.4. Permitir que el “Evaluador” pueda introducir comentarios al comportamiento del Costo de Refinación.		
	Descripción	Permitir que el Analista evaluador pueda introducir los comentarios del análisis del comportamiento del indicador Costo de Refinación.

3.4 Requerimientos No Funcionales

3.4.1 Requerimientos de Usabilidad

Tabla # 6 Requerimientos de Usabilidad

Preparar a los Administradores en la gestión de Roles y Permisos.		
	Descripción	Esta acción específica del sistema llega un alto grado de complejidad, se debe preparar un curso para instruir a los que trabajaran con estas funcionalidades.
	Prioridad	Secundario
	Estimado	Fin de la segunda versión
Preparar a los Administradores en la gestión de los indicadores		
	Descripción	Esta acción específica del sistema llega un alto grado de complejidad, se debe preparar un curso para instruir a los que trabajaran con estas funcionalidades. En el caso específico del trabajo con la definición de la fórmula del indicador, aunque se tratará de que la interfaz sea lo más amigable posible, es necesario mediante un curso adiestrar a este a rol a que pueda definir cualquier indicador mediante la definición de

		fórmulas.
	Prioridad	Secundario
	Estimado	Fin de la segunda versión

3.4.2 Requerimientos de Confiabilidad

Tabla # 7 Requerimientos de Confiabilidad.

Ante fallas del sistema una vez instalado, corregirlas en un período menor a un mes.		
	Descripción	Aunque el sistema se prevé por los estándares de calidad usados tanto en el proceso como a los que será medido el producto, si este presentará algún desperfecto, este será corregido en un período menor a un mes.
	Prioridad	Secundario
	Estimado	Luego de Desplegada la versión definitiva.

3.4.3 Requerimientos Rendimiento

Tabla # 8 Requerimientos de 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 este 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.
	Prioridad	Alta
	Estimado	Durante el ciclo de desarrollo, a prueba después del despliegue del sistema definitivo.
El sistema debe permitir conectados trabajar concurrentemente 200 usuarios como mínimo.		
	Descripción	La cantidad de usuarios que se encuentran en PDVSA que usarían el sistema no deben parar esa cifra.

	Prioridad	Alta
	Estimado	Durante el ciclo de desarrollo, probado durante las actividades de despliegue de la 1era versión.
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
	Prioridad	Alta
	Estimado	Durante el despliegue.
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.
	Prioridad	Alta
	Estimado	Durante el Despliegue.
El sistema necesita en los clientes con 256 Mb de RAM y un navegador WEB que soporte javascript		
	Descripción	Hay funcionalidades que están desarrolladas en javascript por lo que es imprescindible que los clientes lo soporten.
	Prioridad	Alta
	Estimado	Durante el Despliegue.

3.4.4 Requerimientos de Soporte

Tabla# 9 Requerimientos de Soporte

Es necesario de que los documentos Excel se elaboren en un mismo formato estándar.		
	Descripción	De los documentos Excel se importa gran parte de la información de los planes y precios y costos, por lo que es necesario que mantengan un formato estándar.
	Prioridad	Alto
	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.

3.4.5 Requerimientos de Restricciones de diseño

Tabla # 10 Requerimientos de Restricciones del 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.
	Prioridad	Secundario
	Estimado	Durante el ciclo de desarrollo.

3.4.6 Requerimientos de Adquisición de Componentes

Tabla # 11 Requerimientos de Adquisición de Componentes

El sistema debe de conectarse a una Base de Datos externa.		
	Descripción	El sistema necesita de una conexión con la base de datos del SIM en una primera versión. El sistema necesita de una conexión con la base de datos del SIMP en una segunda versión. El sistema necesita de una conexión con la base de datos del STARS en la segunda versión.
	Prioridad	Alta
	Estimado	Durante el ciclo de desarrollo.

3.4.7 Requerimientos de Interfaz

3.4.7.1 Requerimientos de Interfaces de Usuarios

Tabla # 12 Requerimientos de Interfaces de Usuarios

Interfaz de interacción con el usuario.		
	Descripción	Interfaces amigables, fáciles de interactuar con ellas.
	Prioridad	Alta

	Estimado	Ciclo de Desarrollo.
--	----------	----------------------

3.4.7.2 Requerimientos de Interfaces con otros Software

Tabla # 13 Requerimientos de Interfaces con otros Software

Interfaz de interacción con otros sistemas.		
	Descripción	Interfaz con el SIM, por el protocolo de red TCP Interfaz con el SIMP por el protocolo de red TCP Interfaz con el STARS por el protocolo de red TCP
	Prioridad	Alta
	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.

3.4.8 Requerimientos de Hardware

3.4.8.1 Servidor Web

Tabla# 14 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.
	Prioridad	Alta
	Estimado	Durante el Despliegue.

3.4.8.2 Servidor de Base de Datos

Tabla# 15 Requerimientos de hardware. Servidor de Base 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.

	Prioridad	Alta
	Estimado	Durante el Despliegue.

3.4.8.3 Cliente Web

Tabla# 16 Requerimientos de hardware. 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.
	Prioridad	Alta
	Estimado	Durante el Despliegue.

3.4.9 Requerimientos de Software

3.4.9.1 PC Cliente

Tabla # 17 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.
	Prioridad	Alta
	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.

3.4.9.2 PC Servidor Web

Tabla # 18 Requerimientos de Software. PC Servidor Web

Software instalado en el Servidor Web.		
	Descripción	Servidor Web Apache.
	Prioridad	Alta
	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.

3.4.9.3 Servidor de Bases de Datos

Tabla# 19 Requerimientos de Software. Servidor de Base de Datos

Software instalado en el Servidor de Base de Datos.		
	Descripción	Servidor de Bases de Datos PostgreSQL.
	Prioridad	Alta
	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.

3.4.10 Requerimientos de Seguridad

Tabla# 20 Requerimientos de Seguridad

Seguridad del sistema.		
	Descripción	<p>Confidencialidad: La información manejada por el sistema esta protegida de acceso no autorizado y divulgación.</p> <p>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.</p> <p>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.</p>
	Prioridad	Alta

	Estimado	Ciclo de Desarrollo. Despliegue del Sistema.
--	----------	--

3.5 Descripción del Sistema Propuesto

3.5.1 Descripción de los actores sistema

Tabla # 21 Actores del Sistema

Actor	Descripción
Analista de Evaluación	Rol que generaliza a los actores Analista de Gestión de Refinería, Analista de Economía de Refinería, Analista de Finanzas de Refinería, Analista de Gestión de Refinación, Analista de Economía de Refinería, Analista de Finanzas de Economía. Se encargan de realizar la evaluación una vez que cargan la información gráfica, analizan los comentarios e introducen la evaluación en el sistema, tanto en las refinerías como a nivel de complejo.

3.5.2 Diagrama de Caso de Uso del Sistema (CUS).

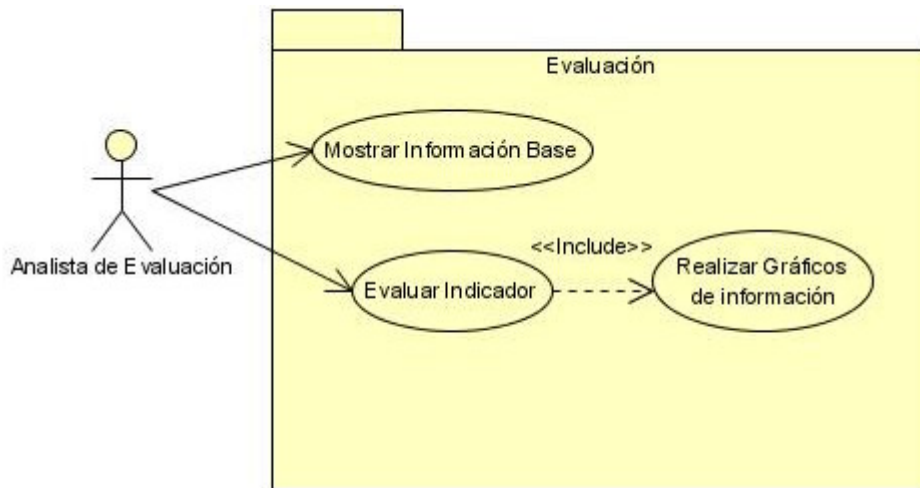


Figura # 9 Diagrama de Casos de Uso del Sistema. Módulo Evaluación

3.5.2.1 Descripción del CUS Evaluar Comportamiento del Indicador

Tabla # 22 Descripción del CUS Evaluar Comportamiento del Indicador

Caso de Uso:	Evaluar Comportamiento del Indicador		
Actores:	Analista de Evaluación		
Resumen:	Permite, al actor con los permisos necesarios, una vez cargada la información gráfica del comportamiento del indicador que se dispone evaluar, realizar la evaluación del mismo y registrar dicha evaluación.		
Precondiciones:	El actor debe tener habilitado los permisos para poder acceder a las gráficas.		
Referencias	R1, R2, R3,		
Prioridad	Crítico		
Flujo Normal de Eventos			
Actor	Sistema		
1- El Analista de Evaluación selecciona la refinería y el período en el que debe evaluar el indicador, selecciona la opción de cargar la información para evaluar el indicador seleccionado.			
	2- El Sistema verifica el tipo de refinería.		
	[Es un complejo] 3- Ir a sección "Evaluar Indicador en Complejo"		
<p>Refinería: Año: Mes: Indicador:</p> <p> <input type="text" value="<< Seleccionar"/> <input type="text" value="<< Seleccionar"/> <input type="text" value="<< Seleccionar"/> <input type="text" value="<< Seleccionar"/> <input type="button" value="Evaluar"/> </p>			
Flujos Alternos			
Actor	Sistema		

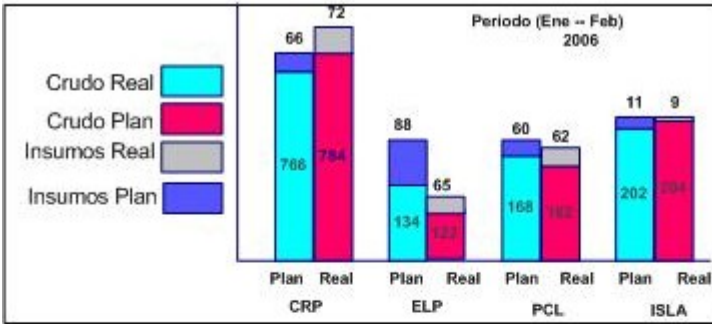
	[No es un complejo] 3a- Ir a Sección “Evaluar Indicador en Refinería”
Sección “Evaluar Indicador en Complejo”	
Actor	Sistema
	1.- El Sistema verifica que el usuario tenga permiso para acceder a la información.
	[Si tiene acceso a la información] 2- El sistema muestra la información a evaluar, desglosada por refinería, los comentarios referidos a la información a evaluar y brinda la opción de registrar nuevos comentarios y modificar los ya existentes.
3- El Analista de Evaluación analiza la información visualizada.	
[Si desea añadir un nuevo comentario] 4- El Analista de Evaluación Registra un comentario o la evaluación en el cuadro de texto.	
5- Selecciona la opción de añadir.	
	6- El Sistema almacena el comentario en la base de datos y pasa la información hacia el cuadro de texto superior.
[Si desea modificar algunos de los Comentarios] 7- El Analista de Evaluación selecciona la opción de Modificar.	
	8- El Sistema pasa la información al cuadro de texto editable, facilitado las correcciones a

	realizar.
9- El Analista de Evaluación modifica la información y selecciona la opción de Añadir.	
	10- El Sistema actualiza la información.
11- El Analista de Evaluación selecciona la opción de Guardar.	
	12- El sistema almacena la información en la base de datos.
Flujos Alternos	
Actor	Sistema
	[Si el usuario no tiene acceso a la información] 2 a- El sistema le niega el acceso a la información.
[Si no desea agregar nuevo comentario] a - Pasa a la acción 11.	
[Si en vez de modificar desea eliminar] 7a- El Analista de Evaluación selecciona la opción de Eliminar.	
	1. El Sistema solicita confirmación elimina el comentario.

Crudos e Insumos

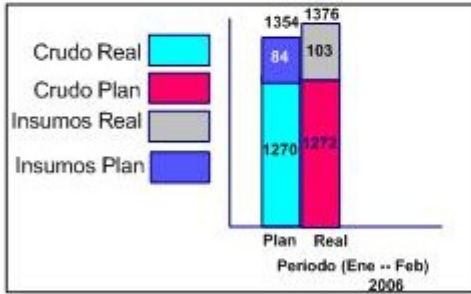
Tipo de Grafica: Barra

Refinerías



- CRP:**
- Mayor procesamiento de crudo en CRP por Incentivo económico.
- ELP:**
- Bajo nivel de procesamiento por dificultades en el atraque de los barcos.

Sistema Nacional de Refinación



Mostrar Información base de:

Crudos e Insumos

Factores que influyeron en el cumplimiento del Plan:

- Mayor procesamiento de Crudo, produciendo más Gasolina.
- Bajos niveles de inventario por problemas en corrientes

El Cr

Añadir Cerrar Cerrar

Sección "Evaluar Indicador en Refinería"

Actor	Sistema
	1- El Sistema verifica que el usuario tenga permiso para acceder a la información de la refinería.
	[Si tiene acceso a la información]

	2- El sistema muestra la información a evaluar, los comentarios referidos a esa información y brinda la opción de registrar nuevos comentarios y modificar los ya existentes.
3- El Analista de Evaluación analiza la información mostrada por el sistema.	
4- Registra un comentario en el cuadro de texto.	
[Si desea añadir un nuevo comentario]	
5- Selecciona la opción de añadir.	
	6- El sistema pasa la información hacia el cuadro de texto superior.
[Desea modificar algunos de los comentarios]	
7- Si el Analista de evaluación desea modificar algún comentario de los que entró al sistema selecciona la opción de modificar	
	8- El sistema pasa la información al cuadro de texto editable.
9- El Analista de evaluación modifica la información y selecciona la opción de añadir.	
	10- El sistema actualiza la información.
11- El Analista de Evaluación selecciona la opción de guardar.	
	12- El sistema almacena la información en la base de datos.
Flujos Alternos	
Actor	Sistema
	[Si el usuario no tiene acceso a la información]
	2 a- El sistema le niega el acceso a la

	información.
[Si no desea agregar nuevo comentario] 5a - Pasa a la acción 11.	
[Si en vez de modificar desea eliminar] 7 a- El Analista de Evaluación selecciona la opción de eliminar.	
Poscondiciones	<ul style="list-style-type: none"> - El sistema debe graficar la información de los indicadores en diferentes tipos de graficas. - El sistema debe mostrar los mensaje de evaluación que se le realizado a indicador.

3.5.2.2 Descripción del CUS Mostrar Información fuente del indicador

Tabla # 23 Descripción del CUS Mostrar Información fuente del indicador

Caso de Uso:	Mostrar Información fuente del indicador
Actores:	
Resumen:	Permite cargar la información base referente a un indicador de un período determinado.
Precondiciones:	Se debe haber ejecutado el caso de uso "Evaluar Comportamiento del Indicador".
Referencias	R51
Prioridad	Crítico
Flujo Normal de Eventos	
Actor	Sistema
	[Si la refinería se comporta como un complejo] 1- Ir a sección "Mostrar Indicador en

Complejo”.

Flujos Alternos

Actor

Sistema

[Si se comporta como una refinería]

1 a- Ir a sección “Mostrar Indicador en Complejo”.

Sección “Mostrar Indicador en Complejo”

Actor

Sistema

1- El Sistema muestra una interfaz con la información base del indicador.

[Si desea ver la información de otra refinería del circuito]

2- El Analista de Evaluación selecciona la refinería de la cual desea ver la información.

3- El Sistema muestra la interfaz con la información base del indicador para la refinería seleccionada.

Crudos e Insumos (MBD)
Sistema de Refinación Nacional

Crudos	Ene	Feb	Ene/feb	Mar	Ene-Mar
Liviano	342,56	244,41	295,98	338,78	310,73
Mediano	648,64	717,88	681,50	640,21	667,28
Pesado	286,08	287,54	286,77	279,44	284,25
Total Crudos	1277,29	1249,83	1264	1258,44	1262
Total Insumos en Procesos	6,87	11,91	9	10,51	10
Total Insumos en Mezcla	104,83	97,35	101	99,91	101
Total de Insumos	111,7	109,26	110	110,42	111
Total Volumen Procesado	1388,98	1359,09	1374,80	1368,87	1373

Isia 

Crudos	Ene	Feb	Ene/feb	Marzo	EneMar
Liviano	145,57	49,48	99,97	142,13	114,49
Mediano	4,15	100,82	50,07	6,17	34,95
Pesado	72,75	75,21	73,91	69,75	72,48
Total Crudos	222,47	225,81	223,96	218,05	222
Insumos en Procesos	3,53	1,26	2,45	6,24	4
Insumos en Mezcla	8,02	3,23	5,74	9,88	7
Total de Insumos	11,55	4,49	8,19	16,12	11
Total Volumen Procesado	234,02	230,10	232,16	234,17	232,85

Cerrar

Flujos Alternos

Actor	Sistema																																																						
[Si no desea ver más información que la mostrada] 2 a- El Analista de Evaluación selecciona la opción de Cancelar.																																																							
Sección "Evaluar Indicador en Refinería"																																																							
Actor	Sistema																																																						
	1- El Sistema muestra una interfaz con la información base del indicador.																																																						
[Si desea ver la información de otra refinería del circuito] 2- El Analista de Evaluación selecciona la refinería de la cual desea ver la información.																																																							
	3- El Sistema muestra la interfaz con la información base del indicador para la refinería seleccionada.																																																						
<p style="text-align: center;">Crudos e Insumos (MBD) Isla</p> <table border="1"> <thead> <tr> <th>Crudos</th> <th>Ene</th> <th>Feb</th> <th>EneFeb</th> <th>Marzo</th> <th>EneMar</th> </tr> </thead> <tbody> <tr> <td>Liviano</td> <td>145,57</td> <td>49,48</td> <td>99,97</td> <td>142,13</td> <td>114,49</td> </tr> <tr> <td>Mediano</td> <td>4,15</td> <td>100,92</td> <td>50,07</td> <td>6,17</td> <td>34,95</td> </tr> <tr> <td>Pesado</td> <td>72,75</td> <td>75,21</td> <td>73,91</td> <td>69,75</td> <td>72,48</td> </tr> <tr> <td>Total Crudos</td> <td>222,47</td> <td>225,61</td> <td>223,96</td> <td>218,05</td> <td>222</td> </tr> <tr> <td>Insumos en Procesos</td> <td>3,53</td> <td>1,26</td> <td>2,45</td> <td>6,24</td> <td>4</td> </tr> <tr> <td>Insumos en Mezcla</td> <td>8,02</td> <td>3,23</td> <td>5,74</td> <td>9,68</td> <td>7</td> </tr> <tr> <td>Total de Insumos</td> <td>11,55</td> <td>4,49</td> <td>8,19</td> <td>16,12</td> <td>11</td> </tr> <tr> <td>Total Volumen Procesado</td> <td>234,02</td> <td>230,10</td> <td>232,16</td> <td>234,17</td> <td>232,85</td> </tr> </tbody> </table> <p style="text-align: center;">Cerrar</p>		Crudos	Ene	Feb	EneFeb	Marzo	EneMar	Liviano	145,57	49,48	99,97	142,13	114,49	Mediano	4,15	100,92	50,07	6,17	34,95	Pesado	72,75	75,21	73,91	69,75	72,48	Total Crudos	222,47	225,61	223,96	218,05	222	Insumos en Procesos	3,53	1,26	2,45	6,24	4	Insumos en Mezcla	8,02	3,23	5,74	9,68	7	Total de Insumos	11,55	4,49	8,19	16,12	11	Total Volumen Procesado	234,02	230,10	232,16	234,17	232,85
Crudos	Ene	Feb	EneFeb	Marzo	EneMar																																																		
Liviano	145,57	49,48	99,97	142,13	114,49																																																		
Mediano	4,15	100,92	50,07	6,17	34,95																																																		
Pesado	72,75	75,21	73,91	69,75	72,48																																																		
Total Crudos	222,47	225,61	223,96	218,05	222																																																		
Insumos en Procesos	3,53	1,26	2,45	6,24	4																																																		
Insumos en Mezcla	8,02	3,23	5,74	9,68	7																																																		
Total de Insumos	11,55	4,49	8,19	16,12	11																																																		
Total Volumen Procesado	234,02	230,10	232,16	234,17	232,85																																																		
Flujos Alternos																																																							
Actor	Sistema																																																						
[Si no desea ver más información que la mostrada] 2a- El Analista de Evaluación selecciona la opción de Cancelar.																																																							

Poscondiciones	El sistema debe de mostrar la información base del indicador que se esta evaluando en el instante.
----------------	--

3.5.2.3 Descripción del CUS Realizar Gráficos de Información

Tabla # 24 Descripción del CUS Realizar Gráficos de Información

Caso de Uso:	Realizar Gráficos de Información	
Actores:		
Resumen:	Permite que visualizar la información mediante esquema en graficas(Barras, líneas, pastel)	
Precondiciones:	Se debe haber ejecutado el caso de uso "Evaluar Comportamiento del Indicador".	
Referencias	R51.13	
Prioridad	Crítico	
Flujo Normal de Eventos		
Actor	Sistema	
	[Si se ejecuto el caso de uso evaluar comportamiento del indicador] <ul style="list-style-type: none"> - El sistema localiza la información del indicador. - El grafica la información del indicador y termina el CUS. 	
Flujos Alternos		
Actor	Sistema	
Post Condiciones	El sistema debe graficar la información del indicador seleccionado.	

3.6 Conclusiones Parciales

Con la realización de este Capítulo se obtuvo una descripción detallada de la propuesta de la solución a través de la modelación del negocio propuesto; identificando los actores y trabajadores que intervienen, así como los Casos de Uso correspondientes. Además se plantean los requisitos funcionales (RF) y los no funcionales (RNF) que debe cumplir el sistema. Igualmente se presenta la modelación de la aplicación en términos de Casos de Uso del Sistema.

CAPÍTULO 4

Capítulo 4: Construcción de la Solución Propuesta

4.1 Introducción

El análisis de un sistema se centra en la investigación del problema y no en la manera de definir una solución, mientras que el diseño pone de relieve una solución lógica: cómo el sistema satisface los requerimientos funcionales, requerimientos de calidad y las restricciones, es decir, en esta parte del proceso de desarrollo del software se decide como se va a llevar a cabo el mismo.

En este capítulo se realiza el análisis y diseño (construcción) de la propuesta de solución, a través de los flujos de Análisis y Diseño e Implementación, se presenta el diagrama de clases del análisis y del diseño de los diferentes casos de usos definidos en el capítulo anterior, además de los diagramas de interacción correspondientes al Análisis, se realiza el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación, además se especifican los principios de diseño. Por ultimo se muestra el diagrama de componentes para una mejor descripción de la solución propuesta.

4.2 Modelo de Análisis

El modelo de análisis es el resultado de un conjunto de artefactos que propone RUP en el flujo de trabajo Análisis y Diseño. El lenguaje que se utiliza en el análisis, se basa en un modelo de objetos conceptual. El modelo de análisis ayuda a refinar las funcionalidades del sistema y permite analizar los aspectos internos del sistema, incluidos sus recursos compartidos. De hecho un recurso interno puede representarse como un objeto en el modelo de análisis. (13)

En este flujo de trabajo se obtienen diferentes artefactos entre ellos se encuentran el diagrama de clases del Análisis y los diagramas de Interacción los cuales serán tratados en los siguientes epígrafes.

4.2.1 Diagrama de Clases del Análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. Las clases que se identifican están asociadas con el contexto del dominio del problema por lo que representan conceptos y relaciones. RUP propone las clases en: (13)

- Clase interfaz:

Las clases interfaz se utilizan para modelar la interacción entre el sistema y sus actores (es decir usuarios y sistemas expertos). Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia) los usuarios y los sistemas externos.

Las clases interfaz modelan las partes del sistema que dependen de sus actores lo cual implica que clarifican y reúnen los requisitos en los límites del sistema. Por tanto, un cambio en una interfaz de usuario o en una interfaz de comunicaciones queda normalmente aislado en una o más clase de interfaz.

➤ Clases entidad:

Las clases entidad se utilizan para modelar información que posee una larga vida y que a menudo es persistente. Las clases entidad modelan la información y el comportamiento asociado a algunos fenómenos, conceptos y sucesos que ocurren en el mundo real. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos que se ha ido elaborando. Algunos autores proponen un estudio del texto, a partir de las frases nominales, de manera que los sustantivos representan objetos y clases.

➤ Clases de control:

Las clases control coordinan el trabajo de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

A continuación se muestran los diagramas de clases del análisis del Caso de Uso del Sistema Evaluación del Comportamiento de Indicadores y Gestionar Visualización de Resultados de Refinación del Módulo Evaluación y del Módulo Acciones de Apoyo a la Evaluación, además de la descripción de las clases pertenecientes a dichos diagrama.

4.2.1.1 Diagrama de Clases del Análisis del CUS Evaluación del Comportamiento de Indicadores.

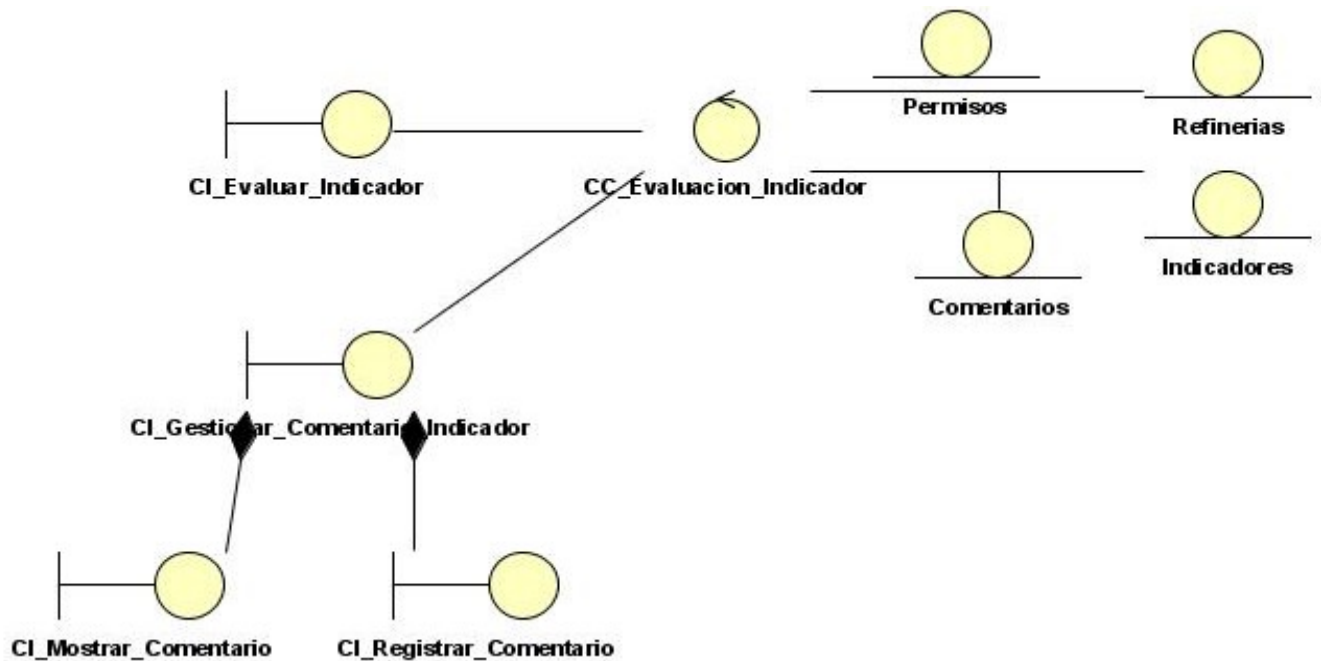


Figura # 11 Diagrama de clases del análisis

En el Anexo # 2 se encuentran los diagramas de clases del análisis de cada caso de uso del sistema.

4.2.2 Diagramas de Interacción

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si se considera el “interior” del sistema, un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje a algún otro objeto, y de esa forma los objetos implicados interactuarán para llevar a cabo el caso de uso. (13)

Los diagramas de interacción están constituidos por dos tipos: los Diagramas de Colaboración y los Diagramas de secuencia. Ambos expresan información similar, pero en una forma diferente.

Los diagramas de secuencia forman parte del modelado dinámico del sistema y proporcionan una vista detallada de los casos de uso. Son diagramas que muestran la interacción organizada de objetos, mediante mensajes que se envían entre sí, en una secuencia de tiempo. Son útiles para observar la vida de los objetos en un sistema, identificar llamadas a realizar o posibles errores del modelado estático que imposibiliten el flujo de información.

Los Diagramas de Colaboración son una forma alternativa al diagrama de secuencia de mostrar un escenario. Este tipo de diagrama muestra las interacciones entre objetos organizadas entorno a los

objetos y los enlaces entre ellos. Proporcionan la representación principal de un escenario, ya que las colaboraciones se organizan entorno a los enlaces de unos objetos con otros.

Los diagramas de colaboración resaltan la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos.

En el Anexo # 3 se encuentran los diagramas de colaboración.

4.3 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización de los CU, y sirve como una abstracción del modelo de implementación y el código fuente. Es usado como una entrada inicial en las actividades de implementación y prueba.

Es usado para concebir un documento del diseño del sistema de SW. Es abarcador, compuesto por artefactos que engloban todas las clases del diseño, subsistemas, paquetes, colaboraciones, y las relaciones entre ellos.

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, o sea el modelo de análisis, que proporciona una comprensión detallada de los requisitos. Además impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema.

4.3.1 Diagrama de Clases del Diseño

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.

Específicamente el diagrama de clases de diseño modela las especificaciones de las clases del sistema y de las interfaces. Las clases del diseño son unos de los artefactos generados en el flujo de trabajo Análisis y Diseño, donde la misma no es más que una construcción similar en la implementación del sistema:

- El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación.
- Las relaciones entre clases de diseño se traducen de manera directa al lenguaje:

- ✓ generalización: herencia
 - ✓ asociaciones, agregaciones: atributos
-
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.
 - Se pueden postergar algunos requisitos a implementación.
 - Una clase de diseño puede proporcionar interfaces si tiene sentido hacerlo en el lenguaje de programación.

A continuación se muestran los diagramas de clases del diseño del Caso de Uso del Sistema Evaluación del Comportamiento de Indicadores y Gestionar Visualización de Resultados de Refinación del Módulo Evaluación y del Módulo Acciones de Apoyo a la Evaluación.

4.3.1.1 Diagrama de Clases del Diseño del CUS Evaluación del Comportamiento de Indicadores .Módulo Evaluación

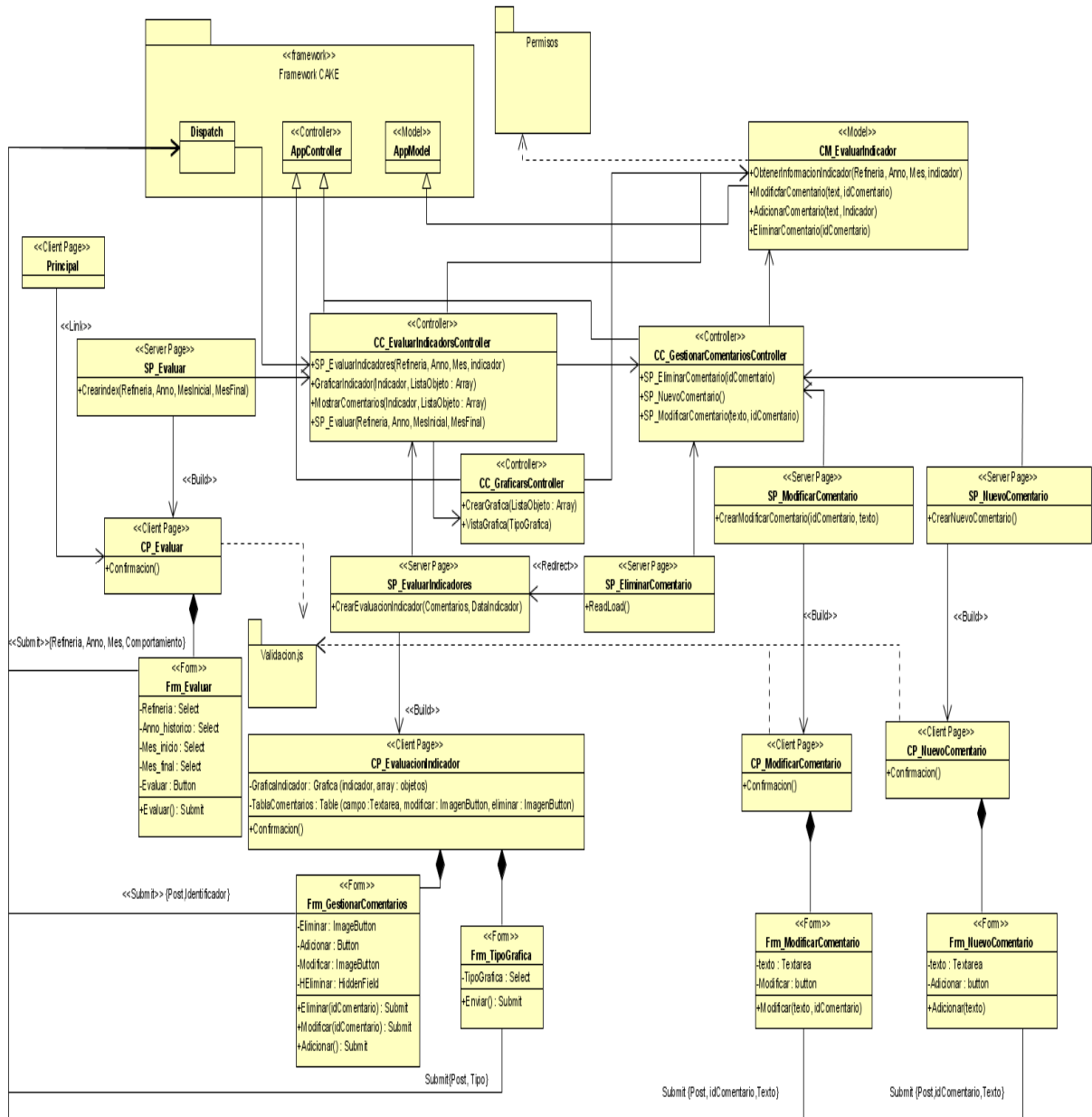


Figura # 13 Diagrama de Clases del Diseño del CUS Evaluación del Comportamiento de Indicadores .Módulo Evaluación

En el Anexo # 4 se encuentran los diagramas de clases del diseño.

4.3.2 Principios del Diseño

La apariencia del sistema está definida por el diseño visual que se realice, sumándole o restándole originalidad al software, y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace. La interfaz de la aplicación debe ser fácil de usar para quienes trabajan con ella de manera que el usuario pueda sentirse cómodo al intercambiar

con el sistema por lo que se ha elegido y se propone una interfaz amigable e intuitiva. Se ha mantenido un diseño consistente en todas las páginas, para lograr que el usuario se sienta satisfecho y logre adaptarse rápidamente a la aplicación.

4.3.2.1 Estándares en la interfaz de la aplicación

La página principal de la aplicación se concibe con la autenticación del usuario en el sistema para acceder a las funcionalidades del mismo. Además la página tiene opciones comunes y visibles para todos los tipos de usuarios.



Figura # 15 Pagina Principal de la Aplicación.

Se crea además un diseño de pagina para después de autenticado el usuario, el menú se encontrara en la parte superior de la pagina donde le permitirá al usuario realizar las operaciones de acuerdo a su rol, además que le permitirá cambiar de rol. El estilo del menú y de los submenús serán los mismos lo único que cambia es el relleno y el formato de negrita a los menús y en los submenús será normal con Color de Fuente rojo. El color que más se destaca es el rojo debido a que ese es el color que identifica a La República Bolivariana de Venezuela. Además que se encuentra el logo de la Empresa.

Bienvenido

El sistema le permite capturar información de las refinерías en otras fuentes, le ayuda a gestionar y evaluar los indicadores de refinación, así como también le presenta dicha información a quien pueda necesitarla.

Esta página fue modificada por ultima vez el 27 de noviembre del 2007, 2:30. Esta página ha sido visitada 60 Veces

Los recursos de Información disponibles en Petropedia responden a una [Política de Protección de Datos](#) tanto para los usuarios como para la Corporación PDVSA 2007

Figura # 16 Página de la Aplicación.

Específicamente en el Módulo Control se realizaran las evaluaciones a los indicadores de refinación donde aparecerán en el lateral derecho, permitiéndole al usuario escoger el tipo de grafica que desee para la mejor comprensión de la evaluación, además de los comentarios correspondientes al Indicador en cuestión, También brinda la opción de mostrar la información base del indicador.

Mensual

Evaluar

Refinería: Año: Mes: Indicador:

COMPLEJO

■ Crudos ■ Insumos

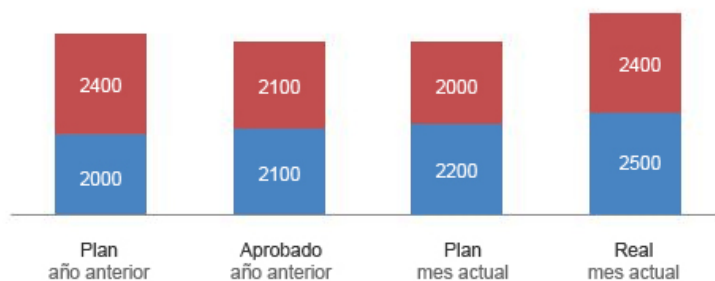


Figura # 17 Página Evaluación de Indicadores

4.4.1 Diseño de la Base de Datos.

Las Bases de Datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para la necesidad de las aplicaciones que la usaran, esto puede ayudar a realizar un proceso del negocio para alcanzar un valor agregado para el cliente.

La puesta en práctica de la base de datos es el paso final en el desarrollo de aplicaciones de soporte del negocio. Tiene que conformarse con los requisitos del proceso del negocio, que es la primera abstracción de la vista de la base de datos. (32)

4.4.1.1 Modelo Entidad Relación

El Modelo Entidad Relación (ER) está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades con sus atributos y de las interrelaciones que existen entre estos objetos. Se desarrolló para facilitar el diseño de Bases de Datos permitiendo la especificación de un esquema del universo de discurso que representa la estructura completa de una Base de Datos. El modelo ER es uno de los diferentes modelos de datos semánticos que existe; el aspecto semántico del modelo reside en su intento de representar el significado de los datos. Este modelo es extremadamente útil para hacer corresponder los significados e interacciones del desarrollo del mundo real con un esquema conceptual. (33)

El Modelo Entidad Relación de SACGIR esta dividido por módulos, específicamente el que generaliza al Módulo Evaluación y Acciones de Apoyo a la Evaluación es el representado a continuación conjuntamente con las descripciones de las entidades:

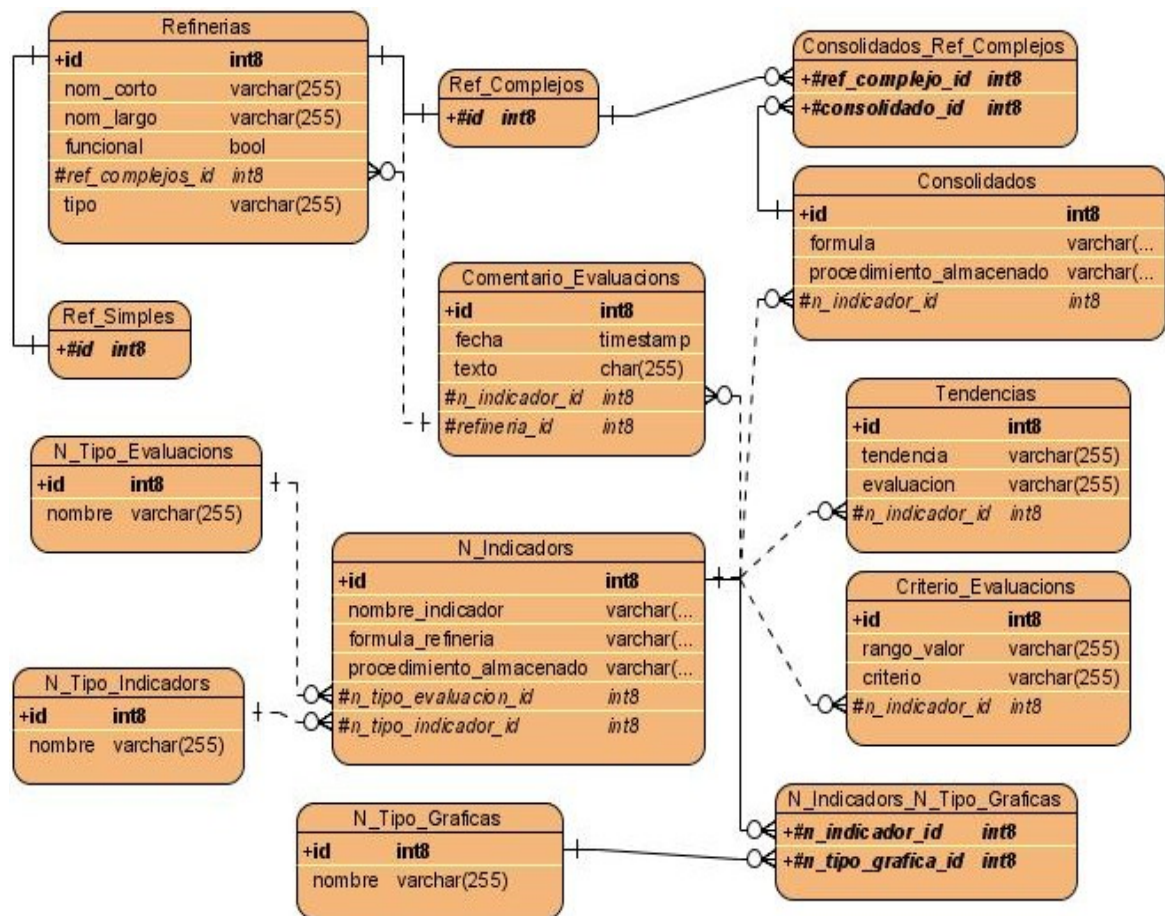


Figura # 18 Modelo Entidad Relación. Módulo Indicadores

4.4.1.2 Diagrama de Clases Persistentes

El Diagrama General de Clases Persistentes está dividido por módulos, específicamente el que generaliza al Módulo Evaluación y Acciones de Apoyo a la Evaluación es el representado a continuación:

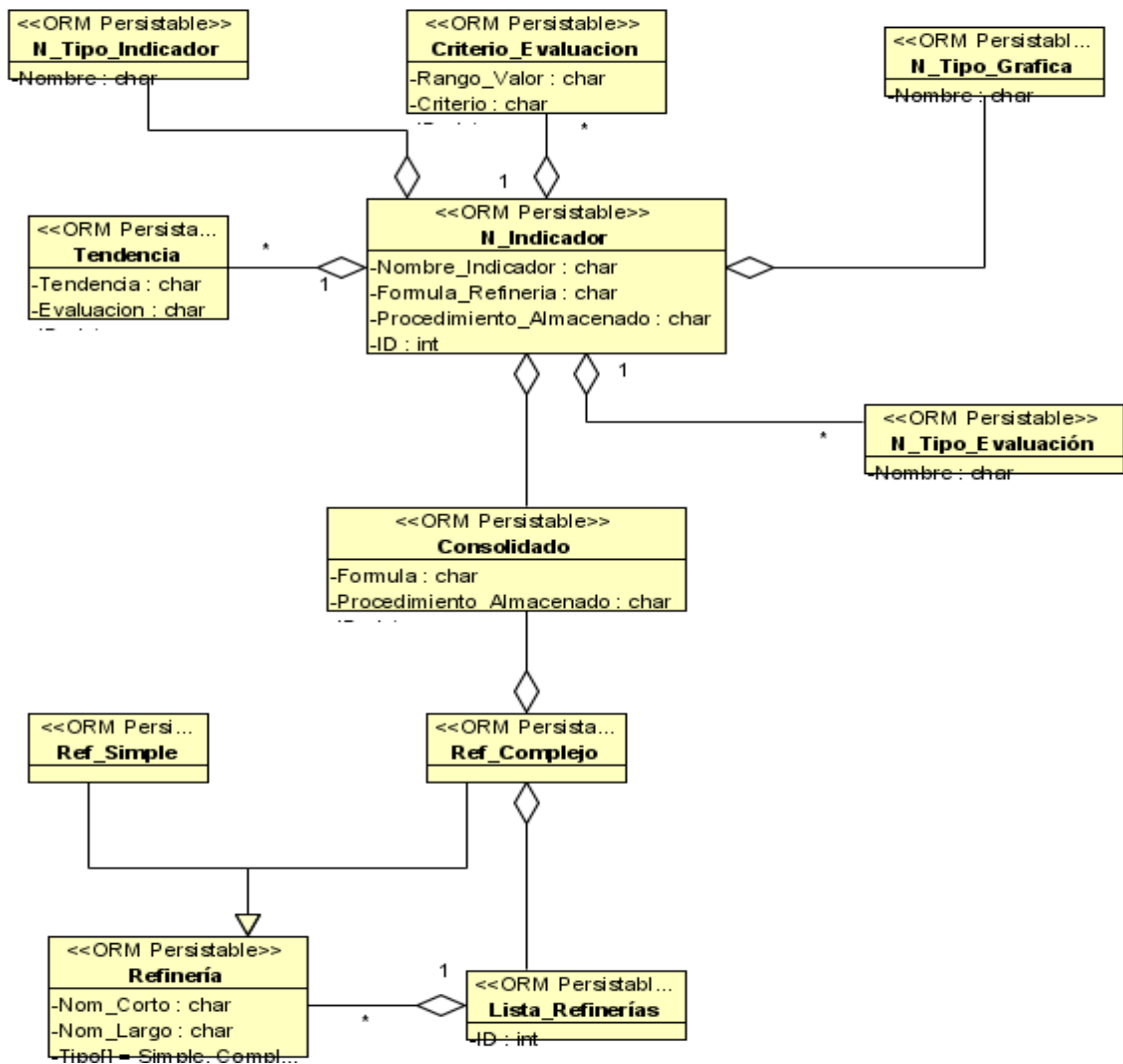


Figura # 19 Diagrama de Clases Persistentes. Módulo Indicadores

4.5 Estándar en la Codificación

PHP es un lenguaje sin chequeo de tipo de variable, esta flexibilidad puede ser tan útil como peligrosa por lo que se adoptaran los siguientes estándares:

1. Las variables deberán comenzar con una letra (en minúsculas) que identifique el tipo de la misma, serían:

Tabla # 27 Estándar de codificación para tipos de variables.

"integer"	Variable de tipo entero	i
"double"	Variables de tipo punto flotante	d

"string"	VARIABLES DE TIPO CADENA	s
"array"	VARIABLES DE TIPO ARREGLO	a
"object"	VARIABLES DE TIPO OBJETO	o
"bool"	VARIABLES DE TIPO BOLEANA	b
"function"	Funciones Variable \$variable() <i>El contenido de la variable se utiliza como nombre de la función a ejecutar</i>	f
"ficheros"	Funciones que representan ficheros lógicos \$pTemp = fopen("Temporal.txt"....);	P

2. Las variables deberán representarse utilizando la **NotaciónCamello**. Se exceptúan los casos en que el framework CAKE dicte algo diferente.
3. En el caso de el alcance de las variables se adoptaran las siguientes reglas solo se *exceptúan los casos en que el framework CAKE requiera algo diferente*.
 - a. Las variables que deban declararse como globales deberán ser marcadas como tal, para ello se le agregará una g (minúscula) al inicio del nombre de la variable, que iría seguido de la letra que representa el tipo de la misma.

\$gsNombreUsuario = "guest";
 - b. Las variables locales comenzarán su nombre con una l (minúscula), seguidos de la variable de tipo de dato.

\$liCount = 0;
 - c. Los nombres de atributos, no requerirán de ningún indicador a su efecto.

\$NombrePersona; //suponer que representa un atributo en la clase Persona.
4. En el caso de las constantes se declaran completamente en mayúscula Ej.

`define(PERMISOS_USUARIOS,"cadena permisos");`
5. La indendación debe ser a cuatro espacios sin caracteres de tabulación

```

if (empty($uri) && defined('BASE_URL')) {
    $uri = setUri();
    if ($uri === '/' || $uri === '/index.php' || $uri === '/.APP_DIR./') {
        $_GET['url'] = '/';
    }
}

```

6. Las estructuras de control deben tener un espacio entre la palabra reservada de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones, y el signo de llaves debe estar sobre la línea de la estructura. Ej.

```

if ($a == 0){

```

7. Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetro, y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

```

$obj->HacerAlgo($var1, $var2, $var3);

```

8. Siempre utilice las etiquetas `<?php ?>` para abrir un bloque de código. No utilice el método de etiquetas cortas, por que esto depende de las directivas de configuración en el archivo PHP.ini y hace que el script no sea tan portable.
9. Si una función, en una clase, es privada; deberá comenzar con el signo de guión mayor para una fácil identificación.

```

private function _HacerAlgo(){

```

10. Usar los modificadores de Visibilidad de php 5. La visibilidad de una propiedad o método puede ser definida al anteponerle a la declaración con las palabras reservadas: **public**, **protected** o **private**. Los elementos declarados con **Public** pueden ser accedidos en todos lados. Los **Protected** limitan el acceso a las clases heredadas (y a la clase que define el elemento). Los **Private** limitan la visibilidad solo a la clase que lo definió.
11. El constructor y destructor de las clases debe definirse como **__construct()** y **__destruct()** respectivamente, según las normas del php 5.
12. Todos los bloques de código deben estar encerrados entre llaves “{ }” aunque tengan una sola línea Ej.

```

if($a >= 10)
$a = 5

```

Error faltan las llaves

```

if($a >= 10){
$a = 5;
}

```

OK

13. En la cabecera de cada fichero debe de incluirse el siguiente comentario en formato phpDoc
Ejemplo.

```

/**
 * <p>Clase Entidad: Accidentes
 *
 * <p>Clase contenedora de informacion referente a
 * todos los accidentes que tengan lugar dentro de una refinería.</p>
 *
 * @filesource /sacgir/Vendors/Entidades/Operacionales/Accidentes.php
 * @author Cesar Santos Sanabria(csanabria@estudiantes.uci.cu)
 * @copyright Copyright 2007-2008, Universidad de las Ciencias Informáticas
 * @package Entidades
 * @subpackage Entidades.Operacionales
 * @modifiedby Cesar Santos Sanabria
 * @lastmodified 2007-10-31 12:00:00 -0500 (Jueves, 31 Octubre 2007)
 *
 * /

```

14. Antes de definir una clase debe ponerse un comentario en formato phpDoc donde se describa las funcionalidades de las clases en cuestión. Ej

```

/**
 * Clase para operar archivos. Escribir, modificar y Leer.
 *
 * @package Entidades
 * @subpackage Entidades. Información
 */
class Fichero {
    ***
    ***
}

```

15. Antes de definir un atributo de una clase debe ponerse un comentario en formato phpDoc donde se describa el atributo y el tipo de datos.

```
/**
 * Nombre del Fichero
 *
 * @var string
 */
public $sFileName = "";
```

16. Antes de definir una función (Método) de una clase debe ponerse un comentario en formato phpDoc especificando una breve descripción de la función y el tipo de datos que retorna así como el de los parámetros que recibe. Ej.

```
/**
 * Escribe un conjunto de datos en el fichero .
 *
 * @param string $data      Datos a escribir en el fichero.
 * @param string $mode Modo de escritura. {@link http://php.net/fwrite See fwrite()}.
 * @return boolean Éxito
 */
function write($data, $mode = 'w') {
    ****
}
```

4.6 Modelo de Despliegue

El Modelo de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto. A continuación se mostrará el diagrama de despliegue que pertenece a SACGIR.

4.6.1 Diagrama de despliegue

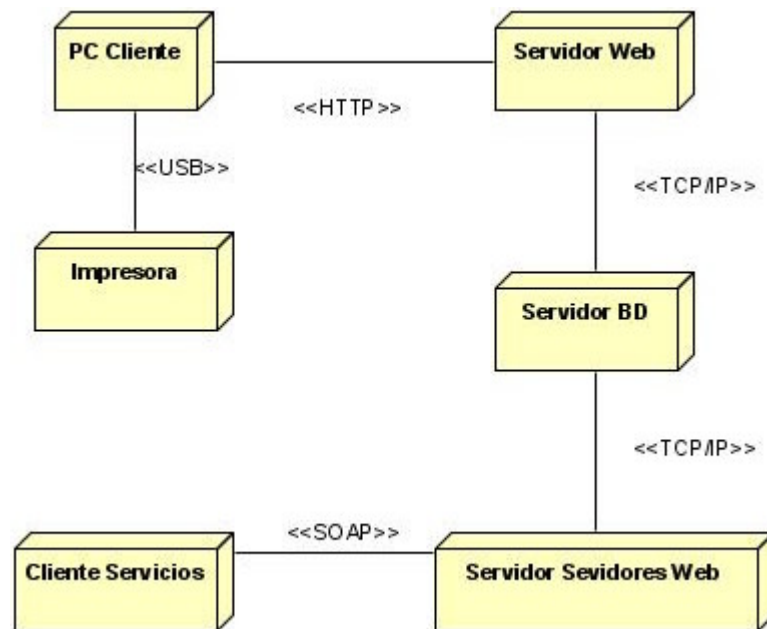


Figura # 20 Diagrama de Despliegue

4.6.1.1 Descripción de los Nodos

Nodo Servidor Web

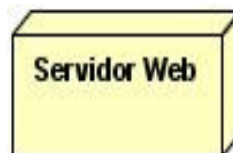


Figura 20.1 Servidor Web.

Requerimientos de Hardware

- ✓ Procesador Pentium D 2x2 cache. 3.2 GHz.
- ✓ Memoria RAM >= 4GB

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

Nodo Servidor de Bases de Datos

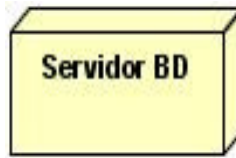


Figura 20.2 Servidor de Bases de Datos.

Requerimientos de Hardware

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

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

Nodo Servidor de Servicios Web

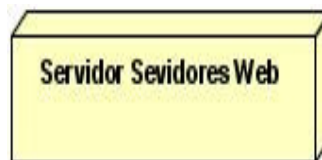


Figura 20.3 Servidor de Servicios Web.

Requerimientos de Hardware

- ✓ Procesador Pentium D 2x2 cache. 3.2 GHz.
- ✓ Memoria RAM \geq 2GB

Subsistemas de Implementación.

En este nodo se atenderán las peticiones de terceras aplicaciones que necesiten consumir los servicios publicados en nuestra solución.

- ✓ Extensión de Capa de Control para Servicios Web.

Nodo Cliente Web



Figura 20.4 Clientes Web.

Requerimientos de Hardware

- ✓ Procesador Pentium 3 (o superior)
- ✓ Memoria RAM mínima de 256 MB
- ✓ Sistema Operativo con un navegador Web compatible con HTML 2.0 y CSS

Subsistemas de Implementación.

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

Nodo Cliente de Servicios Web



Figura 20.5 Cliente de servicios Web.

Requerimientos de Hardware

- ✓ Procesador Pentium 3 (o superior)
- ✓ Memoria RAM mínima de 256 MB
- ✓ Sistema Operativo con soporte para protocolo de comunicación Soap.

Subsistemas de Implementación.

En este nodo se ejecutarán las aplicaciones o sistemas que necesitan consumir los Servicios Web, los requerimientos mencionados son mínimos, las aplicaciones deben establecer requerimientos adicionales si aplica.

4.7 Modelo de Implementación

En la implementación se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Los diagramas de despliegue y componentes, que son artefactos generados en este flujo de trabajo conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará a aplicación. El diagrama de despliegue del sistema ya ha sido abordado en el tema anterior debido a que el mismo se comienza a realizar en el flujo de trabajo Análisis y Diseño, a continuación se representa el diagrama de componentes correspondiente a la Evaluación del comportamiento de Indicadores

4.7.1 Diagrama de Componentes

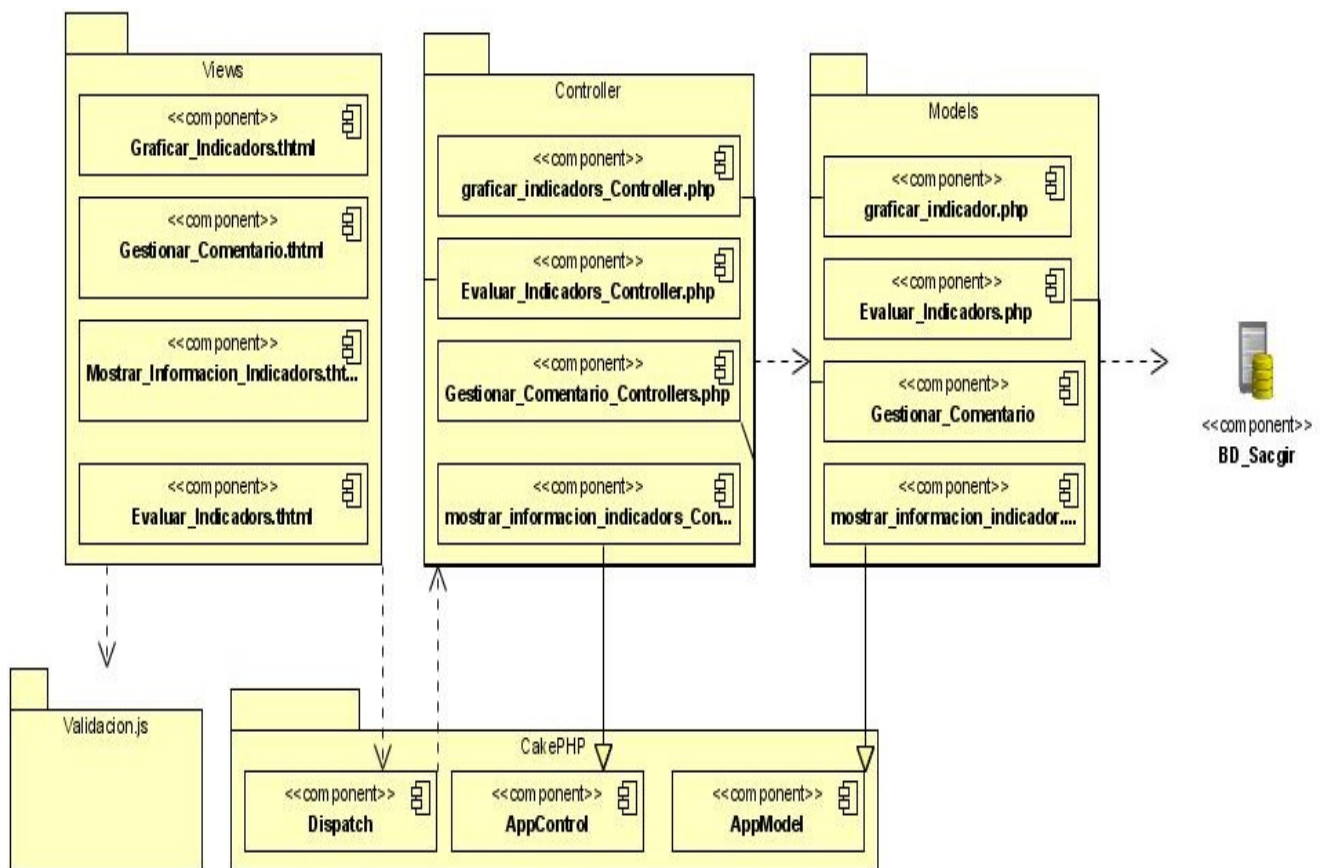


Figura # 21 Diagrama de Componentes

4.8 Conclusiones Parciales

Con la realización de este capítulo se obtiene el diagrama de clases del análisis donde el mismo se realiza utilizando los estereotipos propuestos por UML, permitiendo mostrar las abstracciones mínimas propuestas para la resolución del problema. Además se realiza el diagrama de clases del diseño donde se tiene en cuenta los detalles concretos de la implementación del sistema. Se plasman los principios

del diseño así como los estándares para la realización del diseño de la interfaz de usuario y el diseño de la Base de Datos. También se representa gráficamente el sistema en componentes físicos y nodos mediante el diagrama de componentes y el de despliegue respectivamente.

CONCLUSIONES GENERALES

El valor fundamental de esta Investigación se expresa en la contribución a simplificar el trabajo y la demora que produce el procesamiento manual de la información y mejorar la gestión de la evaluación de los indicadores de refinación en PDVSA.

A partir del análisis de las tendencias y tecnologías actuales que se utilizan para el desarrollo de aplicaciones informáticas, se elaboró como propuesta la utilización de PHP como lenguaje de programación en conjunto con el Framework CakePHP, además de utilizar PostgreSQL como gestor de base datos y Apache como servidor web para la construcción de la propuesta de solución, en conjunto con otras herramientas y tecnologías.

Con la realización de este trabajo se obtiene la propuesta que se propuso de una aplicación que eliminara el trabajo manual reemplazándola por una aplicación Web de fácil uso, entendible y desarrollada con nuevas herramientas. La aplicación permite:

- Navegación placentera a través de las diferentes secciones de trabajo.
- Interfaz cómoda, amigable, de fácil entendimiento y sobre todo de facilidad de uso por los usuarios.
- Mejora la calidad de los procesos de refinería como, Resultados de Refinación en un Período.
- Con la implementación de la aplicación Web se facilita el trabajo con la evaluación de indicadores de refinación.

Además se cumple con el objetivo propuesto de desarrollar una aplicación Web que permita informatizar el proceso: Resultados de Refinación en un Período.

RECOMENDACIONES

Se recomienda:

- Que este trabajo de diploma sea considerado como material de consulta por las personas que deban realizar un sistema similar o que la teoría que se ha expuesto en este trabajo le sirva de apoyo en futuros proyectos.
- Hacer un estudio más profundo sobre el proceso de Gestión de Evaluación de Indicadores de refinación para analizar la posible automatización de otras funciones que no se implementaron en la propuesta de solución.
- Mantener una actualización periódica sobre el sistema, logrando así que se mantenga la fiabilidad y funcionamiento óptimo del mismo así como la integridad de la información que se gestiona a través de él.

REFERENCIAS BIBLIOGRÁFICAS

1. **Edgardo Tejada, Carlos.** Ideas Orientadoras sobre el Sistema Logístico del Componente Ejército del Teatro de Operaciones. *usacac.leavenworth.army.mil.* [En línea] <http://usacac.leavenworth.army.mil/CAC/milreview/download/Spanish/NovDec01/tejada.PDF..>
2. IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONIC TERMS. [En línea]
3. **PONJUÁN DANTE, G.** Aspectos introductorios acerca de los sistemas en: Sistemas de información: Principios y aplicaciones. [En línea] Félix Varela, 2004.
4. **Ramírez, Ana Cano.** Aspectos Para una definición de evaluación. [En línea] 2006. http://www.ulpgc.es/hege/almacen/download/38/38196/tema_5_elementos_para_una_definicion_de_evaluacion.pdf.
5. **Egg., Ander.** Definición de Evaluación. [En línea] 2000).
6. Definición de indicador. *www.definicion.org.* [En línea] <http://www.definicion.org/indicador>.
7. Definición de indicador. *calidad.unad.org.* [En línea] http://calidad.unad.org/asesoramiento/definicion_de_indicadores.html.
8. **Sandoval De Escurdia, Juan Martín.** Evaluación de Indicadores. [En línea]
9. Petroleos Mexicanos. *www.pemex.com.* [En línea] <http://www.pemex.com>.
10. Petroleos Colombianos. *www.ecopetrol.com.co.* [En línea] <http://www.ecopetrol.com.co>.
11. *www.comunicarseweb.com.ar.* [En línea] <http://www.comunicarseweb.com.ar/directorio/petrobras.html>.
12. **Schmuller, Joseph.** Aprendiendo UML en 24 Horas. [En línea] 2000. <http://bibliodoc.uci.cu/pdf/reg00004.pdf>.
13. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Lenguaje Unificado de Modelado. [En línea] 1977. <http://bibliodoc.uci.cu/pdf/reg03050.pdf>.
14. **Barzanallana, Rafael.** Universidad de Murcia. *www.um.es.* [En línea] <http://www.um.es/docencia/barzana/IAGP/lagp2.html>.
15. **Mendoza Sanchez, María Antonia.** Informatizate. *www.informatizate.net.* [En línea] 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
16. Conferencia # 1 Introduccion a la Ingeniería de Software. *Teleformacion.uci.cu.* [En línea] <http://Teleformacion.uci.cu>.
17. ¿Cual XP, RUP, FDD? *FeatureDrivenDevelopment.com.* [En línea] <http://www.featuredrivendevelopment.com>.
18. Una Herramienta Case para Análisis y Diseño Orientado a Objetos. *Universidad de Castilla la Mancha.* [En línea] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.
19. Rational.com. [En línea] <http://www.rational.com>.
20. **Rodríguez Osoria, Danieski.** Trabajo de Diploma para optar por el título de Ingeniero Informático. Diseño y Servicios Web par el Registro de Áreas de Salud de la Atención Primaria del Sistema de Informacion para la Salud. [En línea] http://bibliodoc.uci.cu/TD/TD_0673_07.pdf.

21. **Márquez Avendaño, Bertha Mariel y Zulaica Rugarcía, José Manuel.** Trabajo para optar por el título de Licenciatura en Ingeniería en Sistemas Computacionales. Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español. [En línea] 2004. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.
22. **Fernández, Hernán Darío.** Tutorial Introducción al Framework Web de Jakarta Struts con WSAD 5.1. [En línea] 2004. http://www.scribd.com/full/97147?access_key=ilbwh1c3ih4vg.
23. Especificación HTML 4.01. *html.conclase.net*. [En línea] <http://html.conclase.net/w3c/html401-es/cover.html>.
24. **Rufo Martin, Pedro.** HTML. *ucistore.uci.cu*. [En línea] <ftp://ucistore/Documentation/ASP%20y%20HTML/HTML/CBhtml.pdf>.
25. **Gallego Vazquez, José Antonio.** Desarrollo Web con PHP y MySQL. *bibliodoc.uci.cu*. [En línea] <http://bibliodoc.uci.cu/pdf/reg02138.pdf>.
26. **Pérez Valdéz, Damian.** Los diferentes lenguajes de programación para la web. *maestrodela web.com*. [En línea] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
27. **Ramirez, Javier.** Programacion Rapida con CakePHP. *www.linux-magazine.es*. [En línea] <http://www.linux-magazine.es/issue/23/CakePHP.pdf>.
28. **Mateu, Carlos.** Software Libre. *www.uoc.edu*. [En línea] http://www.uoc.edu/masters/softwarelibre/esp/materials/Desarrollo_web.pdf.
29. **González, Carlos D.** Base de Datos PostgreSQL, SQL avanzado y PHP. *www.usabilidadweb.com.ar*. [En línea] <http://www.usabilidadweb.com.ar/postgre.php>.
30. **Worsley, John y Drake, Joshua.** PostgreSQL Práctico. *www.sobl.org*. [En línea] <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
31. Conferencia de Ingeniería de Software. Modelo de Negocio. *teleformacion.uci.cu*. [En línea] <http://teleformacion.uci.cu>.
32. Conferencia Introductoria de Base de Datos. *teleformacion.uci.cu*. [En línea] <http://teleformacion.uci.cu>.
33. Conferencia de Bases de Datos. Modelo Entidad Relacion. *teleformacion.uci.cu*. [En línea] <http://teleformacion.uci.cu>.

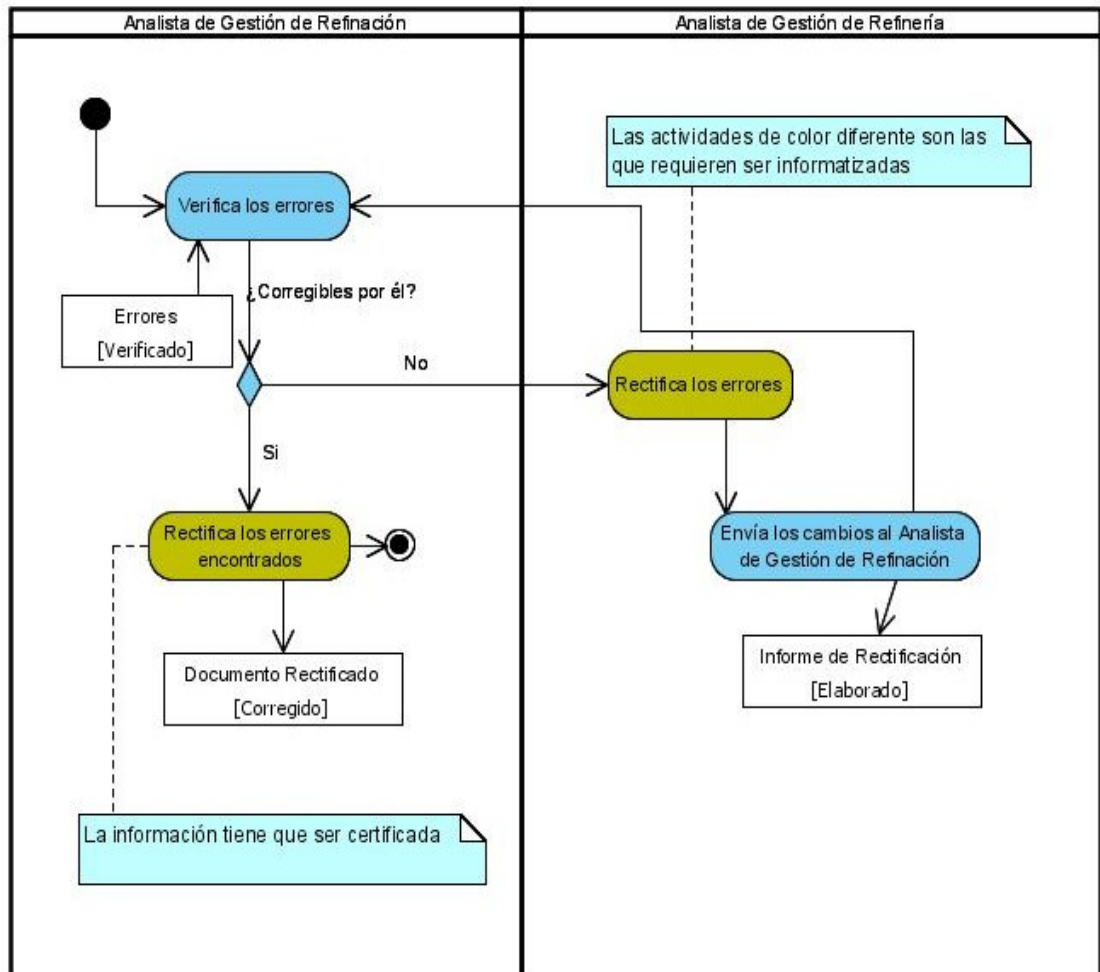
BIBLIOGRAFÍA

1. **Lamarca Lapuente, María Jesús.** SGBD y STRID. *www.hipertexto.info*. [En línea] 2008. <http://www.hipertexto.info/documentos/sghbd.htm#SGBD>.
2. **Márquez Díaz, José, Sampedro, Leonardo y Vargas, Félix.** Instalación y configuración de Apache, un servidor Web gratis. *ciruelo.uninorte.edu.co*. [En línea] http://ciruelo.uninorte.edu.co/pdf/ingenieria_desarrollo/12/instalacion_y_configuracion_de_apache.pdf.
3. Una Introducción a Apache. *linux.ciberaula.com*. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro.
4. Petroleos de Canada. *www.emcsoftware.es*. [En línea] http://www.emcsoftware.es/about_us/customer/profiles/petro_canada.htm.
5. Funcionamiento de un Servidor Web. *teleformacion.uci.cu*. [En línea] <http://teleformacion.uci.cu/mod/book/view.php?id=2060&chapterid=615>.
6. **López Cerezo, José Antonio y Sánchez Ron, José Manuel.** Ciencia, Tecnología, Sociedad y Cultura en el cambio de siglo. *Biblioteca.uci.cu*. [En línea] <http://bibliodoc.uci.cu/pdf/8470309129.pdf>.
7. **Larman, Craig.** UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos. *Biblioteca.uci.cu*. [En línea] <http://bibliodoc.uci.cu/pdf/reg00061.pdf>.
8. **Pressman, Roger S.** Ingeniería de Software. Un Enfoque Práctico. *biblioteca.uci.cu*. [En línea] <http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
9. **González, Fabio Antonio.** Diseño, Construcción y Mantenimiento de Sistemas de Software Grandes. *Departamento de Ingeniería de Sistemas e Industrial*. [En línea] <http://dis.unal.edu.co/~fgonza/courses/2003/ingSoft1/CAP1.pdf>.
10. **Feamster, Nick.** La Seguridad en la Ingeniería de Software. *www.acm.org*. [En línea] <http://www.acm.org/crossroads/espanol/xrds7-4/onpatrol74.html>.
11. **García Fanjul, José y Álvarez, Claudio de la Riva.** Diagramas de Clases(En los Procesos del Análisis). *Universidad de Oviedo. Departameto de Informática*. [En línea] <http://www.di.uniovi.es/~claudio/isoft/recursos/DCLA.pdf>.
12. **Astudillo, Hernán y Visconti, Marcello.** Fundamentos de Ingeniería de Software. *Departamento de Informática.Universidad Técnica Federico Santa María*. [En línea] <http://www.inf.utfsm.cl/~visconti/ili236/Documentos/10-DisenoOO.pdf>.
13. **Valencia, María Eugenia.** Diagrama de clases del Diseño. *Dpto. Ciencias de la Computación.Escuela de Ingeniería de Sistemas y Computación*. [En línea] http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISCLASES_A12.pdf.
14. Manual de CakePHP. *cakephp.org*. [En línea] <http://cakephp.org/>.
15. **Ferré Grau, Xavier y Sánchez Segura, María Isabel.** Desarrollo Orientado a Objetos con UML I. *mononeurona.org*. [En línea] <http://www.mononeurona.org/pages/display/549>.
16. UML Home Page. . *Object Management Group (OMG)*. [En línea] <http://www.uml.org/>.

17. **Fernandez Vilas, Ana.** Diagramas de Componentes. *ww-gris.det.uvigo.es*. [En línea] <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.
18. **Figueroa, Pablo.** Conceptos en un Diagrama de Implementación. [En línea] <http://www.cs.ualberta.ca/~pfiguero/soo/uml/implementacion01.html>.
19. Diagrama de Componentes UML 2. *www.sparxsystems.com.ar*. [En línea] http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html.
20. **de Diego Rodríguez, Jose Ramón.** Diagramas de Colaboración según Métrica V3. *Ingeniería de Software. Universidad de Oviedo*. [En línea] http://www.di.uniovi.es/~dediego/is/recursos/d_int.pdf.
21. **Vizcaíno, Aurora, García, Felix y Caballero, Ismael.** Prácticas de Ingeniería de Software. Diagramas de Interacción. Análisis y Diseño Orientado a Objetos. *Universidad Castilla la Mancha*. [En línea] http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr4_VP.pdf.
22. **Guerrero, Luis Antonio.** UML - Diagramas de interacción. *Departamento de Ciencias de la Computación. Universidad de Chile*. [En línea] www.dcc.uchile.cl/~luguerre/cc61j/recursos/interaccion.ppt.
23. **Caballero, Leonardo y Lara, Jesús.** Patrones de Diseño y Orientación a Objetos en PHP5. [En línea] <http://www.slideshare.net/lcaballero/patrones-de-diseo-y-orientacin-a-objetos-en-php5/>.
24. **Ortín, María José, y otros.** El Modelo del Negocio como base del Modelo de Requisitos. *Departamento de Informática y Sistemas. Facultad de Informática. Universidad de Murcia*. [En línea] <http://docencia.udea.edu.co/ingenieria/ArquitecturaSoftware/documentos/Del%20Modelo%20Del%20Negocio%20AI%20Modelo%20De%20Requisitos.pdf>.
25. **Insfrán, Emilio, y otros.** Ingeniería de Requisitos aplicada al modelado conceptual de interfaz de usuario. *Universidad Politécnica de Valencia*. [En línea] <http://www.dsic.upv.es/~pjmolina/papers/IDEAS2001.pdf>.
26. **Molina Moreno, Pedro Juan.** Tesis Doctoral Especificación de Interfaz de Usuario: de los requisitos a la generación automática. *Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia*. [En línea] <http://pjmolina.com/papers/TesisPjmolina.pdf>.
27. **Pérez Jaramillo, Carlos Mario.** Los Indicadores de Gestión. *www.soporteycia.com.co*. [En línea] <http://www.soporteycia.com.co/documentos/SOPLOSINDICADORESDEGESTION.pdf>.
28. **Mérida Mingarro, Angela.** Validación de un sistema de indicadores para medir el desempeño en la empresa de materiales de la construcción de Holguín. *www.monografias.com*. [En línea] <http://www.monografias.com/trabajos15/valoracion/valoracion.shtml>.

ANEXOS

Anexo # 1 Diagrama de



Actividades

Figura # 22 Diagrama de Actividades Consolidar Resultado Mensual de Refinación

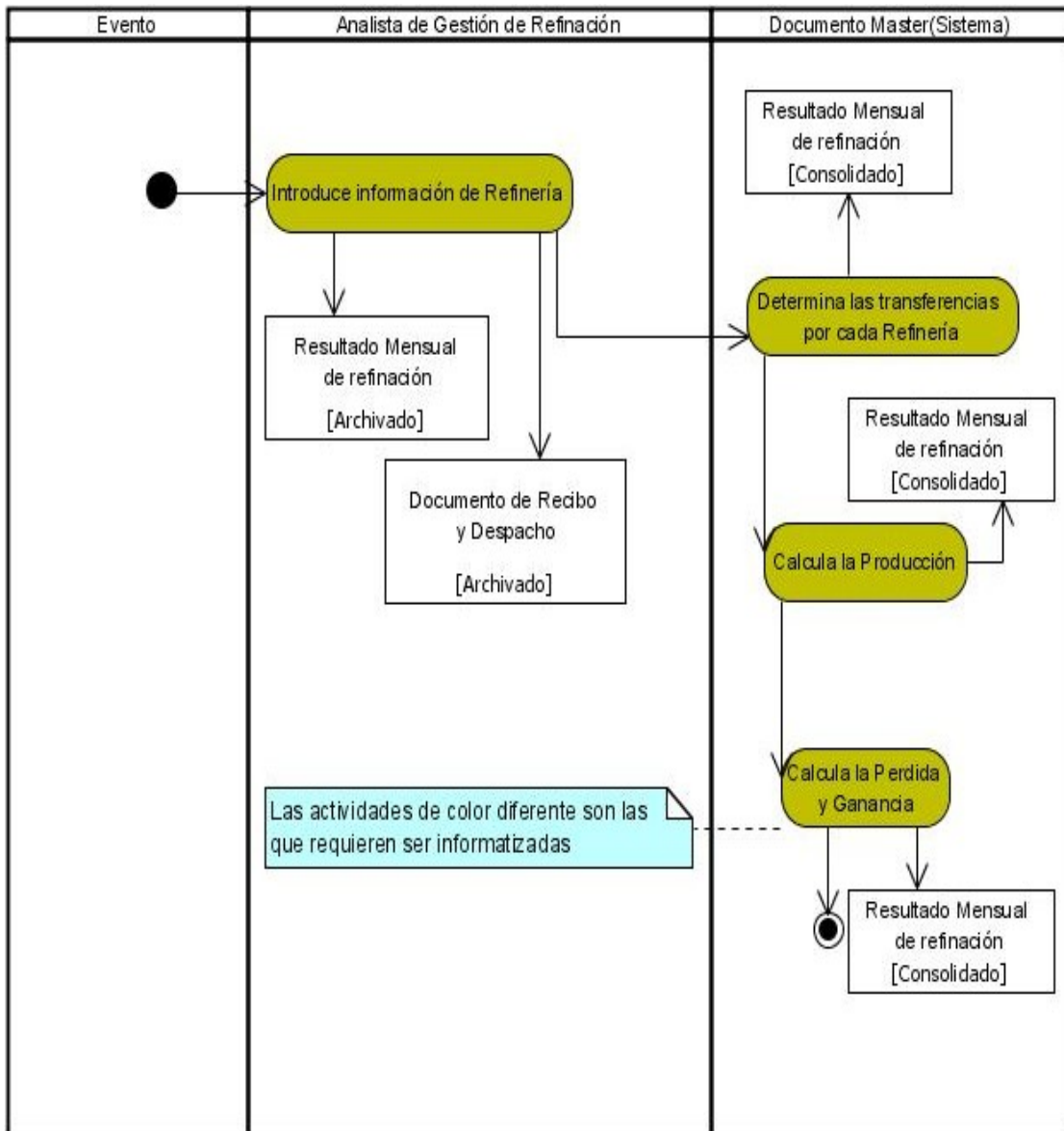


Figura # 23 Diagrama de Actividades Emitir Resultado Mensual de Refinación Consolidado

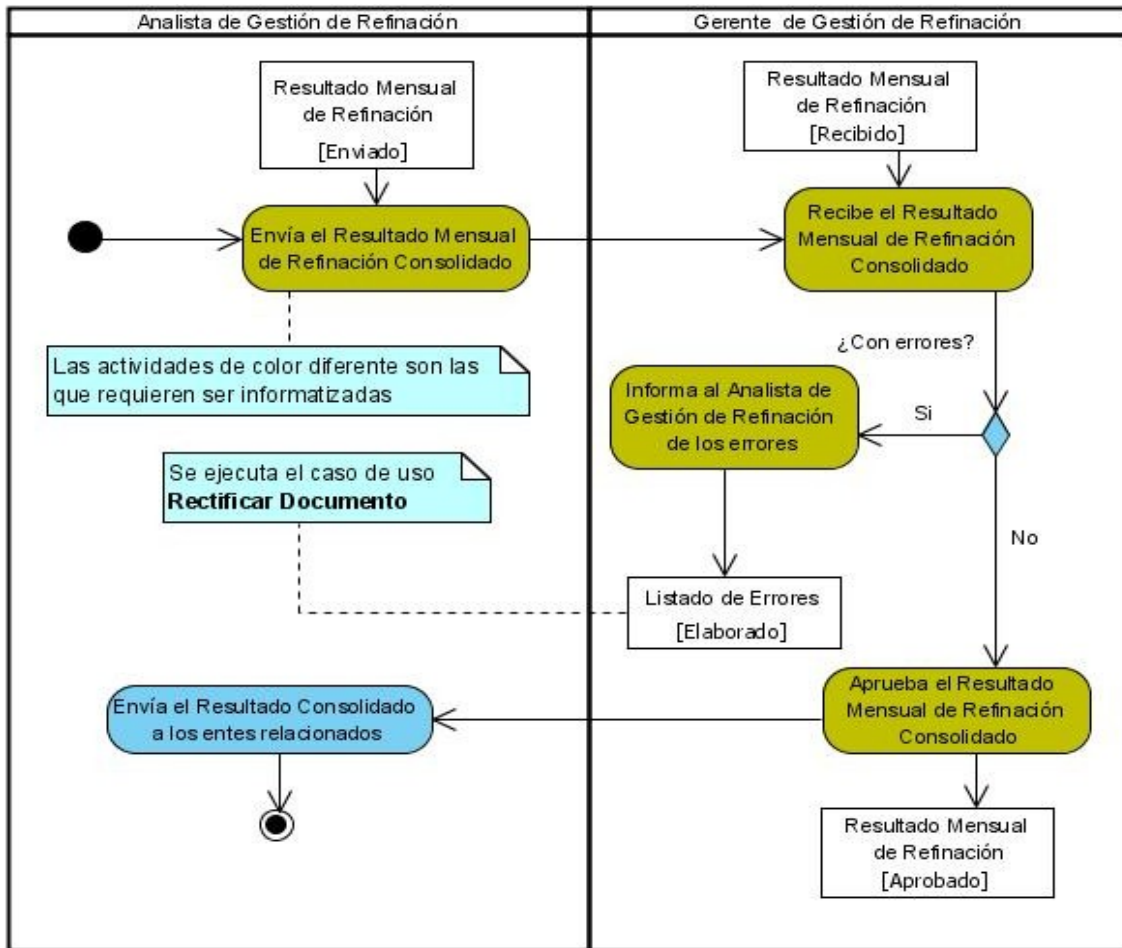


Figura # 24 Diagrama de Actividades Emitir Resultados de Refinación de un Período

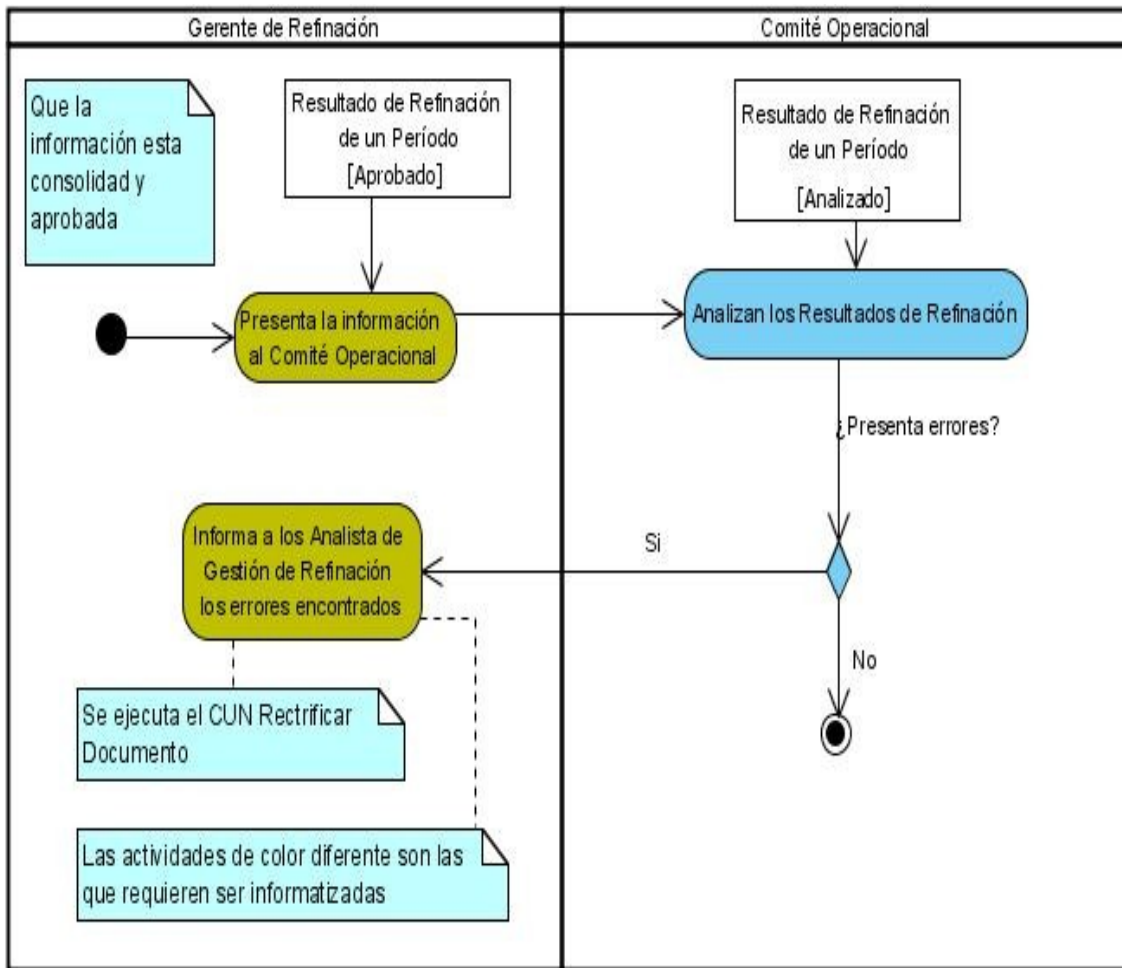


Figura # 25 Diagrama de Actividades Evaluar la variación de Indicadores

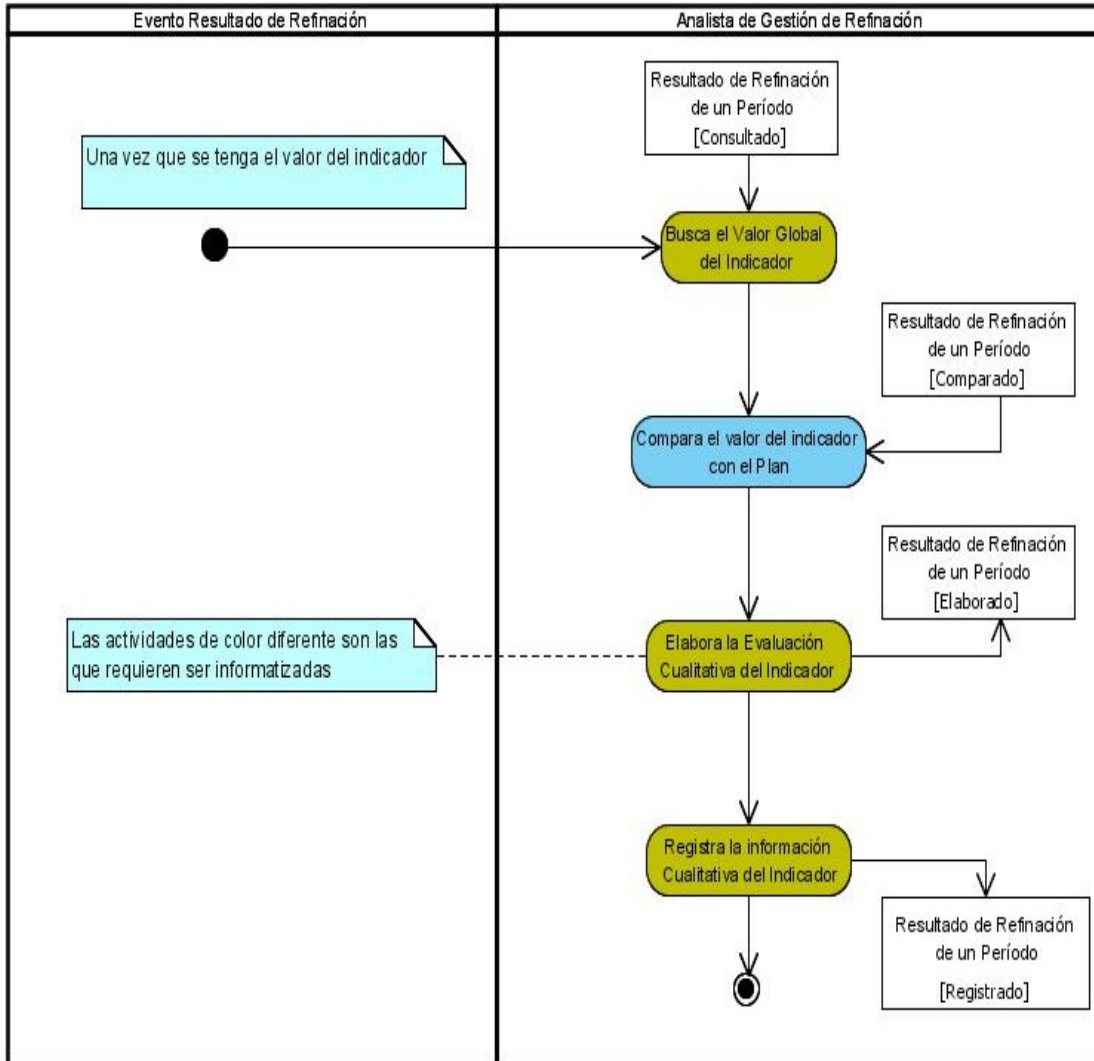


Figura # 26 Diagrama de Actividades Evaluar la variación de Indicadores

Anexo # 2 Diagrama de clases del Análisis



Figura # 27 Diagrama de clases del Análisis del CUS Mostrar Información Fuente del Indicador.

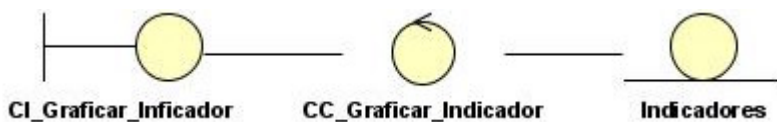


Figura # 28 Diagrama de clases del Análisis del Caso de Uso Graficar Información.

Anexo# 3 Diagramas de Colaboración

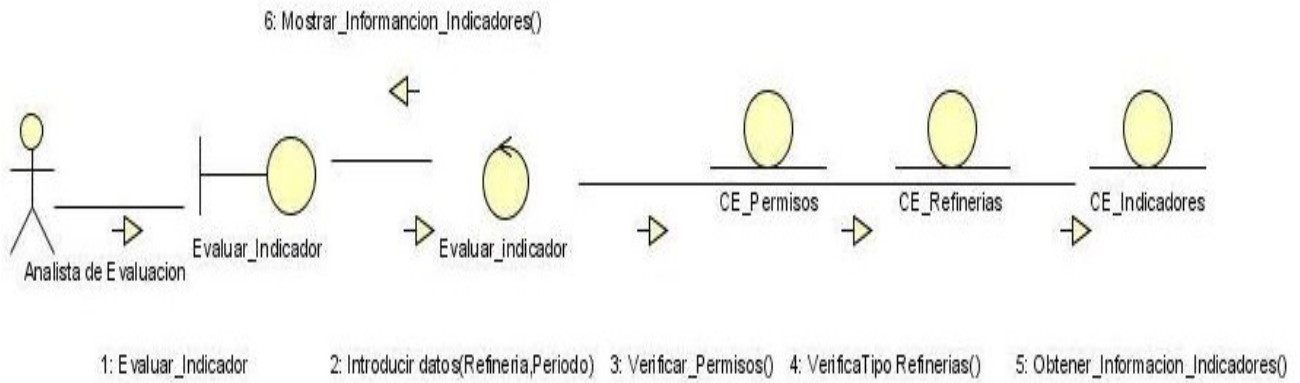


Figura # 29 Diagrama de Colaboración del CUS Evaluar Comportamiento del Indicador.

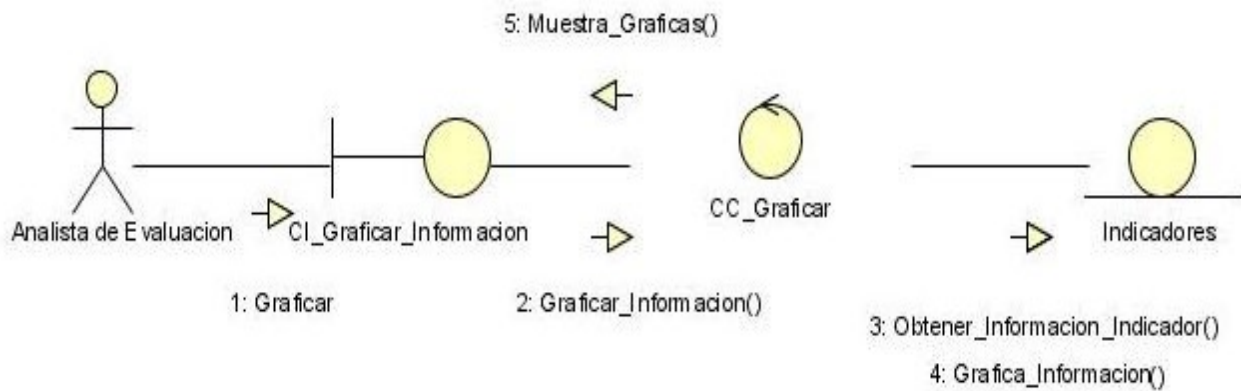


Figura # 30 Diagrama de Colaboración del CUS Graficar Información del Indicador.

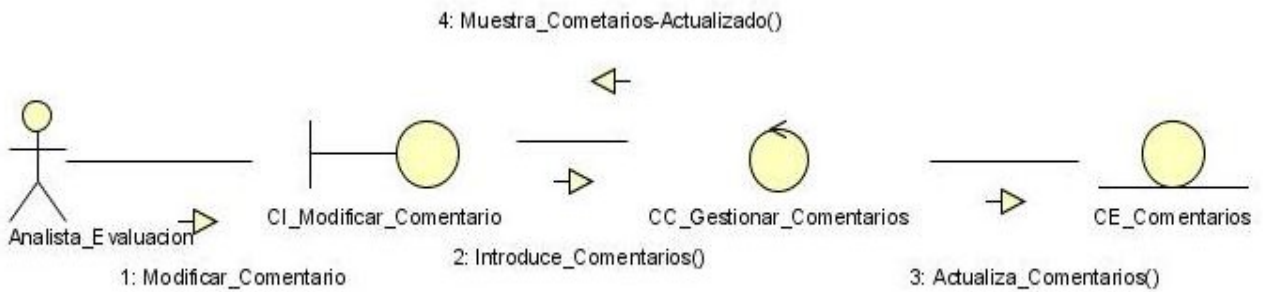


Figura # 31 Diagrama de Colaboración del CUS Evaluar Comportamiento del Indicador. Sección Modificar Comentarios.

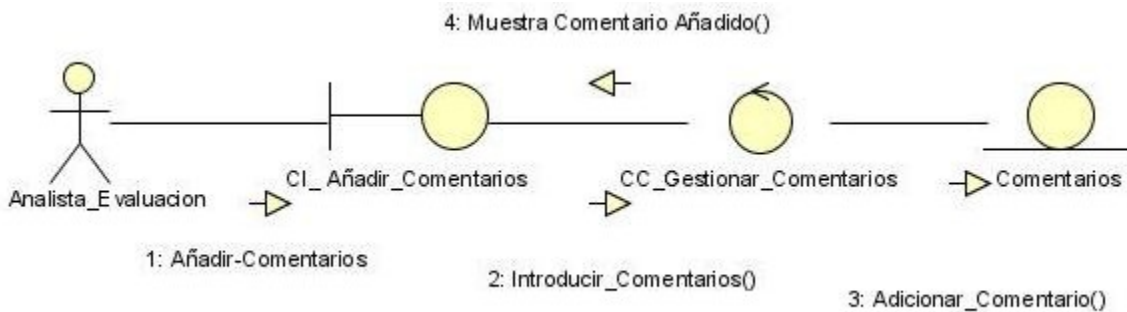


Figura # 32 Diagrama de Colaboración del CUS Evaluar Comportamiento del Indicador. Sección Añadir Comentarios.

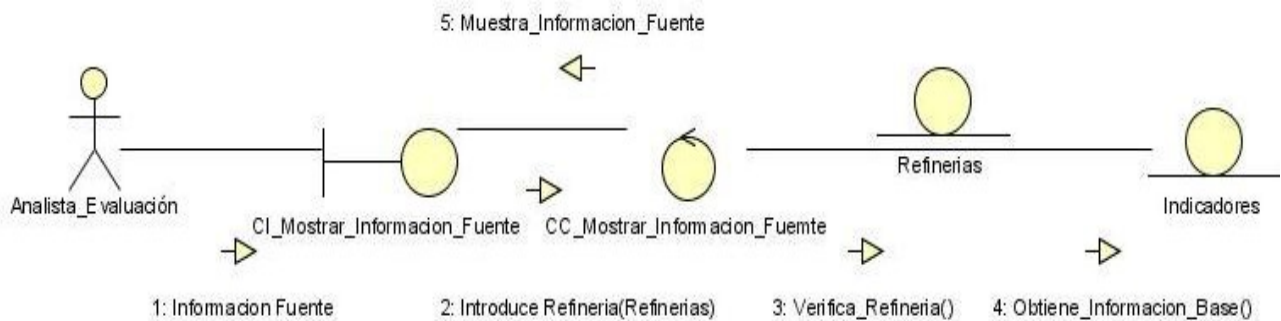


Figura # 33 Diagrama de Colaboración del CUS Mostrar Información Fuente.

Anexo # 4 Diagramas de clases del diseño.

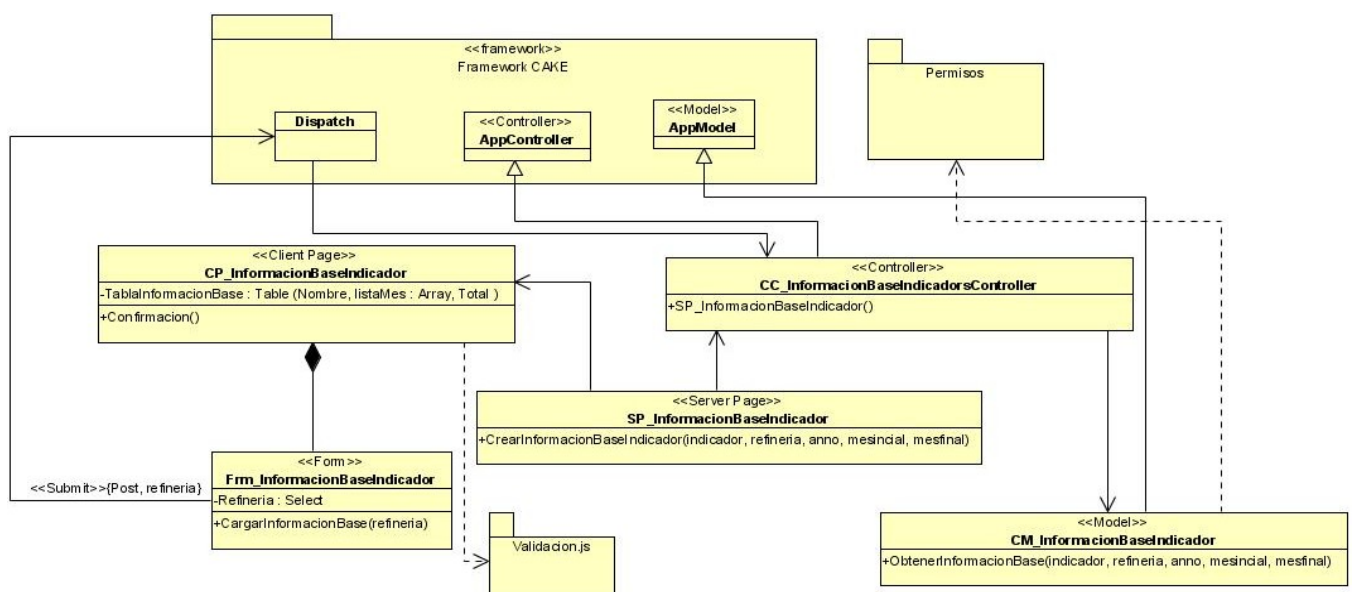


Figura # 34 Diagrama de Clases del Diseño Mostrar Información Fuente.

GLOSARIO

Accidentes: Expresa la cantidad de accidentes ocurridos en un período de tiempo determinado.

Costo de Refinación: Costo del crudo procesado en las refinerías.

Crudo: Mezcla compleja de diferentes tipos de Hidrocarburos y constituye la materia prima fundamental de una refinería.

Exportaciones: Son los volúmenes de productos exportados.

Fuerza Laboral: Es el personal que pertenece a PDVSA ya sea plantilla fija o contratados por un período determinado.

IMAP: Acrónimo inglés de *Internet Message Access Protocol*. Protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet

LDAP Acrónimo inglés de *Lightweight Directory Access Protocol* es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Linux: Sistema operativo derivado de UNIX que, manteniendo casi todas las ventajas que este último ofrece, puede ser ejecutado en computadoras personales. Fue desarrollado originalmente por el estudiante finlandés de informática Linus Torvalds, que publicó su código fuente en 1990, en la forma de código abierto.

NNTP: Acrónimo inglés de *Network News Transfer Protocol*. Protocolo empleado en Internet para transferir artículos en los grupos de noticias.

Paradas de plantas: Numero de días que las plantas están paradas (Programadas o no) en un período dado.

Plan: Es la cifra del plan de volumen de crudo procesado del período (período comprendido desde el inicio del año hasta el mes que se evalúa).

POP3: Acrónimo inglés de *Post Office Protocol* versión 3. Protocolo de Internet utilizado para que clientes locales recuperen su correo electrónico de un servidor remoto a través de Internet.

Red Hat: Compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux.

Refinación: Conjunto de procesos que convierten el crudo en unos productos terminados mediante procesos de separación y/o transformación.

Salidas: Entregas al departamento de ventas. Son los volúmenes de productos entregados al departamento de ventas para el Mercado Nacional.

SNMP: Acrónimo inglés de Simple Network Management Protocol. Es un protocolo que describe cómo se transfiere la información entre dispositivos que informan del estado de la red y los programas que recogen los datos.

Unix: Sistema Operativo multiusuario y multitarea, desarrollado originalmente por Ken Thompson y Dennis Ritchie en los laboratorios Bell en 1969, para su uso en minicomputadoras. Ofrece múltiples ventajas y se considera potente, más portable e independiente de equipos concretos que otros sistemas operativos