

Universidad de las Ciencias Informáticas

FACULTAD # 9



TÍTULO: Diseño del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos.



Autor: Milton Benito León Rodríguez.

Tutor: Ing. David Tavares Cuevas.

Cuidad de la Habana. 2010

“Año 52 de la Revolución”

Dedicatoria

Le dedico este trabajo:

- *A mi mamá por darme la vida y brindarme apoyo incluso en los momentos más difíciles por los que hemos pasado.*
- *A mi papá, por haberme dado educación y carácter.*
- *A Kíke a quien le agradezco eternamente por entrar en nuestras vidas.*
- *A mi hermana a la que extraño mucho, y por proporcionarme los medios para realizar este trabajo con mayor facilidad.*
- *A mi abuelo por no permitir nunca que me desvíe del camino correcto.*
- *A mi novia Yení por llenarme de felicidad dándome siempre ánimo, cariño, confianza y amor en estos casi 4 años.*
- *A toda mi familia por ser una fuente de amor inagotable y enseñarme que la unión es lo más importante que tenemos.*
- *A la gente del aula de las viejas y la nueva, mis hermanos del apartamento y todas las personas maravillosas que he conocido en estos 5 años.*

Agradecimientos

Le agradezco:

- A mi tutor por todo el apoyo que me ha brindado sin importar la hora, le deseo que mucha suerte en su vida y que logre cumplir todos sus grandes sueños.
- A todo el equipo de trabajo del proyecto por ser un grupo maravilloso con los que compartí momentos inolvidables.
- A todos mis profesores por otorgarme el conocimiento para hacer posible este trabajo.
- A todo el tribunal de tesis, gracias a ustedes este trabajo actualmente cuenta con una alta calidad.
- A todas las personas que me han ayudado en el transcurso de estos difíciles 5 años que hemos pasado juntos.
- A mi familia por siempre confiar en mí y nunca dudar de que algún día estaría parado donde ahora me encuentro.
- A mi novia por comprenderme y apoyarme en todos estos años que hemos estado lejos.
- A todos Muchas Gracias.

Declaración de Autoría

Quien suscribe Milton Benito León Rodríguez, hace constar que es el autor del presente trabajo titulado: Diseño del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos, el cual constituye una elaboración personal realizada con la dirección del asesor de dicho trabajo, Ingeniero David Tavares Cuevas y autoriza a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

En tal sentido queda garantizada su originalidad, dejando establecido que aquellos aportes intelectuales de otros autores han sido debidamente referenciados.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Milton Benito León Rodríguez.

David Tavares Cuevas

Resumen

Nos hallamos en una era donde la información crece a ritmo vertiginoso, gracias a las nuevas tecnologías la humanidad dispone de las vías para controlarla. Las empresas tanto pequeñas como las más grandes e importantes hacen grandes esfuerzos por mantener organizados todos sus procesos de negocio ya que es un factor fundamental para el desarrollo de las mismas.

La industria petrolera al ser una de las mayores e importantes mundialmente está inmersa en todo este fenómeno, principalmente porque una equivocación puede traer catástrofes millonarias tanto para la compañía como para el medio ambiente. Además, la toma de decisiones a la hora de perforar un nuevo reservorio se basa en la información previa que se tenga sobre los suelos y la correcta localización del mismo.

Debido al creciente éxito de la industria petrolera cubana se hace necesario la implantación de un sistema de software que controle todo el flujo de información que se crea en los pozos de petróleo en perforación y viaja hacia la dirección de intervención y perforación de pozos.

La presente investigación consiste en modelar los procesos que solamente fueron identificados en la primera versión, además, los nuevos que puedan haber surgido en el período de construcción del sistema. También se propone un análisis de metodologías y herramientas de desarrollo de software a utilizar quedando todo esto firmemente plasmado como resultado de la investigación.

La puesta en marcha del sistema propuesto permitirá cambiar la situación actual de la empresa. Cuenta con una interfaz amigable donde todos los servicios están adaptados a las necesidades específicas de cada trabajador en cada módulo. Todo esto traerá consigo un ahorro en tiempo, mayor usabilidad y que el trabajo sea más cómodo y económico.

Índice

Introducción.	9
Capítulo 1: Fundamentación Teórica	14
1.1 Introducción	14
1.2 Conceptos asociados al dominio del problema	14
1.3 Objeto de Estudio.	17
1.4 Descripción del Proceso Actual.....	17
1.5 Análisis de Otras Soluciones Existentes.	17
1.6 Análisis y Diseño.....	24
1.6.1 Objetivos del Análisis.	24
1.6.2 Objetivos del Diseño.....	24
1.6.3 Análisis y Diseño Orientado a Objetos.	26
1.6.4 Análisis y diseño según RUP.....	27
1.7 Patrones.	28
1.8 Conclusiones Parciales.....	29
Capítulo 2 Tendencias y Tecnologías Actuales a Desarrollar.	30
2.1 Introducción.	30
2.2 Metodologías de Desarrollo de Software	30
2.3 Metodologías Pesadas.....	31
2.4 Metodologías Ágiles.....	33
2.5 Conclusiones Parciales.....	36
2.6 Selección de la Metodología Adecuada.	36
2.7 Caracterizar las herramientas Case y procedimientos más utilizados.	38
2.7.1 Enterprise Architect 7.5	38
2.7.2 Rational Rose.....	38
2.7.3 Visual Paradigm for UML 6.1 Enterprise Edition:	40

2.8	Selección del Lenguaje de Modelación.	42
2.9	Conclusiones Parciales.....	43
Capítulo 3: Presentación de la Solución Propuesta.		44
3.1	Introducción.	44
3.2	Modelo de Negocio.	44
3.2.1	Reglas del Negocio.	44
3.2.2	Actores y Trabajadores del Negocio.	45
3.2.3	Casos de Uso del Negocio.	47
3.2.4	Diagrama de Casos de Uso del Negocio.	49
3.2.5	Descripción textual de los casos de uso del negocio.	49
3.3	Requisitos del software.	54
3.3.1	Requisitos Funcionales.	54
3.3.2	Requisitos No Funcionales.	56
3.4	Modelo de Sistema	57
3.4.1	Actores del Sistema.....	57
3.4.2	Diagrama de Casos de Uso del Sistema.	58
3.4.3	Diagrama de Casos de Uso por Subsistemas.	58
3.4.4	Descripción Textual de los Casos de Uso del Sistema.	58
3.5	Conclusiones Parciales.....	61
Capítulo 4 Diseño del sistema.....		62
4.1	Introducción.	62
4.2	Modelo de Análisis.	62
4.3	Modelo de Diseño.	62
4.3.1	Artefactos del Diseño.	62
4.3.1.1	Diagramas de Clases.....	64
4.3.1.2	Patrones de Diseño.	68
4.4	Conclusiones Parciales.....	69

Conclusiones70

Recomendaciones71

Bibliografía.74

Glosario de Términos76

Introducción.

El siglo XXI ha sido denominado el siglo de la tecnología de la información. Los sistemas informáticos han alcanzado niveles altos de complejidad. Los software de gestión con el transcurso del tiempo lograron lugares de importancia en la economía de los países, ya que la información se ha convertido en un componente de alto valor para el desarrollo de las empresas.

En la carrera tecnológica por la supremacía en el desarrollo de la humanidad, las Tecnologías de la Información y la Comunicación (TIC) se han insertado en todas las esferas de la sociedad y en las industrias más importantes para el desarrollo de los países, como la industria del petróleo. El petróleo es uno de los recursos más importantes para la sostenibilidad de la vida humana como la conocemos en la actualidad, invirtiéndose millones de dólares para la actualización de las tecnologías que se utilizan como sustento en cada uno de los procesos internos que se realizan en la industria petrolera.

Desde hace casi un siglo este es el sustento energético para el desarrollo de la humanidad. Es usado para casi todas las tareas que realizamos cotidianamente, hasta para los principios más básicos como: la elaboración de alimentos, el transporte, la generación de energía, la producción de lubricante, plásticos, vidrio, azufre, entre muchas otras aplicaciones. El uso desmedido de este hidrocarburo puede llevar a la humanidad hacia su auto destrucción.

Se han realizado estudios donde se estima el tiempo de explotación de este recurso natural no renovable al nivel de consumo mundial actual. Estos estudios han dado como resultado que las reservas actuales se agotarán dentro de aproximadamente 45 años. Por tal razón la eficiencia en cada uno de los procesos dentro de la industria es primordial para la explotación máxima de los reservorios. La utilización de las tecnologías y sistemas que automaticen y mejoren estos procesos, posibilitará que podamos seguir utilizando los combustibles fósiles como fuente de energía no renovable.

Alrededor del mundo existen empresas especializadas en la producción de software empleadas únicamente a la producción de sistemas para la industria del petróleo, los cuales automatizan los procesos fundamentales que se realizan. Entre estas empresas se encuentra TECNA, es una compañía que se dedica a realizar operaciones, así como dar mantenimiento a equipos en plantas petroleras. (1)

RCLAB S.R.L Inc. Se convirtió también en una empresa líder en la venta de productos principalmente de medición de indicadores en cuanto a la calidad del petróleo ofertando equipos que automatizan procesos como el cálculo del índice de saturación, entre muchos otros. (2)

Schlumberger fundada en 1926 actualmente es la principal y mayor de las empresas proveedoras de productos y servicios en el área del petróleo. Hace operaciones en más de 80 países. Presenta una vasta experiencia en el campo y está avalada por sus altos resultados en compañías de gas y petroleras alrededor del mundo. Schlumberger trabaja al mismo tiempo en los campos de investigación, desarrollo y creación de nuevos productos, soluciones y servicios pensados para un mejor aprovechamiento de recursos de una forma eficiente y ecológica. (3)

Otras empresas como la empresa de software SAP que implementa un software de gestión empresarial adecuado que ayuda a gestionar las dificultades operativas, a integrar la información, los procesos y proporcionar potentes herramientas para la toma de decisiones que mejoran el rendimiento, reducen los costos y facilitan la colaboración y la comunicación en la empresa. (4)

En Cuba la unión CUBAPETROLEO (CUPET), es la empresa que se encarga de toda la actividad petrolera desde la exploración hasta la comercialización. A partir del incremento de la extracción de crudos nacionales, CUPET ha comenzado una fase de automatización de sus procesos en cada una de sus empresas y unidades, imprescindible para mejorar los resultados cualitativos y cuantitativos en cada uno de ellos.

En la Universidad de las Ciencias Informáticas (UCI), se encuentra en desarrollo un proyecto (Sistema de Manejo Integral de Perforación de Pozos - SIPP) que pretende informatizar todos los procesos y actividades que se realizan en la Dirección de Intervención y Perforación de Pozos (DIPP), radicada en la Empresa de Producción y Extracción de Petróleo del Centro (EPEPC). Este proyecto tiene como meta automatizar toda la información generada por los procesos que se realizan en los pozos en perforación, de manera tal que se eliminen las ambigüedades en la información y se ahorre tiempo en el proceso. De los reportes que se generan diariamente en los pozos en perforación, depende la toma de decisiones de esta entidad (DIPP).

El mismo se encuentra a punto de culminar su primer ciclo de vida, y debido a la existencia de un próximo ciclo, se identifica como **situación problemática** la necesidad de modelar los

procesos que fueron identificados y no fueron modelados, además de los nuevos procesos que puedan haber surgido en el período de construcción del sistema. Continuar limando los problemas de gestión y control de la información, que existe entre los pozos en perforación y la DIPP.

Ya existe una primera versión del diseño de este sistema y es el punto de partida para lograr la modelación del segundo ciclo de vida del proyecto. Se desea agilizar el proceso, para que el siguiente ciclo de desarrollo se realice en menor tiempo y optimizar la fecha de entrega al cliente, sin sacrificar la calidad del sistema.

Partiendo de la situación expuesta con anterioridad se ha definido como **problema a resolver** la inexistencia de artefactos ingenieriles para realizar el Segundo Ciclo de Vida del Sistema de Manejo Integral de Perforación de Pozos. Para darle solución a este problema se define como **objetivo general** diseñar el Segundo Ciclo de Vida del Sistema de Manejo Integral de Perforación de Pozos, definiendo los siguientes **objetivos específicos**:

- Identificar procesos de negocio y funcionalidades del sistema.
- Diseñar los Casos de Uso Críticos del Ciclo de Desarrollo.

La investigación estará centrada en los procesos de negocio en la gestión de información en los pozos de petróleo en perforación y en la Dirección de Intervención y Perforación de Pozos, así como el análisis del primer ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos, lo cual constituye el **objeto de estudio**. La investigación está enmarcada en la automatización de los flujos de información en pozos de petróleo en perforación y en la Dirección de Intervención de Pozos en Perforación de la empresa Cuba Petróleo (CUPET), definiéndose como **campo de acción**.

Se **defiende la idea** de que si se logran identificar correctamente los procesos de negocio y las funcionalidades, se podrá obtener un Modelo de Diseño del segundo ciclo de desarrollo del Sistema de Manejo Integral de Perforación de Pozos.

Con el fin de lograr los objetivos propuestos se han planificado las siguientes **tareas de la investigación**:

1. Describir el Estado del Arte.

2. Seleccionar la metodología de desarrollo de software a utilizar, Patrones y Estereotipos UML para el diseño efectivo del sistema.
3. Identificar los procesos de negocio y las funcionalidades en cada uno de los pozos de petróleo en perforación y la DIPP.
4. Estimar la complejidad del sistema.
5. Diseñar los casos de uso definidos como críticos del Ciclo de Desarrollo.

Para dar cumplimiento a estas tareas se hace necesario aplicar métodos de investigación científica que faciliten el proceso de recogida de información. Entre los usados se pueden encontrar los **métodos teóricos** y **métodos empíricos**. Dentro de los métodos teóricos se encuentra el método **analítico sintético** y el **histórico lógico**.

El método analítico sintético se usa para modelar todo el funcionamiento de los pozos de petróleo en perforación. Permite extraer las características más importantes para lograr un entendimiento completo de todos los procesos que se realizan en los mismos. Este conocimiento que brinda es usado a la hora de elaborar el modelo de negocio, análisis y diseño del sistema.

El método histórico lógico es usado para confeccionar un estado del arte sobre las soluciones que den respuesta al problema en cuestión. Su principal objetivo es la recopilación de información referente a las principales empresas del área, así como las herramientas utilizadas por estas. Se registran las investigaciones que precedieron a esta para proveer una base científica que apoye todo el conocimiento que se ha obtenido.

Los métodos empíricos no son menos importantes y entre estos se encuentran las **entrevistas** y la **observación**. La realización de **entrevistas** es uno de los pasos más importantes en el desarrollo de una investigación, ya que permite conocer interioridades de cada proceso que en un análisis general no se identifican. El principal objetivo de este método es extraer el conocimiento sobre diferentes procesos, pero a través del individuo que es el que está a diario en contacto con los mismos. Ayuda a esclarecer detalles específicos del funcionamiento de los procesos que son necesarios para una correcta modelación e implementación de los mismos. La **observación** tiene como meta obtener un nivel de realidad sobre los procesos que conforman el objeto de estudio, sin la cual se corre el riesgo de no entender a cabalidad los procesos desarrollados en el mismo. Es por excelencia el método empírico indispensable para comprender el funcionamiento y la interrelación entre los componentes. Se realizó en visitas a

la DIPP y al Pozo, teniendo en cuenta la realización de los otros métodos mencionados mientras se compartía con sus trabajadores.

Para un mayor entendimiento de este trabajo se propone la distribución del mismo de la siguiente manera:

- **Capítulo I:** Se describe un estado del arte. Se buscan propuestas a nivel internacional y nacional que den solución al negocio en cuestión. Se establece una comparación entre las soluciones encontradas. Se describe el proceso actual y el objeto de estudio como consecuencia del planteamiento del problema.
- **Capítulo II:** Se selecciona la metodología de desarrollo de software a utilizar. Se identifican metodologías de desarrollo de software, además, se caracteriza cada una de las metodologías identificadas y se comparan las metodologías identificadas. Se justifica el por qué la utilización de la metodología utilizada.
- **Capítulo III:** Se identifican los procesos de negocio y funcionalidades, de los pozos en perforación. Se identifican procesos de negocio en la DIPP. Se hace un análisis completo del sistema.
- **Capítulo IV:** Se abordan aspectos referentes al uso de patrones. Se analizan los patrones de diseño a utilizar y estereotipos UML para la web. Se realiza el diseño de los casos de uso críticos, así como el diagrama de clases correspondiente.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción

En este capítulo se tiene como objetivo principal reflejar por dónde se encuentra actualmente el desarrollo mundial de sistemas informáticos para la extracción de información en los pozos de petróleo, así como otros que realicen actividades semejantes como automatizar procesos comerciales dentro de una empresa. De esta forma se logra un estado del arte en la presente investigación, así como una visión general del negocio en cuestión. Se dedicará un espacio también a abordar aspectos teóricos referentes al análisis y diseño y el uso de patrones.

1.2 Conceptos asociados al dominio del problema

- **DIPP:** (Dirección de Intervención y Perforación de Pozos), única entidad en Cuba responsable de dirigir el proceso de perforación e intervención de todos los pozos de petróleo que se encuentran en ese estado. Recibe diariamente los reportes de los mismos sobre el avance de la perforación o si hubo alguna urgencia. Informa al CEIMPET semanalmente sobre las características principales de cada pozo.
- **Perforación:** Proceso más importante dentro de la Industria del Petróleo ya que proporciona la certeza de la existencia del hidrocarburo. Aunque existe una gran interdependencia entre los procesos, ya que del éxito de uno de ellos depende el éxito del siguiente. La importancia de este proceso se debe principalmente a que aproximadamente 1 de cada 5 pozos en perforación llega al estado de extracción, debido a que no siempre los cálculos son correctos y los métodos para localizar el petróleo no son exactos. Mantener un pozo en perforación cuesta varios miles de dólares diarios.
- **EPEPC:** Empresa de Producción y Extracción de Petróleo del Centro. Entidad con la finalidad principal de regular y controlar la explotación de los yacimientos del centro del país. Es la empresa más grande de este tipo en Cuba a la cual se le subordina la dirección de intervención y perforación de pozos.
- **CUPET:** Es la empresa que se encarga de toda la actividad petrolera desde la exploración hasta la refinación, así como de satisfacer las necesidades del mercado nacional de hidrocarburos, a partir del incremento de la extracción de crudos nacionales.

- **Procesos de la Industria del Petróleo:** Exploración, Perforación, Producción, Refinación y Comercialización. Gracias a estos procesos es posible obtener y comercializar el hidrocarburo en Cuba. Existen varios métodos para realizar la exploración, actualmente la más usada es la exploración magnética que se basa en el envío y recepción de ondas a través de la superficie terrestre. La perforación más usada en Cuba es la direccional que se trata de perforar el caño hasta aproximadamente 200 metros y darle un ángulo para buscar el reservorio horizontalmente. Se usa de tierra al mar. La producción fundamental es la surgente o por bombas de agua. La surgente es en la que el petróleo sale sin ayuda. La otra se realiza inyectando agua en el reservorio y como el petróleo es menos denso sale a la superficie. La refinación se realiza en plantas como la Níco López, se trata de separar el petróleo de otros componentes como el azufre, ácidos y otros minerales. La comercialización la realiza CUPET en más de 200 puntos de venta de gasolina y diesel distribuidos en todo el país, así como en la generación de electricidad donde se usa la mayoría del petróleo extraído en Cuba.
- **Reporte operativo de perforación:** Es la información que recibe el personal de la DIPP de los distintos pozos en perforación. En esta parte se informa todo el avance diario del pozo, el estado en que se encuentra el mismo, la barrena que se está usando, la profundidad actual, si ha ocurrido algún inconveniente que atrase la perforación, así como todas las actividades importantes en la perforación.
- **Reporte operativo de intervención:** Se recibe por las tardes en la DIPP y se envía al día siguiente. Cuando un pozo termina su límite de perforación y no ha llegado al reservorio u ocurre algún accidente que no permita continuar con la perforación, este pasa al estado de intervención, donde se realizan diferentes maniobras con otras herramientas para tratar de sacar el mayor provecho del pozo. En esta parte también se reflejan todas las actividades diarias que se realizan en el pozo.
- **Resumen por combustible:** Es un reporte que indica el consumo diario en litros de combustible en cada uno de los pozos, así como el pronóstico para cada uno.
- **Resumen de las principales necesidades:** Las necesidades las piden por teléfono por la tarde sobre las 2:00 p.m. y llenan la planilla del Pedido de Logística. Donde se recogen los Servicios de Necesidades. Es el reporte que envían a las diferentes empresas (EMPERCAP, EMSERPET, EMPET, etc.), para satisfacer estas necesidades. Se lo envían al despacho central.

- **Reporte Diario de Consumo (En Litro) de Perforación:** Se envía en el día sobre las 10:00 a.m. Principalmente este es importante para el Logístico, pero se lo envían a todos los sectores mencionados anteriormente.
- **Informe al Semanal MINBAS:** Envían semanal a la Ministro(a) un resumen de la semana. Es un resumen de todos los pozos de petróleos tanto nacionales como de empresas extranjeras donde se refleja el estado en que se encuentra cada uno de estos.
- **Exploración-Producción:** Área de Dirección de CUPET, la cual engloba tres procesos dentro de la industria del petróleo: la exploración, la perforación y la producción.
- **Reporte de Encamisado:** El encamisado no es más que el proceso donde se forra el hoyo perforado con una serie de tuberías de acero que cada vez son más estrechas según la perforación se acerca al reservorio. El objetivo del encamisado es sostener las paredes del pozo evitando derrumbes.
- **Camisa de Superficie:** Es la primera con la que se forra el pozo, la cual debe ser la más ancha según el pozo a perforar.
- **Camisa de Liner:** Es la última de las camisas introducidas en el pozo, después de la cual se encuentra el reservorio.
- **Reporte de Evaluación Técnica:** Este reporte se envía a la DIPP por intervalos de encamisado. Contiene la cantidad de metros que se ha perforado así como todas las labores que se han realizado y el tiempo que se consumió en cada una.
- **Reporte de Medición del Casing:** No es más que la medición de todas las camisas que llegan al pozo y el orden en el que se colocan en la herramienta.
- **Reporte de Tally:** Después de concluir el proceso de encamisado este reporte indica la cantidad real de camisas bajadas más el zapato de perforación y la profundidad del pozo.
- **Zapato de Perforación:** Es una herramienta que se utiliza para cerrar la entrada de los fluidos del pozo hacia el interior de las camisas, se coloca en la última de estas en cada intervalo de encamisado.
- **Reporte de Lot:** Consiste en realizar una prueba donde se insertan fluidos a la formación perforada para detectar cuanta presión esta es capaz de aguantar.

1.3 Objeto de Estudio.

El objeto de estudio como consecuencia del planteamiento del problema delimita la parte de la realidad que es necesario estudiar para solucionar el mismo. Son los procesos de negocio en la gestión de información de los pozos de petróleo en perforación y en la Dirección de Intervención y Perforación de Pozos. En los pozos de petróleo en perforación se generan una serie de reportes diarios y semanales que son enviados a la dirección de intervención y perforación de pozos. Los mismos se transmiten por teléfono o por correo electrónico, lo que puede ocasionar pérdida de la información y errores en la misma. También se puede eliminar información que se repite dentro de los mismos pozos ya que en muchas ocasiones dos trabajadores trabajan sobre el mismo parte y no se reutiliza la información, por lo que se duplica el esfuerzo. El análisis del primer ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos es el punto de partida de esta investigación, el cual permite que se puedan mejorar y optimizar procesos que ya no se realicen de la misma forma, además de optimizar los mismos ya identificados.

1.4 Descripción del Proceso Actual.

En nuestro país la DIPP (Dirección de Intervención de Pozos en Perforación), maneja toda la información proveniente de los pozos en perforación por vía telefónica y correos electrónicos. Esta información es fundamental para poder tomar decisiones importantes sobre el futuro de un pozo en perforación o intervención. Entre los directivos de la DIPP la información se maneja vía correo electrónico y sin un diseño estipulado por lo que esta es arreglada reescrita muchas veces. Esto provoca que se duplique el esfuerzo y es propicio para que se introduzcan errores humanos en los datos. Los supervisores reciben la información de los reportes por vía telefónica o solo se almacenan en hojas de papel, además, estos no siempre se almacenan cuando un pozo termina su estado de perforación. Para ello es necesario la creación de un sistema que se encargue de manejar y almacenar toda la información resultado de los procesos que se realizan en los pozos, así como todos los reportes diarios que se envían a la DIPP. Dentro de la DIPP estandarizar la manera en que estos partes son vistos y definir la información que se puede modificar en un estándar generalizado para todo el departamento.

1.5 Análisis de Otras Soluciones Existentes.

Existen una gran cantidad de software o compañías que pueden aportar soluciones o respuestas parciales a alguno de los procesos que se pretenden automatizar. La mayoría son programas que vienen ya prediseñados para resolver diferentes problemas en la extracción y refinación del petróleo, tales como:

- **NeuraLog Inc.** La línea de productos de NeuraLog Inc. le da al usuario un flujo de trabajo completamente integrado a las funciones de las aplicaciones que se encuentran en su empresa, sea usando la conectividad directa a estaciones de trabajo o bases de datos de Landmark, Schlumberger/GeoQuest o usando formatos propios y comunes de aplicaciones más usadas en la Industria petrolera en más 70 países en el mundo. Geólogos, Geofísicos, Ingenieros y todo aquel envuelto de una manera u otra en el proceso de manejo de datos y procesamientos de información de la industria petrolera para las áreas de exploración, producción y explotación necesitan las aplicaciones de NeuraLog Inc para la captura de Imágenes, digitalización y rápida interpretación con la mejor calidad existente en el mercado. La integración de datos a aplicaciones de Schlumberger, GeoQuest, Landmark, Geographix, AutoCad, ArcView y muchas más a través de formatos especializados o de forma directa a ellas ayuda a integrar los datos provenientes de imágenes en formato tiff existentes en los archivos de cualquier empresa a estaciones de trabajo para su pronta evaluación y estudio. (5)
- **Petrokem Loggin Services (PLS).** Es una compañía ecuatoriana con más de 13 años de experiencia en el sector del petróleo. Ha trabajado para varias operadoras, entre ellas: el Consorcio Petrolero B14, Pacifpetrol, Halliburton-Petrobras, Halliburton-Repsol, Kerr MCGee y otras. A todas estas compañías operadoras PLS les ha proporcionado el servicio de Control Litológico con una muy alta calidad, sistemas confiables para la adquisición de datos y un versátil programa capaz de ejercer las tareas de control cálculo y graficación de todos los requerimientos del cliente en materia de control litológico. (6)
- **Schlumberger GeoQuest:** Es una unidad operativa que proporciona software de gestión de la información, infraestructura de TI y servicios. A través de sus tecnologías y servicios a compañías de petróleo y gas pueden mejorar el rendimiento empresarial, reducir el riesgo de exploración y desarrollo además de aprovechar el potencial del campo petrolero digital. (3)

Como también existen otras empresas que se dedican al diseño de software privativo que darían una solución factible. Solo que habría que pagar una licencia para la instalación y uso del producto. A continuación se encuentra un ejemplo de empresas que actualmente están tratando de introducirse en el mercado cubano:

- **Well Wizard:** Es un sistema de muestreo de todos los factores que son fundamentales en la perforación de pozos como aire comprimido, temperatura de lodo, los cuales son monitoreados por un software que está actualmente instalado en todos los pozos en perforación de Cuba. Este software le cuesta a cada pozo de petróleo en perforación alrededor de 12 000 dólares diarios.

Ventajas:

- ✓ EPA-aceptadas bajo flujo de muestreo de precisión.
 - ✓ Más de 75 000 bombas de Well Wizard asistentes de la cámara de aire están en uso más que todas las otras marcas y tipos de muestras de agua y tierra combinados.
 - ✓ Modelos para cada pozo, bajo rendimiento, corta columna de agua, a profundidades de 1.000 pies (304.8m), para envolver ID hasta 1,25 (31,75 mm).
 - ✓ Confiabilidad probada desde 1982, con el primer estándar de la industria de 10 años de garantía.
 - ✓ Exclusiva formulación de la Cámara de aire de PTFE clasificado para muchos años de vida, mayor flexibilidad de los materiales de la de la misma. (8)
- **Well Logger. Creador de Registros de Pozos.**

Well Logger permite crear informes de perforación de suelos y diagramas de construcción de pozos. Los ingenieros de proyecto y los geólogos pueden al usar Well Logger en un ordenador portátil en su lugar de trabajo completar rápidamente la documentación necesaria. Este proceso ocurre normalmente durante las paradas de los trabajos de perforación. Well Logger ofrece una sencilla, aunque robusta interfaz de usuario que ofrece presentaciones personalizables, patrones definidos por el usuario, escala ajustable y vista previa de la impresión. Well Logger tiene una interface de hoja de cálculo fácil de usar, con cajas de entrada que simplifican la entrada de datos para cada perforación. La información de entrada incluye litología de la perforación, muestras tomadas, construcción del pozo o detalles anexos de la perforación, y la información general acerca del proyecto y la perforación. Well Logger puede crear sus informes de perforación y pozo en la mitad del tiempo que los programas tradicionales CAD. (9)

- **Software WELLSIGHT.**

El software WellSight son un conjunto de software utilizados para captación, informes, monitoreo, seguimiento, reconciliación y exportación de datos. Originalmente este software era para el desarrollo petrolífero en la Cuenca occidental de Canadá pero luego de ver sus beneficios se ha extendido su uso a diferentes usuarios del mundo. Dentro de sus diferentes funciones se hará alusión a la de exportación de datos. (10)

La Exportación de datos del software WELLSIGHT brinda cuatro opciones para extraer datos de la base de datos de WELLSIGHT. Estas opciones representan normas de la industria o formatos específicos de clientes e incluyen lo siguiente:

- Lenguaje para marcar normas para transferir información del sitio del pozo WITSML, por sus siglas en inglés.
- Gestión de datos de Landmark.
- Gestión de datos DISWIN de Chevron.
- Gestión de datos DBR de Statoil.

Características:

El WITSML es una iniciativa de la industria petrolera que establece nuevas normas para la transferencia de información sobre perforación lo cual ha sido de gran ayuda en el proceso de control y supervisión de los pozos. Archivo de exportación WITSML Export, versión 1.3.1 (*.xml). Una amplia gama de objetos con datos pueden transferirse a través de WITSML:

- Pozos
- Datos del pozo
- Informes de fluidos
- Inventario de fluidos
- Volúmenes en tanques
- Volúmenes de lodo
- Bombas de lodo
- Zarandas y mallas de zaranda
- Descargas
- Descripción de tratamientos

- Personal

La Gestión de datos DIMS de Landmark y la exportación de aplicaciones Open Wells® incluye:

- Propiedades de fluidos de perforación
- Inventario de lodos

La exportación de la aplicación de Gestión de datos DISWIN de Chevron incluye:

- Propiedades de fluidos
- Inventario
- Volúmenes
- Tratamientos
- Costo

La aplicación de Gestión de datos DBR de Statoil posibilita la exportación de volúmenes de fluido de perforación a un archivo Excel específicamente formateado de modo que los datos puedan importarse en DBR. La exportación incluye:

- Volúmenes de fluido de perforación
- Propiedades

Ventajas

- Minimización de tiempo de transferencia de datos con aplicaciones del cliente.
- Mejor calidad de datos.
- Cuatro opciones diferentes para exportación de datos.
- Disponible a todos los usuarios de software WELLSIGHT.
- (Sitio Web/Vínculos para descargas)

WellSight Systems Inc. es una empresa de software Canadiense. (10).

- **Pelotón.**

Pelotón es una empresa líder en el desarrollo de sistemas de información de Construcción y Operaciones de pozos. La oficina principal de Pelotón está ubicada en

Calgary, Alberta. Las operaciones en los Estados Unidos se manejan desde Houston, Texas. Desde Aberdeen, Escocia se apoyan las operaciones europeas. Los productos de Pelotón incluyen WellView®, SiteView®, RigView® y el Generador de Gráficos de Estados Mecánicos.

- ✓ Su producto insignia, WellView, es un sistema completo de administración de datos de pozos, que permite a las empresas de petróleo y gas, administrar sus datos: desde la solicitud para perforar hasta el abandono de la localización. Más de 180 empresas petroleras en el mundo usan esta aplicación de Pelotón.
- ✓ Hasta la fecha, Pelotón se ha auto-financiado y ha administrado su crecimiento cuidadosamente mientras mantiene los estándares de calidad que son fundamentales para el futuro. Han crecido de una pequeña operación de dos personas en 1995 a sesenta personas con oficinas en Aberdeen, Ámsterdam, Calgary, Houston, Yakarta, Moscú y Dubái. (11).

- **Sistema Manejo Integral de Perforación de Pozos.**

La industria del petróleo en nuestro país, como en muchos otros, se dividen en tres grandes procesos: 1- Exploración-Producción, 2- Refinación y 3- Comercialización. Las entidades especializadas en Exploración-Producción, como su nombre lo indica, se encargan de la exploración y desarrollo de los campos, la perforación de nuevos pozos, así como la reparación de los que ya se encuentran en producción (Intervención de Pozos). El sistema a diseñar involucra al CEIPET(Centro de Investigaciones del Petróleo), los pozos en perforación y la DIPPP(Dirección de Intervención y Perforación de Pozos), entidad única en el país que dirige la perforación e intervención de pozos de petróleo. Esta radica en La Empresa de Producción y Extracción de Petróleo del CENTRO (EPEPC). Controla todas las operaciones y demás actividades que se realizan en los pozos en perforación e intervención. Este sería el punto de del presente trabajo de diploma, se trata del diseño de un sistema que tiene como principal objetivo manejar todo el flujo de información que se genera en los pozos de petróleo en perforación de manera tal que se eliminen las ambigüedades en la información y se ahorre tiempo en el proceso, ya que del éxito de la realización de estos reportes y/o partes depende mucho la toma de decisiones de esta entidad. Además, este sistema propicia un mejor control de la información generada, así como un acceso rápido y simple a la información ya que el proceso de gestión de información se realizaba manualmente. Esto trae consigo redundancia en los datos, así como errores y des actualización en la información. No existe un orden y

control para el almacenamiento de la información. Actualmente se encuentran ya modelados 24 casos de uso del sistema, los cuales ya han sido implementados y están próximos a pasar a su fase de prueba. Este sistema actualmente se encuentra incompleto ya que existen muchos procesos de negocio que no habían sido identificados y son fundamentales para el correcto funcionamiento del producto en general. Por lo que se hace necesaria la existencia de una segunda versión que tendrá como principal objetivo identificar estos procesos de negocio y llevar a cabo la modelación de los mismos; así como optimizar el negocio y la modelación ya existente.

- Conclusiones Parciales.

Casi todos los sistemas expuestos aquí son los de más alto prestigio en la rama de la informática vinculada a la industria petrolera. Pero tienen inconvenientes que hacen que el uso del producto que se pretende ofrecer después de la implementación de la modelación propuesta sea fundamental para Cuba y competitivo a nivel internacional; estos inconvenientes son:

- Para el Mundo.
 - ✓ Todos son software sobre la plataforma de escritorio.
 - ✓ Estos sistemas pertenecen a empresas, canadienses y de EE.UU por lo que todos estos están bajo licencias propietarias y hay que pagar un costo elevado. Además de que algunos serian imposibles de comprar debido a la política restrictiva de EE.UU implantada a nuestro país a través del Bloqueo Económico y Comercial, muchas veces más elevado de lo normal.
- Para Cuba.
 - ✓ Gracias al sistema socialista vigente en Cuba, este producto sería completamente gratuito para CUPET; así como todas sus actualizaciones.
 - ✓ Debido a que todos los procesos de negocio fueron levantados en un pozo de petróleo localizado en Matanzas, es el software que más se adapta a la industria petrolera nacional y cumple con todas sus especificaciones al 100%.
 - ✓ El negocio de por sí ya es bastante complicado y para lograr sistemas que sean comercializables a nivel global, se debe ganar un alto nivel de capacitación referente a la industria y alta confiabilidad en los clientes nacionales.

1.6 Análisis y Diseño.

1.6.1 Objetivos del Análisis.

- **Identificación de Necesidades.**

Es el primer paso del análisis del sistema, en este proceso el Analista se reúne con el cliente y/o usuario (un representante institucional, departamental o cliente particular), e identifican las metas globales. Se analizan las perspectivas del cliente, sus necesidades y requerimientos, sobre la planificación temporal y presupuestal, líneas de mercadeo y otros puntos que puedan ayudar a la identificación y desarrollo del proyecto.

Algunos autores suelen llamar a esta parte Análisis de Requisitos y lo dividen en cinco partes:

- Reconocimiento del problema.
- Evaluación y Síntesis.
- Modelado.
- Especificación.
- Revisión

Antes de su reunión con el analista, el cliente prepara un documento conceptual del proyecto. Aunque es recomendable que este se elabore durante la comunicación Cliente – analista, ya que de hacerlo el cliente solo, de todas maneras tendría que ser modificado, durante la identificación de las necesidades. **(11)**

1.6.2 Objetivos del Diseño.

- **Definir el producto final.**

Definir el producto final antes de empezar a programar es algo tan obvio en la teoría como difícil de encontrar en la práctica. Otras ciencias más maduras que la informática, como por ejemplo la arquitectura y la ingeniería civil, trabajan el tiempo necesario para cumplir con este objetivo antes de comenzar con las tareas de campo. Todas las herramientas son válidas: maquetas, planos, memorias descriptivas, animaciones y un bagaje enorme de técnicas permiten dar visiones completas del producto terminado a los clientes, a los inversores, a los constructores, a los albañiles, subcontratistas y todos los participantes de la obra sin necesidad de poner el primer ladrillo. El ejemplo es válido para trasladarlo a la industria informática. En ésta, apenas se conocen los primeros esbozos del sistema, se comienza a programar, con la

ilusión de que programar y diseñar en paralelo ahorra tiempo y dinero. Lamentablemente muchas veces es considerado entre los programadores una viveza, un rasgo de picardía, obviar el trabajo previo de diseño y documentación.

- **Acotar y minimizar los costos**

Las dificultades en determinar costos reales para el desarrollo de los sistemas derivan principalmente de la falta de diseño. La carencia de diseño determina que no se pueda trazar un camino, un cronograma confiable, y sin éste no hay presupuesto. Esto se agrava enormemente con los problemas que genera el hecho de que los clientes (internos y externos) se defrauden al ver los resultados del trabajo a medida que van apareciendo. Esto impone cambios que se suman a las definiciones tardías del sistema, que van contra las definiciones tomadas implícitamente al empezar a programar. Las pulseadas y roces que supone este proceso de interacción y sucesivas aproximaciones generan en algunos casos verdaderas batallas campales de interna empresarial. Cuando los costos comienzan a dispararse, el departamento de sistemas comienza a acusar al resto de la empresa por las indefiniciones, y el resto de la empresa pide la cabeza de sistemas por los retrasos. El daño ya está hecho.

- **Poner foco en el usuario**

Determinar con la mayor claridad posible el producto final, y con esta definición determinar el camino a seguir, previene de las desviaciones que las presiones imponen. Siempre es posible torcer el camino hacia un derrotero erróneo; alcanza apenas con tomar una o dos decisiones equivocadas durante el proceso de desarrollo. Pero mientras que el diseño no garantiza que el usuario se va a sentir a gusto con el sitio o sistema, la inversa sí es garantizada.

Quien alguna vez trabajó como programador conoce la desesperación de tener que desarrollar un sistema pobremente especificado. Mientras que las primeras armas como programador hacen creer que eso es una oportunidad para brillar, la vida enseña que es una carta blanca para criticar impunemente el trabajo, incluir funcionalidades que no estaban previstas, obliga a hacer cambios enormes que de otro modo no hubieran sido necesarios.

- **Hacer creíbles y cumplibles los cronogramas**

Si bien este ítem podría haberse incluido dentro del ítem de costos por su parentesco, merece un capítulo aparte por sus consecuencias. Las causas son prácticamente las mismas, pero las

consecuencias son distintas: mientras que los problemas de costos impactan en la rentabilidad de los proyectos, la credibilidad es un intangible que cala mucho más hondo y los problemas de credibilidad tienen consecuencias mucho más profundas para las empresas. Las desviaciones en costos se arreglan con más dinero, la falta de credibilidad no. (12)

1.6.3 Análisis y Diseño Orientado a Objetos.

Para el desarrollo de software orientado a objetos no basta usar un lenguaje orientado a objetos. También se necesitará realizar un análisis y diseño orientado a objetos, teniendo como habilidad más importante la de asignar eficientemente las responsabilidades a los componentes de software.

Actualmente en la industria del desarrollo de software tenemos al UML como un estándar para el modelamiento de sistemas OO. Fue la empresa Rational que creó estas definiciones y especificaciones del estándar UML, y lo abrió al mercado. La misma empresa creó uno de los programas más conocidos hoy en día para este fin; el Rational Rose, pero también existen otros programas como el Poseidón que trae licencias del tipo community edition que permiten su uso libremente. (12)

El Análisis Orientado a Objetos (AOO) se basa en conceptos sencillos, conocidos desde la infancia y que aplicamos continuamente: objetos y atributos, el todo y las partes, clases y miembros. Puede parecer llamativo que se haya tardado tanto tiempo en aplicar estos conceptos al desarrollo de software. Posiblemente, una de las razones es el éxito de los métodos de análisis estructurados, basados en el concepto de flujo de información, que monopolizaron el análisis de sistemas software durante los últimos veinte años, ofrece un enfoque nuevo para el análisis de requisitos de sistemas software. En lugar de considerar el software desde una perspectiva clásica de entrada/proceso/salida, como los métodos estructurados clásicos, se basa en modelar el sistema mediante los objetos que forman parte de él y las relaciones estáticas (herencia y composición) o dinámicas (uso) entre estos objetos. Este enfoque pretende conseguir modelos que se ajusten mejor al problema real, a partir del conocimiento del llamado dominio del problema, evitando que influyan en el análisis consideraciones de que estamos analizando un sistema para implementarlo en un ordenador. Desde este punto de vista, el AOO consigue una abstracción mayor que el análisis estructurado, que modela los sistemas desde un punto de vista más próximo a su implementación en un ordenador (entrada/proceso/salida). (12)

A diferencia de otros métodos convencionales de diseño el DOO alcanza diferentes niveles de modalidad. La mayoría de los componentes de un sistema están organizados en subsistemas, módulos a un nivel del sistema, además, debe describir la organización específica de los datos, los cuales se encapsulan en objetos y el detalle de cada operación. Por tales razones la naturaleza del DOO reside en su capacidad de construir cuatro conceptos fundamentales: abstracción, ocultamiento de información, independencia, funcional y modularidad. (13)

1.6.4 Análisis y diseño según RUP.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes. Además si el sistema usará una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas. Otro producto importante de este flujo es la documentación de la arquitectura de software, que captura varias vistas arquitectónicas del sistema. (14)

Modelo de Análisis

Modelo conceptual, porque es una abstracción del sistema y permite aspectos de la implementación

Genérico respecto al diseño (aplicable a varios diseños)

Tres estereotipos conceptuales sobre las clases: Interfaz, Control y Entidad.

Modelo de Diseño

Modelo físico, porque es un plano de la implementación

No genérico, específico para una implementación

Cualquier número de estereotipos (físicos) sobre las clases, dependiendo del

lenguaje de implementación	
Menos formal	Más formal
Dinámico(no muy centrado en la secuencia)	Dinámico(muy centrado en la secuencia)
Bosquejo del diseño del sistema, incluyendo la arquitectura	Manifiesto del diseño del sistema, incluyendo su arquitectura(una de sus vistas)
Puede no estar mantenido en todo el ciclo de vida de software	Debe ser mantenido durante todo el ciclo de vida de software
Define una estructura que es una entrada esencial para modelar el sistema, incluyendo la creación del modelo de diseño. (15)	Da forma al sistema mientras que intenta preservar la estructura definida por el modelos de análisis lo más posible

1.7 Patrones.

Los Patrones son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas. Son una disciplina problema solución que está en constante evolución entre los diseñadores y desarrolladores que trabajamos con lenguaje orientado a objetos.

- Un patrón es:
 - ✓ La abstracción de una forma concreta que puede repetirse en contextos específicos.
 - ✓ Una información que captura la estructura esencial y la perspicacia de una familia de soluciones probadas con éxito para un problema repetitivo que surge en un cierto contexto y sistema.

Una unidad de información nombrada, instructiva e intuitiva que captura la esencia de una familia exitosa de soluciones probadas a un problema recurrente dentro de un cierto contexto. (16)

- Los Patrones de diseño se clasifican en:
 - ✓ Patrones de creación: Inicialización y configuración de objetos.
 - ✓ Patrones estructurales: Separan la interfaz de la implementación. Se ocupan de cómo las clases u objetos se agrupan, para formar estructuras más grandes.
 - ✓ Patrones de comportamiento: Más que describir objetos o clases, describen la comunicación entre ellos. (17)
- Aunque cuando se habla sobre los patrones salen a relucir los patrones de diseño como los más importantes en el mundo de la programación orientada a objeto, estos no son los únicos que existen ya que actualmente hay una inmensa cantidad de patrones que sería imposible para un programador el tratar de conocerlos todos, aquí le mostramos una cantidad modesta de estos por ejemplo:
 - ✓ **Patrones de Arquitectura:** Soluciones probadas para estructurar los componentes, como son Modelo-Vista-Controlador, arquitectura en tres capas, peer to peer, arquitectura orientada a servicios, por citar ejemplos.
 - ✓ **Patrones Web:** Soluciones probadas para la creación de sitios web, como son la maquetación en tres columnas, efectos de rollover, estructuras de blog.
 - ✓ **Patrones de Diseño:** Las soluciones probadas para el diseño de software.
 - ✓ **Patrones de Programación:** Soluciones específicas para algoritmos y estructuras de control, como son algoritmos de ordenación, procedimientos de recursión e iteración, etc.
 - ✓ **Patrones de Refactorización:** Soluciones para simplificar el código, como la variable explicativa, extracción de métodos, sobrecarga de constructores, etc. (15)

1.8 Conclusiones Parciales.

Las TIC se destacan por favorecer el mejoramiento de la cultura; han permitido ofrecer soluciones factibles para casi todas las ramas del desarrollo industrial. A través de ellas se encuentran poderosas herramientas que sirven de apoyo en la construcción de software, tanto para el desarrollo económico como social de un país o una empresa, pero para usarlas correctamente se debe tener el conocimiento necesario sobre el problema que se enfrenta. De ahí la importancia de conocer los conceptos asociados al dominio del problema; las ventajas que traerá consigo el Análisis y Diseño del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos. Además, se dieron a conocer varias empresas que pueden dar soluciones parciales u otras empresas que se dedican a la construcción de software de gestión para pozos, todas ellas liberadas bajo la sombra del software propietario.

Capítulo 2 Tendencias y Tecnologías Actuales a Desarrollar.

2.1 Introducción.

En los últimos años, el uso de las llamadas TIC, que engloban a la prensa, la radio, la televisión, el cine y la red mundial se ha incrementado. En especial cabe destacar el explosivo desarrollo de la Internet que permite comunicación diferida o en tiempo real. Además, se ha hecho necesario automatizar los procesos de negocio en cada empresa para así optimizar todo el envío y recibo de partes y reportes haciendo esto de una manera más segura y rápida. La mayoría de los software que se dedican a esto no son de fácil construcción además presentan una complejidad bastante elevada. Dada esa razón se hace necesario la utilización de metodologías de desarrollo de software con el propósito de seguir una serie de procesos que vienen con cada una de ellas; para así poder organizar y luego controlar el proyecto.

2.2 Metodologías de Desarrollo de Software

Desarrollar un software depende de un sin número de requisitos y actividades. Además, hay que pasar por muchas etapas; la elección de la metodología de desarrollo se convierte en un paso importante y trascendental en el éxito del producto. Estas tienen como principal objetivo el servir de guía y organizar las actividades para que se realicen todas las metas trazadas.

Existen dos grupos fundamentales de metodologías:

- **Las Tradicionales** (robustas o pesadas): Están pensadas para generar una gran cantidad de documentación y presentan una mayor cantidad de pasos, disciplinas y artefactos.
- **Las Ágiles:** Ponen mayor importancia en la capacidad de respuesta a cambios, acortan el tiempo de entrega al cliente, lo que se necesita gran confianza y habilidades en el equipo de desarrollo.

Dentro de las metodologías más usadas se encuentran el Proceso Unificado de Rational (RUP), la programación extrema o extreme Programming (XP), Microsoft Solution Framework MSF, Scrum, Iconix, Crystal Methodologies y AUP. Existen disímiles metodologías de desarrollo en la actualidad, así como categorías para diferenciar las mismas según sus

características. A continuación se presentan las más utilizadas en la actualidad; utilizando la diferenciación en Robustas (Pesadas) y Ágiles.

En la siguiente figura se puede hacer una comparación entre las metodologías ágiles y tradicionales para escoger la más adecuada según las características del proyecto a desarrollar:

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. Diferencias entre metodologías ágiles y no ágiles

2.3 Metodologías Pesadas.

Rational Unified Process RUP:

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, se divide en 4 fases de desarrollo con sus respectivos hitos, estas fases son: Inicio, Elaboración, Construcción y Transición.

Cada una de estas etapas es desarrollada mediante un ciclo de iteraciones; que consisten en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

RUP muestra como modelar software visualmente para capturar la estructura y comportamiento de arquitecturas y componentes. Las abstracciones visuales ayudan a comunicar diferentes aspectos del software; comprender los requerimientos, ver como los elementos del sistema se

relacionan entre sí, mantener la consistencia entre diseño e implementación y promover una comunicación precisa. El estándar UML (Lenguaje de Modelado Unificado), creado por Rational Software, es el cimiento para un modelado visual exitoso. (19)

La característica de RUP de ser Iterativo Incremental 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. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, es decir, deben ser seleccionadas y llevadas a cabo de una manera planeada. (20)

La iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto y trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior.

La iteración proporciona un resultado completo, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. Para ello, cada requisito se debe completar en una única iteración: el equipo debe realizar todas las tareas necesarias para completarlo y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos. (20)

En cada iteración el equipo evoluciona el proyecto o producto (hace una entrega incremental) a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos requisitos o mejorando los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la priorización de los requisitos en función del valor que aportan al cliente.

Se dice que RUP es una metodología Dirigida por Casos de Uso ya que un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema.

Los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de

desarrollo. Los casos de uso son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida.

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).

Para entender el planteamiento de que RUP está centrado en la arquitectura hay que tener en cuenta que el concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros stakeholders, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales (ej. desempeño, confiabilidad). (20)

La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. La arquitectura debe proveer espacio para la realización de todos los casos de uso, ambos deben evolucionar en paralelo. (20)

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 casos de uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. (20)

2.4 Metodologías Ágiles.

AUP:

(Agile Unified Process) AUP, es una versión simplificada de Rational Unified Process(RUP), se describe de una manera muy simple, fácil de comprender, el enfoque al desarrollo de software

para aplicaciones empresariales usando técnicas y conceptos ágiles se mantienen fieles a las de RUP. Se ha tratado de mantener de la manera más simple posible tanto en su enfoque como en su descripción.

En la siguiente figura se muestran los ciclos de vida de AUP. El primer aspecto que da a resaltar es que las disciplinas han cambiado, primeramente la disciplina de modelo incluye toda la modelación del negocio, sistema, análisis y diseño de RUP, la disciplina de modelado es una de las partes más importantes de AUP como se puede observar, pero no domina todo el proceso. En segundo lugar la configuración de la disciplina y el cambio de gestión son ahora la disciplina de gestión de la configuración. En el desarrollo ágil las actividades de gestión de cambio suelen formar parte de la disciplina del modelo.

Fases de AUP.

Incepción: El objetivo es identificar el alcance inicial del proyecto, la arquitectura del sistema, la financiación inicial y la aceptación de las partes interesadas (cliente).

Elaboración: El objetivo fundamental de esta fase es probar la arquitectura del sistema.

Construcción: Construir un software que trabaje sobre una base fuerte, que sea incremental dando la más alta prioridad a las necesidades de los clientes.

Transición: Validar y desplegar el sistema en su entorno de ejecución.

Disciplinas o Flujos de Trabajo de AUP.

Modelado: El objetivo de esta disciplina es entender el negocio de la organización, el dominio del problema que aborda el proyecto, y definir una solución viable para hacer frente al dominio del problema.

Implementación: El objetivo de esta disciplina es la de transformar el modelo de diseño en código ejecutable y para llevar a cabo un nivel básico de las pruebas, en las pruebas de unidad en particular.

Testeo: El objetivo de esta disciplina consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye encontrar defectos, validar que el sistema funciona según lo previsto, y verificar que se cumplen los requisitos.

Despliegue: El objetivo de esta disciplina es el plan para la entrega del sistema y ejecutar el plan para que el sistema a disposición de los usuarios finales.

Gestión de la Configuración: El objetivo de esta disciplina es la gestión de acceso a sus artefactos del proyecto. Esto incluye no sólo el seguimiento de las versiones de artefactos en el tiempo, sino también el control y la gestión de los cambios para ellos.

Gestión de Proyectos: El objetivo de esta disciplina es dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye los riesgos de gestión y dirección de personas (la asignación de tareas, el seguimiento del progreso, etc.), y coordinar con las personas y los sistemas fuera del alcance del proyecto para asegurarse de que es entregado a tiempo y dentro del presupuesto.

Ambiente: El objetivo de esta disciplina es apoyar el resto de los esfuerzos por garantizar que el proceso sea adecuado, normas de orientación (directrices), y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario. (21)

Extreme Programming (XP):

Esta es la tecnología más destacada de todos los procesos de desarrollos ágiles, fue formulada por Kent Beck, la programación extrema se diferencia de las otras tecnologías tradicionales en que pone más énfasis en la adaptabilidad que en la previsibilidad. (20)

Dentro de las características fundamentales del método se encuentran desarrollo iterativo e incremental que permite pequeñas mejoras; pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión; programación por parejas, permitiendo que el código sea revisado y discutido mientras se escribe. Propone que la mayor calidad del código escrito de esta manera sea más importante que la posible pérdida de productividad inmediata; frecuente interacción del equipo de programación con el cliente o usuario; corrección de todos los errores antes de añadir nueva funcionalidad. (20)

Otras de sus características más importantes incluyen la Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Propiedad del código compartida, este método promueve que todo el personal pueda corregir y entender cualquier parte del proyecto; simplicidad en el código. La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más

simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores. (20)

Prácticas XP

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. XP apuesta por un crecimiento lento del costo del cambio y con un comportamiento asintótico. Esto se consigue gracias a las tecnologías disponibles para ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas que se mencionan a continuación.

El juego de la planificación, entregas pequeñas, metáfora, diseño simple, pruebas, refactorización (Refactoring), programación en parejas, propiedad colectiva del código, integración continua, 40 horas por semana, cliente in-situ, estándares de programación, comentarios respecto de las prácticas. (15)

2.5 Conclusiones Parciales.

Después de haber conocido este número de metodologías que pueden ser utilizadas en el desarrollo de un proyecto de software se llega a la conclusión de que no existe una metodología universal para desarrollar con éxito un proyecto de software. Toda metodología debe ser adaptada al contexto del proyecto en sí. Las metodologías tradicionales, han intentado abordar la mayor cantidad de proyectos, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación.

2.6 Selección de la Metodología Adecuada.

Haciendo un análisis de todas las metodologías estudiadas se demuestra que las metodologías ágiles, pese a disminuir costos y brindar gran flexibilidad a los proyectos de software, son mejores en proyectos donde exista bastante incertidumbre y dentro de un entorno volátil. Donde no se conozcan los requisitos con exactitud. Según las características del proyecto que se está desarrollando se decide continuar utilizando una tecnología más tradicional.

Las Metodologías Tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo mientras que las ágiles están basadas en heurísticas

provenientes de prácticas de producción de código siendo ineficiente aplicarlas en un sistema donde se pueden identificar y definir claramente sus requerimientos tanto funcionales como no funcionales. Otra de las diferencias que llevaron a que se tomara la decisión de usar una metodología tradicional en vez de una ágil, es que las primeras presentan resistencia a los cambios y son impuestas externamente es decir por el cliente mientras que las segundas permiten que se produzcan cambios durante el proyecto. (20)

A continuación se muestra una tabla comparativa entre las metodologías más utilizadas utilizando varios criterios, mientras mayor sea el número más elegible será la metodología. Esto servirá de fundamentación sobre por qué se continúa con RUP como metodología de desarrollo de software.

Aspectos	MSF	SCRUM	RUP	XP	Crystal	ASD
Tipo	5	5	5	5	5	5
Modelo	5	5	5	5	5	5
Características Metodológicas(CM)						
Sistema como algo cambiante	5	5	5	5	5	5
Simplicidad	4	5	3	5	4	4
Adaptabilidad	4	4	5	3	5	5
Agilidad(CM)	4.3	4.8	4.3	4.8	4.5	4.7
Énfasis en la Arquitectura	5	1	5	1	1	5
Generación de Documentación del Negocio	3	2	5	2	3	2
Conocimiento	0	0	3	1	0	0
Restricciones	0	0	5	0	5	0
Total	3.18	2.54	4.61	2.68	3.35	3.1

2.7 Caracterizar las herramientas Case y procedimientos más utilizados.

2.7.1 Enterprise Architect 7.5

Es una herramienta de construcción y modelado de software de alto rendimiento basado en el estándar de UML 2.1, propiedad de la empresa SPARX SYSTEMS. Con una trazabilidad completa desde los requisitos iniciales hasta las decisiones de diseño de software. Provee el tipo de visualización y colaboración eficiente y robusta requerida en los entornos de desarrollo de software que actualmente son altamente demandantes. Como una solución de modelado verdaderamente ágil, Enterprise Architect provee una sobrecarga de instalación baja, un rendimiento brillante y una interfaz intuitiva (incluyendo una versión de “sólo lectura”). Permite la integración con los IDE Visual Studio y Eclipse, para el trabajo en equipo se integra con la herramienta de integración Teamcenter Systems Engineering. Posee un modulo que permite importar proyectos realizados en Vicio, además de un módulo que amplía las capacidades de la herramienta permitiendo incluso soportar la ingeniería de código directa e inversa para el lenguaje Python. EA es una herramienta para ayudar en su trabajo a prácticamente todo el equipo de trabajo, desde el analista hasta el equipo de despliegue, genera documentación compatible con MS Word. Soporte para ActionScript, Java, C++, Visual_Basic, Python y PHP, permite importar y exportar XML. Aunque es un software bajo licencia comercial (propietaria), sus licencias se consideran de bajo coste, además brinda la posibilidad de trabajar con una versión de prueba por 30 días. (15)

2.7.2 Rational Rose.

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 1.1. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (30)

Desarrollo Iterativo:

Rational Rose utiliza un proceso de desarrollo iterativo controlado (controlled iterative process development), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño; los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Trabajo en Grupo:

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

Características adicionales incluidas:

- Soporte para análisis de patrones ANSI C++, Rose J y Visual C++ basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java 1.5.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Soporte Enterprise Java Beans™ 2.0.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Capacidad de crear definiciones de tipo de documento XML (DTD) para el uso en la aplicación.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo. (31)

2.7.3 Visual Paradigm for UML 6.1 Enterprise Edition:

Visual Paradigm para UML 6.1 Enterprise Edition (VP-EE), Es una herramienta CASE propiedad de la compañía Visual Paradigm que integra en un paquete completo una gran cantidad de funcionalidades (herramientas) en torno al proceso de desarrollo de software. (15)

- Herramienta de modelado UML 2.1
- Herramienta de modelado de requerimientos que permite la captura, descripción, modelado de casos de uso.
- Herramienta de Modelado de Posesos de Negocio, que permite visualizar comprender u mejorar el modelado de procesos de negocio.
- Herramienta de modelado de base de datos, generación de documentación, generación automática Modelo Entidad-Relación y el Modelo Objeto-Relacional, a partir del modelo de clases persistentes.
- Herramienta de Mapeo de Modelos Objeto-Relacionales (Object-Relational Mapping), que automáticamente genera una capa de tratamiento de las tablas del modelo como objetos en el lenguaje que vaya a programar.
- Herramienta de Trabajo en Equipo, para ello puede utilizar VP Teamwork Server, CVS and Subversion.
- Herramienta generación e ingeniería inversa en Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python. Además, apoya la generación de código C #, VB. NET, Lenguaje de Definición de Objetos (ODL), Flash ActionScript, Delphi, Perl, Objective-C, y Ruby. Ingeniería inversa también apoya clase Java class . NET dll y exe, JDBC.
- Herramienta para generar documentación automáticamente a partir de los diagramas en diferentes formatos PDF, HTML y Microsoft Word.
- Herramienta de Integración con varios IDE (Eclipse, NetBeans, entre otros). Esta herramienta te permite sincronizar el proyecto (el código), con el modelo en el diseño, lo cual permite que no exista diferencia entre el diseño y el código.

Otra importante característica del VP-EE es la interoperabilidad, soporta importación y exportación de XMI de versiones 1.0, 1.2 y 2.1. Rational Rose archivos de proyecto (.MDL / .CAT) también pueden ser importados a través de la Rose Importer. Para aprovechar al máximo la interoperabilidad de productos de VP EE con otras aplicaciones, se han introducido

a la importación / exportación de modelado de proyectos desde / a un formato XML abierto, además de poder importar proyectos realizado en ERwin Studio, entre otros, de esta manera, los usuarios y proveedores de tecnología pueden integrar modelos en VP EE en sus soluciones con un mínimo esfuerzo. Otra de las características de VP EE es que no está diseñado hacia ninguna metodología de desarrollo en específico, se adapta a la mayoría de las metodologías de desarrollo existentes. (15)

A continuación se muestra la siguiente tabla comparativa para utilizarla como base y escoger la metodología adecuada.

Aspectos	VP-EE	Rational Rose
Modelado UML	Si.UML 2.1	Si. UML 1.0
Modelado de Base de Datos	Si	Si
Metodología de Desarrollo	Independiente de la Metodología	Orientado a RUP
Integración con el IDE	Alto	Bajo
Trabajo en Equipo	Si	Si
Generación de Código	Alto	Bajo
Generación de Documentación	Si	SI
Licencia	Comercial (Privada). Versión Community Gratis	Comercial (Privada)
Soporte	Multiplataforma	Multiplataforma

Conocimiento de la Herramienta	Medio	Bajo
---------------------------------------	-------	------

Después de haber analizado esta tabla comparativa se llega a la conclusión de que la metodología más factible para continuar con la elaboración del Sistema de Manejo Integral de Perforación de Pozos es Visual Paradigm Enterprise Edition. Resumiendo, las razones por la cual se escoge VP por encima de las demás herramientas abordadas:

1. Independiente de la metodología que se utiliza.
2. Soporta UML 2.0
3. Genera código directo e inverso en PHP 5.0.
4. Permite el Modelado de Base de Datos. Genera automáticamente el modelo entidad relación y el modelo objeto relacional, a partir de un modelo de clases persistentes.

2.8 Selección del Lenguaje de Modelación.

Para modelar el segundo ciclo de desarrollo del Sistema de Manejo Integral de Perforación de Pozos se ha decidido continuar con el uso de UML como lenguaje de modelado, siguiendo la metodología RUP.

UML es un conjunto de herramientas que permite modelar, analizar y diseñar sistemas orientados a objetos. Se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real.

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (20)

Los Diagramas de Comportamiento enfatizan en lo que debe suceder en el sistema modelado. Incluyen los Diagramas de Interacción que enfatiza sobre el flujo de control y datos entre los elementos del sistema modelado. Contienen objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos:

Diagramas de Secuencia para modelar el paso de mensajes entre objetos. Diagramas de

Colaboración para modelar interacciones entre objetos. Diagramas de Estado para modelar el comportamiento de los objetos en el sistema; muestra una máquina de estado que consta de estado, transiciones, eventos y actividades y los Diagramas de Actividad para modelar el comportamiento de los casos de uso, objetos u operaciones. (20)

Los Diagramas de Implementación para modelar la distribución del sistema: Incluyen los Diagramas de Componentes para modelar componentes; es decir: Organización y las dependencias entre un conjunto de componentes y el diagrama de despliegue, que permite la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

Puede conectarse con lenguajes de programación (Ingeniería directa e indirecta). Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.). Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos. Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas. Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar; Permite establecer conceptos y artefactos ejecutables; Es considerado como el mejor soporte a la planeación y al control de proyectos y Permite una alta reutilización y minimización de costos. (20)

2.9 Conclusiones Parciales.

Vistas las características de cada una de las metodologías expuestas anteriormente, se puede comprender el por qué de la tecnología utilizada en el desarrollo de la segunda iteración de esta aplicación, observa cuán superior es Visual Paradigm sobre Rational Rose, para un software sobre el que se utiliza Zend Studio como IDE de implementación. Además, este capítulo puede ser de gran utilidad a otros desarrolladores a la hora de escoger las herramientas de modelado más propicias cuando se va a construir un determinado sistema teniendo en cuenta las características del mismo.

Capítulo 3: Presentación de la Solución Propuesta.

3.1 Introducción.

En este capítulo se hace referencia al ambiente donde se desarrolla el negocio. Se realiza el análisis del mismo. Se hace una descripción detallada de actores, trabajadores y casos de uso que intervienen, así como sus diagramas correspondientes. Se plantean los requerimientos del sistema tanto los funcionales como los no funcionales, y para finalizar la descripción textual de cada uno de ellos.

3.2 Modelo de Negocio.

El modelado del Negocio tiene como principal objetivo:

- 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. (32)

3.2.1 Reglas del Negocio.

Las reglas del negocio describen condiciones o políticas que deben cumplirse o satisfacerse, por lo que regulan algunos aspectos en el negocio.

- A partir de las 12:00 a.m. y hasta las 6:00 a.m. los supervisores del pozo elaboran el Parte Diario y el Cierre Operativo Diario del Pozo.
- Parte Operativo de Perforación: Es la información que recibe el personal de la DIPP de los distintos pozos en perforación.
- Parte Operativo de Intervención: Se recibe por las tardes en la DIPP y se envía al día siguiente. (actualmente la información es recibida por medio del teléfono).
- Reporte Diario de Consumo (En Litros) de Perforación: que envían en el día sobre las 10:00 a.m., principalmente este es importante para el Logístico, pero se lo envían a todos los sectores mencionados anteriormente.
- Informe al MINBAS: Se envía un resumen semanal a la ministra.
- Para elaborar el Parte Diario o Cierre Operativo Diario del Pozo el supervisor debe tener

la información referente a las empresas de servicio al pozo, así como la información recogida del proceso de perforación del pozo.

- El Reporte Diario de Geología solo puede ser elaborado por la geóloga del pozo, para reflejar las características de la litología en la cual se está perforando.
- El Resumen del Pozo solo puede ser realizado por el Supervisor y se confecciona después de elaborados los reportes Diario de Perforación, Reporte de Encamisado y Reporte de Evaluación Técnica.
- El Reporte de Evaluación Técnica solo puede ser realizado por el Supervisor y se confecciona después de elaborado el Reporte Diario de Perforación.
- El Reporte de Construcción del Pozo solo puede ser realizado por el Supervisor y se confecciona después de elaborado el Reporte de Encamisado y el Reporte Diario de Perforación.
- El Informe Final es elaborado por el supervisor del pozo, incluye todos los reportes durante el periodo de perforación del pozo y solo es elaborado cuando termina el estado de perforación.

3.2.2 Actores y Trabajadores del Negocio.

Un actor del negocio puede ser cualquier individuo, entidad, organización, máquina o sistema de información externos con los que el negocio interactúa. Esto se muestra por la llamada asociación de comunicación entre el actor del negocio y el caso de uso del negocio que representa el proceso, lo que se modela como actor es el rol jugado para obtener resultados.

(15)

Actor	Descripción
Oficina de Supervisores	Es el encargado de supervisar todo el trabajo en el pozo.
Oficina de Despacho de la DIPP	Es el encargado de recepcionar y supervisar el trabajo con los reportes enviados de los pozos en perforación.

Oficina de Planificación de la DIPP	Es la encargada de Realizar el Plan Operativo mensual, Semanal y el Reporte al MINBAS.
CEINPET	Se encarga de recepcionar la información geológica enviada de los pozos en perforación y supervisar la construcción del Parte Diario de Geología.

Los **trabajadores** del negocio son personas o sistemas dentro del negocio que realizan las actividades que están comprendidas dentro de un caso de uso del negocio. Además, se encuentran dentro de la frontera del negocio y son los posibles a convertirse en usuarios del sistema que se quiere construir. (15)

Trabajador	Descripción
Supervisor de Pozo.	Es el encargado de la elaboración de todos los reportes diarios que se realizan en el pozo, por lo cual se convierte en una figura clave en el proceso de manejo y control de información del pozo.
Químico de Pozo.	Es el encargado de elaborar el Reporte Diario de Lodo, a partir de las pruebas realizadas al suelo según los días de la perforación. Este es entregado al supervisor del pozo para que posteriormente elabore el reporte diario de perforación del pozo.
Técnico de Control de la Producción.	Encargado de la elaboración de los reportes en la oficina de Despacho de la DIPP, a partir de la información enviada de los pozos de petróleo, por lo cual se convierte en una figura clave en este proceso.

Técnico de Control de la Planificación.	Es el encargado de la elaboración del plan semanal y mensual, además envía el reporte al MINBAS, por lo que se convierte en un figura clave en este proceso.
Geólogo de Pozo.	Encargado de elaborar el Reporte Diario de Geología del Pozo. Este se entrega al supervisor del pozo para que posteriormente se elabore el reporte diario de perforación del pozo.
Geólogo del CEINPET.	Encargado de elaborar el Reporte Diario de Geología, a partir de la información enviada por los geólogos del pozo. Este reporte es enviado posteriormente a la DIPP de perforación y a Cupet.

Referente a la versión anterior se divide el trabajador que fue la secretaria de despacho DIPP, en 2, técnico de control de la producción y técnico de control de la planificación, ya que el proceso en sí se realiza de esta manera.

3.2.3 Casos de Uso del Negocio.

Describen una secuencia de acciones realizadas en el negocio que produce un resultado de valor observable para un actor individual del negocio. Por tanto, desde la perspectiva de un actor individual, un caso de uso del negocio define el flujo de trabajo completo que produce los resultados deseados.

En este documento quedarán plasmados todos los casos de uso que actualmente forman parte del negocio pero solo se van a describir los que se identificaron en esta versión.

Casos de Uso del Negocio :

1. Módulo de Negocio, CEINPET

- Elaborar Reporte Diario de Geología

2. Módulo de Negocio, Despacho DIPP.

- Elaborar Parte Diario Operativo de Perforación.

- Elaborar Reporte de la Situación del Combustible.
- Elaborar Inventario en Campo del Material Pesante.
- Elaborar Reporte de Control de Equipos Tecnológicos para Servicio Residual.
- Elaborar Parte Operativo de Intervención.
- Elaborar Reporte de Control de Solicitudes de Servicios a la Perforación.
- Elaborar Plan de DTM (Desmontaje, Traslado y Montaje).
- Elaborar Plan Operativo Mensual.
- Elaborar Plan Operativo Semanal.
- Elaborar Informe Semanal al MINBAS.

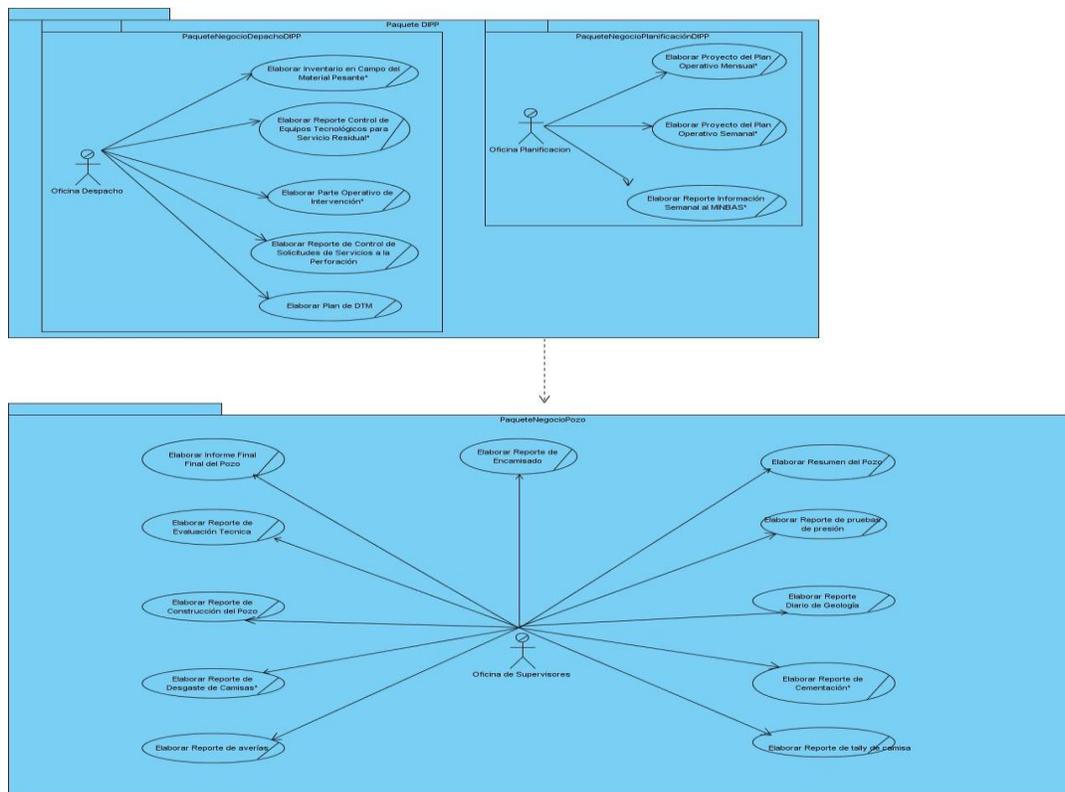
3. Módulo de Negocio, Pozo.

- Elaborar Cronograma de Perforación y Metraje.
- Elaborar Reporte Record de Barrenas.
- Elaborar Reporte de Composición de Herramientas.
- Elaborar Reporte de Fluidos de Perforación.
- Elaborar Reporte Diario de Perforación del Pozo.
- Elaborar Reporte Diario Operativo del Pozo.
- Elaborar Resumen Semanal de Costos del Pozo.
- Elaborar Reporte Diario de Costos de Productos Químicos.
- Elaborar Reporte Diario de Geología.
- Elaborar Reporte de Lodo.
- Elaborar Reporte Distribución de Tiempo.
- Elaborar Reporte de Presupuesto-Costo.
- Elaborar Reporte de Encamisado.
- Elaborar Reporte de Evaluación Técnica.
- Elaborar Resumen del Pozo.
- Elaborar Reporte de Construcción del Pozo.

- Elaborar Informe Final del Pozo.
- Elaborar Reporte de Tally de Camisas.
- Elabora el Reporte de Pruebas de Presión.
- Elaborar Reporte de Averías.
- Elaborar Reporte de Cementación.
- Elaborar Reporte de Desgaste de Camisas.

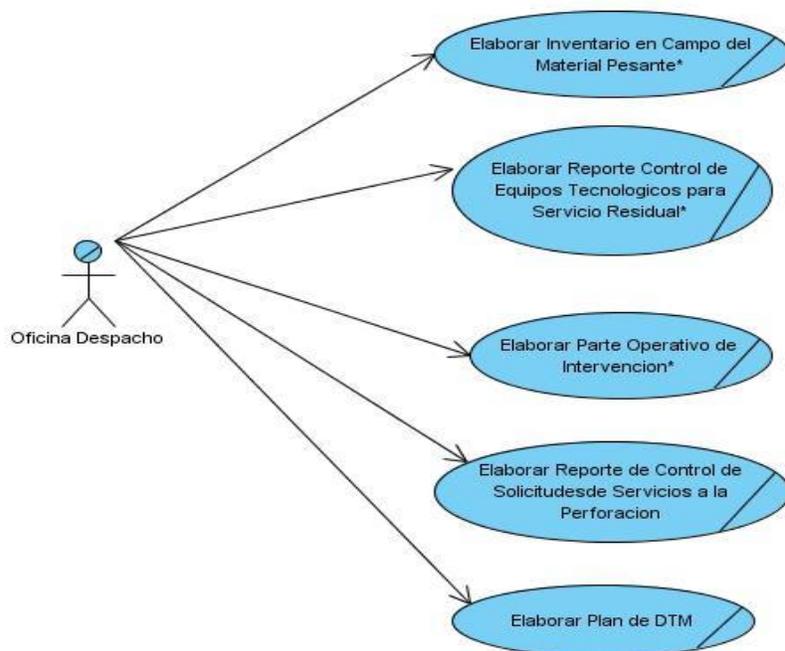
Un caso de uso del negocio representa a un proceso de negocio y una funcionalidad del sistema por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. De no identificarlos correctamente esto traería consigo un atraso para el proyecto y podría significar la pérdida del cliente.

3.2.4 Diagrama de Casos de Uso del Negocio.



3.2.5 Descripción textual de los casos de uso del negocio.

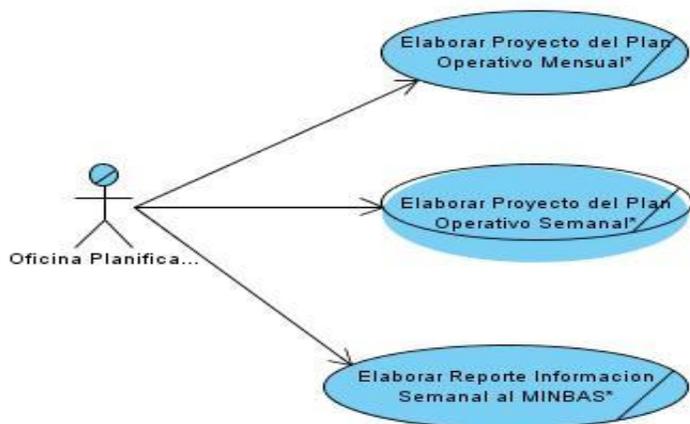
Módulo de Negocio de Despacho DIPP.



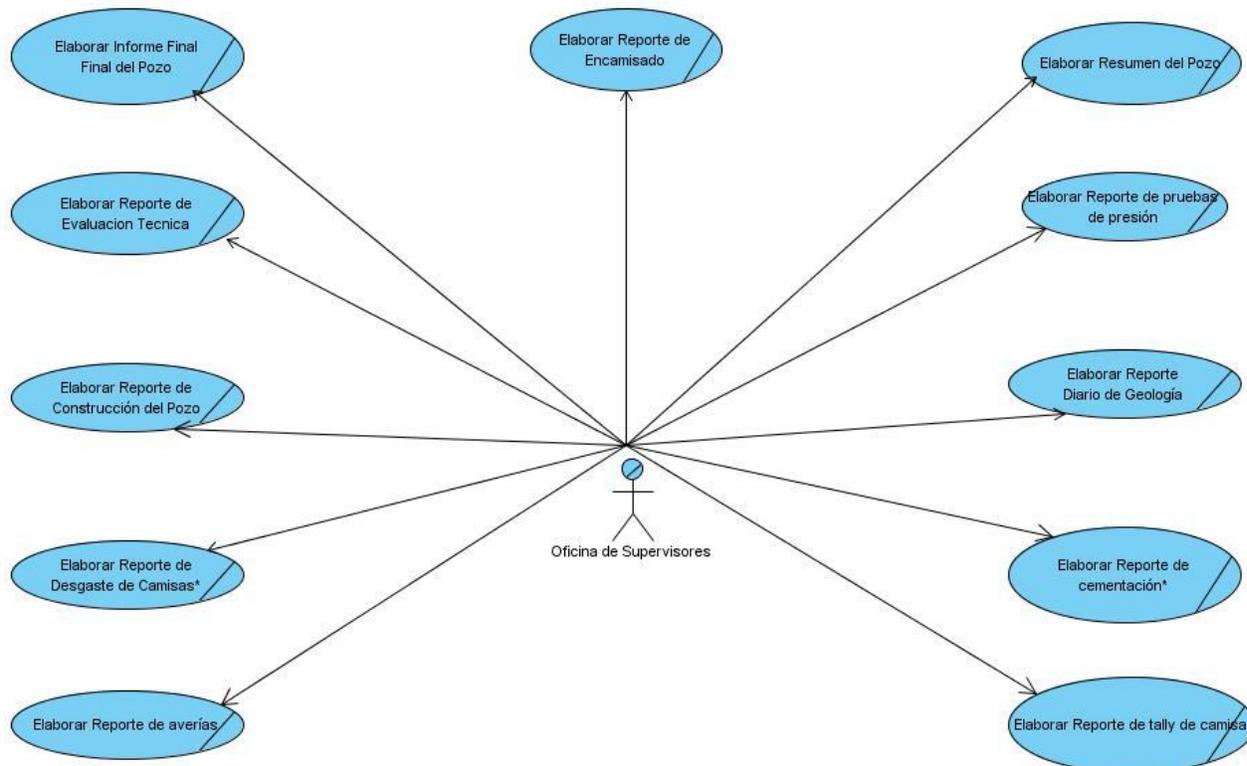
Caso de uso del negocio	Elaborar Reporte de Control de Solicitudes de Servicios a la Perforación.
Actor	Oficina Despacho de la DIPP.
Resumen	El caso de uso se inicia cuando la Oficina de Despacho de la DIPP recibe las solicitudes de servicios provenientes de los diferentes pozos en perforación e intervención, estos servicios se solicitan para los próximos días, se almacenan en un reporte y se envía por correo a la Dirección de la DIPP.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio

<p>1. La Oficina de Despacho de la DIPP solicita el Reporte de Control de Solicitudes de Servicios a la Perforación.</p>	<p>1.1 Elabora el Elabora el Reporte de Control de Solicitudes de Servicios a la Perforación.</p> <p>1.2 Almacena el Reporte de Control de Solicitudes de Servicios a la Perforación.</p> <p>1.3 Envía por correo el Reporte de Control de Solicitudes de Servicios a la Perforación a la Dirección de la DIPP.</p>
<p>Mejoras propuestas:</p>	

Módulo de Negocio Planificación DIPP.



Módulo de Negocio Pozo.



Caso de uso del negocio	Elaborar Reporte Diario de Geología
Actor	Oficina de Supervisores
Trabajadores	Geólogo del Pozo
Resumen	El caso de uso se inicia cuando la Oficina de Supervisores solicita el Reporte Diario de Geología. El Geólogo del Pozo consulta el Reporte de Lodo, emplea estos datos y elabora el reporte diario de geología, lo almacena, lo envía al Geólogo del CEINPET.
Casos de uso asociados	Ninguno

Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores solicita el Reporte Diario Geología.	<p>1.1 El Geólogo del Pozo consulta los datos entregados por los químicos.</p> <p>1.2 El Geólogo del Pozo procesa los datos recopilados del análisis de los cortes.</p> <p>1.3 El Geólogo del Pozo elabora el Reporte Diario de Geología.</p> <p>1.4 El Geólogo del Pozo almacena Reporte.</p> <p>1.5 El Geólogo del Pozo entrega el Reporte.</p>
1. La Oficina de Supervisores recepciona este Reporte	
2. La Oficina de Supervisores almacena el Reporte	
Mejoras propuestas:	

Caso de uso del negocio	Elaborar Reporte de Encamisado
Actor	Oficina de Supervisores.
Resumen	El caso de uso se inicia cuando es detenido el proceso de perforación para empezar a introducir un determinado tipo de camisa.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores	1.1 El Supervisor del Pozo confecciona el

ordena Elaborar el Reporte de Encamisado.	Reporte a partir de los datos de la camisa y la maniobra realizada durante el encamisado. 1.2 El Supervisor del Pozo almacena este Reporte.
Mejoras propuestas:	

3.3 Requisitos del software.

Para modelar el sistema, se identifican sus requisitos, tanto funcionales como no funcionales, y se modelan los funcionales en términos de casos de uso del sistema. Se especifican las condiciones o capacidades que el sistema debe cumplir. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos. (15)

3.3.1 Requisitos Funcionales.

Módulo de Negocio Pozo.

- ✓ Elaborar Parte Diario de Geología.
- ✓ Insertar Datos del Parte Diario de Geología.
- ✓ Actualizar Datos del Parte Diario de Geología.
- ✓ Elaborar Reporte de Encamisado.
- ✓ Insertar Datos del Reporte de Encamisado.
- ✓ Actualizar Datos del Reporte de Encamisado.
- ✓ Actualizar Datos del Resumen del Pozo.
- ✓ Elaborar Reporte de Evaluación Técnica.

Estos requisitos tienen como función generar de manera automática el Reporte de Evaluación Técnica, el cual toma información que necesita de las actividades diarias del Reporte Diario de Perforación.

- ✓ Elaborar Reporte de Construcción de Pozo.
- ✓ Insertar Datos del Reporte de Construcción de Pozo.
- ✓ Actualizar Datos del Reporte de Construcción de Pozo.

Estos requisitos tienen como función gestionar la información de la construcción íntegra del pozo, este reporte se va elaborando a medida que se van terminando los intervalos de perforación.

- ✓ Elaborar Informe Final de Pozo.

Este requisito tiene como función la generación automática del Informe Final del Pozo, el cual toma los datos que necesita de varios reportes creados en el tiempo que el pozo estuvo en perforación.

- ✓ Elaborar Reporte de Tally de Camisas.
- ✓ Insertar Datos en el Reporte del Tally de Camisas.
- ✓ Actualizar Datos del Reporte de Tally de Camisas.

Estos requisitos tienen como función la Gestión del Reporte de Tally, el cual no es más que ir tomando las camisas en el orden en que se encuentran en el Reporte del Tally Medición de Camisas para llevar la longitud de las camisas en el pozo a medida que se van introduciendo en el mismo.

- ✓ Elaborar Reporte de Pruebas de Presión.

Este reporte está conformado por dos el Reporte de Leak of Test o Reporte de prueba de fit.

- ✓ Elaborar Reporte de Leak of Test.
- ✓ *Insertar Datos en el Reporte LOT.*
- ✓ *Actualizar datos en el Reporte de LOT.*
- ✓ *Insertar Datos en el Reporte de FIT.*
- ✓ *Actualizar Datos del Reporte de FIT.*

Estos requisitos tienen como función la conformación del Reporte de Leak of Test o la gestión del Reporte de Prueba de FIT, los cuales, son una prueba de presión a la que se somete el pozo una vez que se le termina de colocar un tipo de camisa específico. Estos reportes brindan la misma información por lo que se puede elaborar uno de ellos o los dos.

- ✓ Elaborar Reporte de Averías.
- ✓ Insertar Datos en el Reporte de Averías.
- ✓ Actualizar Datos del Reporte Averías.

Estos requisitos tienen como función la Gestión de todas las Averías que puedan ocurrir en el pozo, la causa y como darle solución a las mismas.

3.3.2 Requisitos No Funcionales.

Apariencia o interfaz externa.

El sistema tendrá un menú de herramientas para que el usuario pueda disponer de él en la medida de sus necesidades, visible todo el tiempo para propiciar el fácil acceso a las funcionalidades del software.

El sistema tendrá un ambiente agradable y sencillo, combinado con colores y una estructura amigable que permita que el usuario se adapte con facilidad.

El sistema tendrá una buena navegabilidad, propiciando varias opciones para acceder a cada servicio brindado.

La interfaz contará con una gama de colores, para visualizar los Registros de Pozo, el Histograma de frecuencia de las curvas y las Propiedades de Gráficas cruzadas entre otras estructuras que lo requieran.

Ayuda.

El sistema debe contar con un manual de usuario.

El sistema debe garantizar que la opción de Ayuda esté visible todo el tiempo, para que el usuario la pueda consultar la cantidad de veces necesarias y facilite su trabajo con cada servicio.

El sistema mostrará mensajes de ayuda durante las operaciones que lo requieran, permitiendo la orientación del usuario en cada procedimiento.

3.4 Modelo de Sistema

3.4.1 Actores del Sistema.

Actor del Sistema	Descripción
Supervisor de Pozo.	Inicia los Casos de Uso del Subsistema Pozo donde participa el Supervisor de Pozo (Trabajador del Negocio), convirtiéndose en un actor de alta importancia, ya que todos los casos de uso que inicia son críticos para sistema.
Químico de Pozo.	Inicia el caso de uso Gestionar Reporte de Lodo.
Técnico de Control de la Producción DIPP.	Inicia los Casos de Uso del Subsistema de Despacho de la DIPP.
Técnico de Control de la Planificación DIPP.	Inicia los Casos de Uso del Subsistema de Planificación de la DIPP.
Geólogo de Pozo.	Inicia el Caso de Uso Gestionar Reporte Diario de Geología el Subsistema Pozo, así como el Caso de Uso Gestionar Columna Litológica del Pozo del Subsistema Visualización de Información.
Geólogo del CEINPET.	Inicia el Caso de Caso de Uso Gestionar Columna Litológica del Pozo del Subsistema Visualización de Información.
Administrador del Sistema	Encargado de la Seguridad del Sistema, además se encarga de insertar lo datos iniciales del pozo.

Como se puede observar se elimina el actor Direccional ya que este es un servicio que lo efectúa una compañía extranjera, el sistema a diseñar no va a intervenir en este proceso.

3.4.2 Diagrama de Casos de Uso del Sistema.

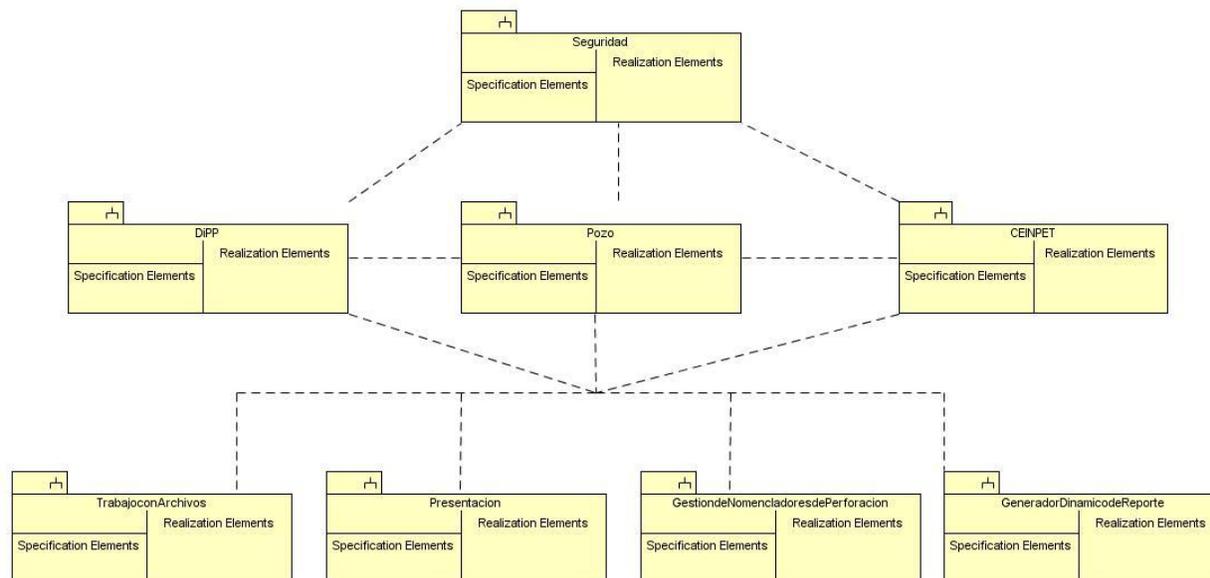
- Subsistema Pozo.

Este es el subsistema más importante ya que es para el cual está dirigido el producto realmente, es el encargado de responder a todos los requerimientos derivados del mismo.

- Subsistema DIPP.
 - ✓ Módulo Despacho DIPP.

Contiene todos los casos de uso que responden a los Requerimientos derivados al Módulo de Despacho de la DIPP, garantizando la trazabilidad del sistema.

3.4.3 Diagrama de Casos de Uso por Subsistemas.



Constituye una buena práctica el agrupar los casos de uso del sistema en subsistemas de acuerdo a sus funcionalidades, principalmente para el entendimiento del sistema mismo, como se puede observar en esta versión se han agregado dos subsistemas más: El componente de Gestión de Nomencladores de Perforación, y el Generador Dinámico de Reportes. Además, el componente Visualización de la Información pasó a ser el componente de presentación.

3.4.4 Descripción Textual de los Casos de Uso del Sistema.

Subsistema Pozo:

Caso de Uso:	Gestionar Reporte de Encamisado
---------------------	---------------------------------

Actores:	Supervisor del Pozo
Resumen:	Este caso de uso se encarga de gestionar el reporte de encamisado una vez que se completa un tipo específico de camisa en el pozo.
Precondiciones:	El usuario debe estar autenticado en el sistema.
Referencias	RF 2.12, RF 2.13, RF 2.14
Prioridad	Crítico

Flujo Normal de Eventos

Sección “Insertar Datos del Reporte Encamisado”

Acción del Actor	Respuesta del Sistema
1. El Supervisor del Pozo selecciona la opción de Insertar Datos en el Reporte de Encamisado.	1.1 El Sistema muestra una interfaz con un formulario para insertar los datos del reporte.
2. El Supervisor del Pozo introduce los datos, así como el tipo de reporte de Encamisado.	2.1 El Sistema habilita la opción de Guardar y Enviar.
3. El Supervisor del Pozo selecciona la opción de Guardar y Enviar.	3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación.
4. El Supervisor del Pozo confirma la operación de Guardar y Enviar.	4.1 El Sistema comprueba que los datos no contengan errores 4.2 El Sistema almacena el reporte en la base de datos 4.3 El Sistema muestra un mensaje del éxito de la operación.

Sección “Actualizar Datos del Reporte de Encamisado”

Acción del Actor	Acción del Actor
1. El Supervisor del Pozo selecciona la	1.1 El Sistema muestra un formulario con todos

<p>opción de Actualizar Datos del Reporte de Encamisado.</p> <p>2. El Supervisor actualiza los datos.</p> <p>3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar.</p> <p>4. El Supervisor de Pozo confirma la operación de Guardar y Enviar.</p>	<p>los datos que se encuentran almacenados en el reporte.</p> <p>2.1 El Sistema verifica si los datos son correctos.</p> <p>2.2 El Sistema habilita la opción de Guardar y Enviar.</p> <p>3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación.</p> <p>4.1 El Sistema comprueba que los datos no contengan errores.</p> <p>4.2 El Sistema almacena el reporte en la base de datos.</p> <p>4.3 El Sistema muestra un mensaje del éxito de la operación.</p>
--	--

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
1a. El usuario escoge insertar datos del reporte de liner sin haber llenado el reporte de técnico.	1a.1 El sistema envía un mensaje de que debe insertar primeros datos del reporte técnico.
2a. Inserta los Datos o Actualiza los Datos.	<p>2a.1 “Datos introducidos incorrectos” El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% ,etc.) en ese campo.</p> <p>2a.2 El Sistema limpia todos los caracteres extraños.</p>
4a. No confirma la operación de Guardar y Enviar.	4a.1 El Sistema no envía los datos.
4b. Confirma la operación de Guardar y Enviar.	4b.1 El Sistema muestra un mensaje: “Los datos enviado son incorrectos”.
5a. El usuario escoge insertar datos del reporte técnico sin haber llenado el reporte	5a.1 El sistema envía un mensaje de que debe insertar primeros datos del reporte de

de superficie.	superficie.
Pos condiciones	Redirección a la Interfaz Principal

3.5 Conclusiones Parciales.

En este capítulo se mostraron todas las descripciones y representaciones gráficas de los casos de uso del negocio, como base para enmarcar el diseño del segundo ciclo de desarrollo de Sistema de Manejo Integral de Perforación de Pozos. Se Identifican nuevos requerimientos funcionales y no funcionales, y se representan los diagramas de casos de uso del sistema.

Capítulo 4 Diseño del sistema.

4.1 Introducción.

Este capítulo trata todos los temas relacionados con el diseño del sistema usando la metodología RUP, se muestran todos los artefactos referentes al modelo de diseño orientado a symfony. Los diagramas de diseño se han acoplado lo más cercano posible al framework logrando una vista mayor estructurada y organizada que pueda ser utilizada de guía por los programadores del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos. También se abordará sobre los patrones de diseño utilizados y otros elementos fundamentales en esta fase de la metodología.

4.2 Modelo de Análisis.

El modelo de análisis no es más que el paso que contribuye a lograr entender todos los aspectos internos del sistema y ayuda a refinar los requerimientos del software. Según RUP el análisis puede no ser generado. Esto solo se debe poner en práctica en proyectos donde se tenga un conocimiento elevado del negocio o halla una experiencia de trabajo en los mismos.

La presente investigación como se expuso anteriormente se trata de la Segunda Versión del Sistema de Manejo Integral de Perforación de Pozos, por lo que cuenta con una investigación de respaldo y una experiencia de trabajo en el desarrollo del software. Tomando en cuenta lo anterior se decide no generar el artefacto de análisis ya que como la metodología plantea este es solo un paso transitorio al diseño.

4.3 Modelo de Diseño.

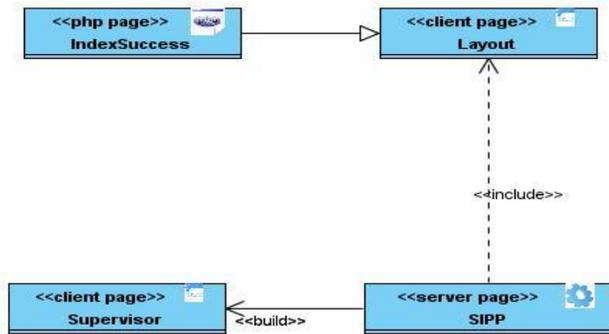
El diseño constituye la culminación de la fase de elaboración y el comienzo de las iteraciones de construcción. Mediante el siguiente diseño se modela el sistema logrando que el mismo soporte todos los requerimientos funcionales, no funcionales y sus restricciones. Logrando una arquitectura estable.

4.3.1 Artefactos del Diseño.

El hecho de usar un framework para el desarrollo restringe el diseño obligando al diseñador a utilizar la estructura del mismo, la actual versión del sistema trabaja con tres módulos (componentes) fundamentales, los cuales se identifican sus respectivas vistas y controladoras:

- Modulo Supervisor

✓ Vista.

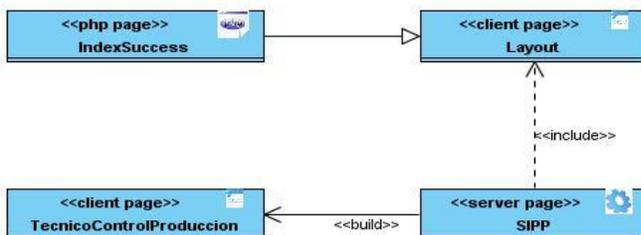


✓ Controladora.

SupervisorActions
+executeIndex()
+executeEncamisado()
+executeCosto_pozo()

- Módulo Despacho DIPP.

✓ Vista.

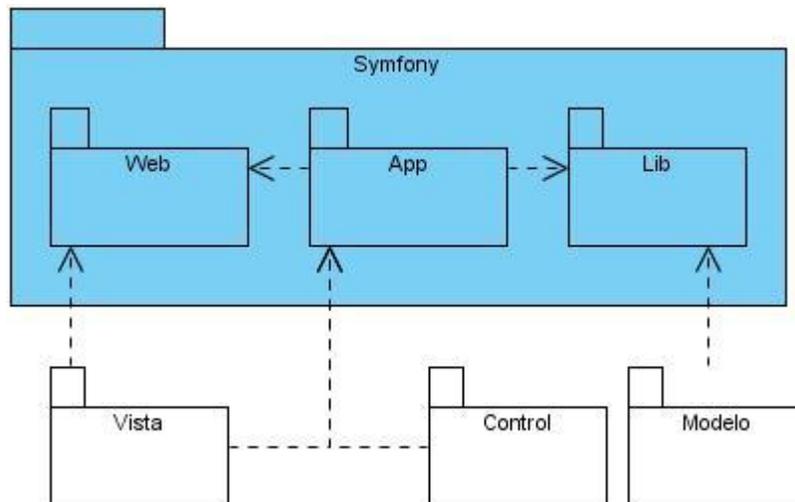


✓ Controladora.

dippActions
+executeIndex()
+executePointervencion()
+executeExportar()
+executeParte_operativo_perforacion()
+executeTarea_seguintimiento_preorizado()
+executeTarea_principal_semanal()
+executeParte_operativo_dipp()

A continuación se muestra la manera en que se ve la estructura del diseño utilizando symfony. Se observa el uso del patrón Modelo Vista Controlador (MVC), el cual separa la lógica de

negocio (modelo) de la presentación (vista); esto permite que pueda haber varias vistas que utilicen el mismo modelo y controlador.



Como se aprecia anteriormente el proyecto queda dividido en 3 carpetas fundamentales:

App: Contiene un directorio por cada aplicación del proyecto. Almacena todos los módulos creados con sus respectivos success (vistas) y sus actions (controladoras).

Web: En este directorio se almacenan los únicos archivos visibles desde internet, vistas, imágenes, CCS y archivos JavaScript, entre otros.

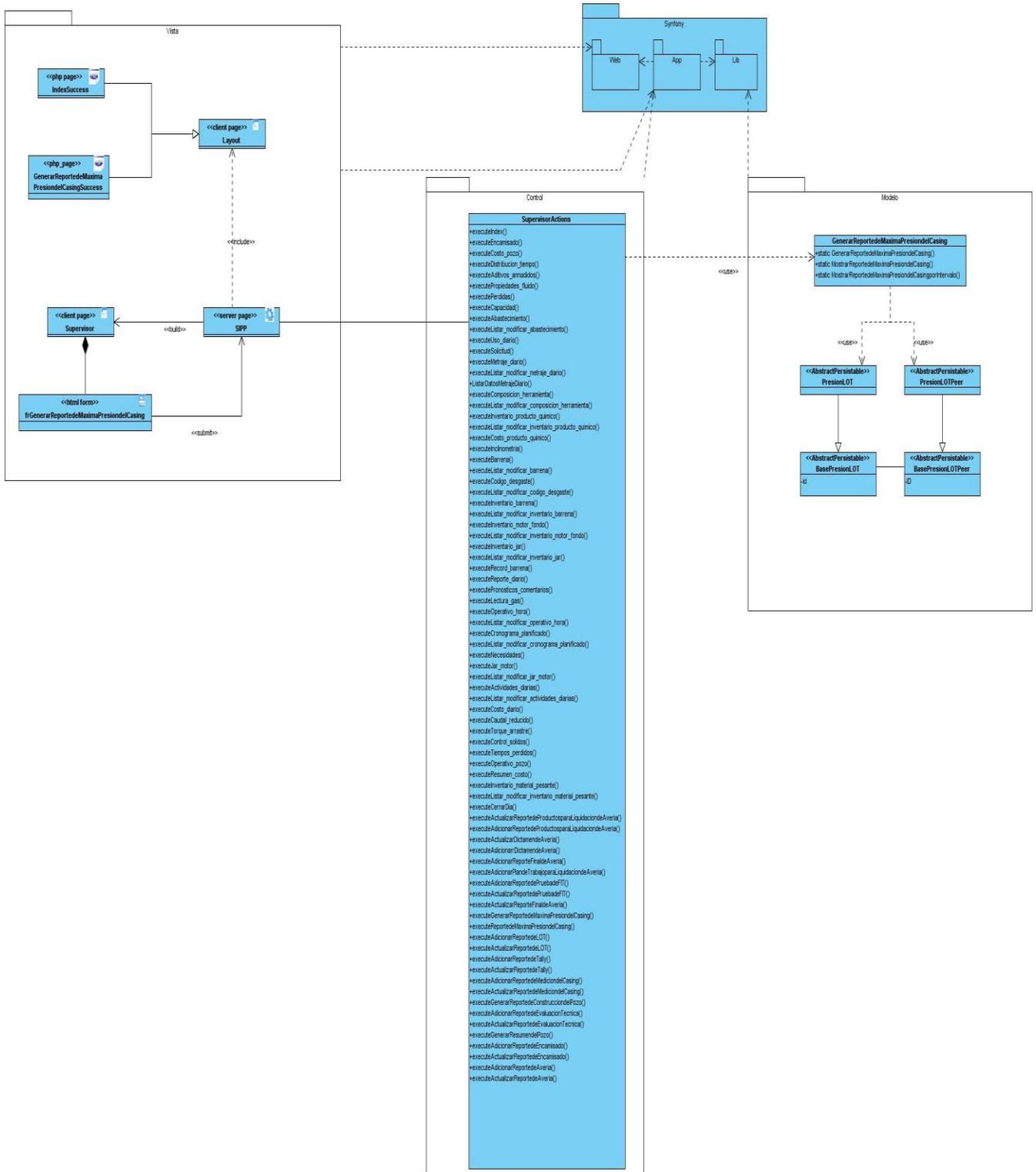
Lib: Almacena clases y librerías externas, así como todo lo referente a la del modelo, la cual se agrupa específicamente en la carpeta modelo. En esta carpeta se encuentran las om donde se almacena la abstracción de los datos y el mapeo de la base de datos ubicado en las map.

4.3.1.1 Diagramas de Clases.

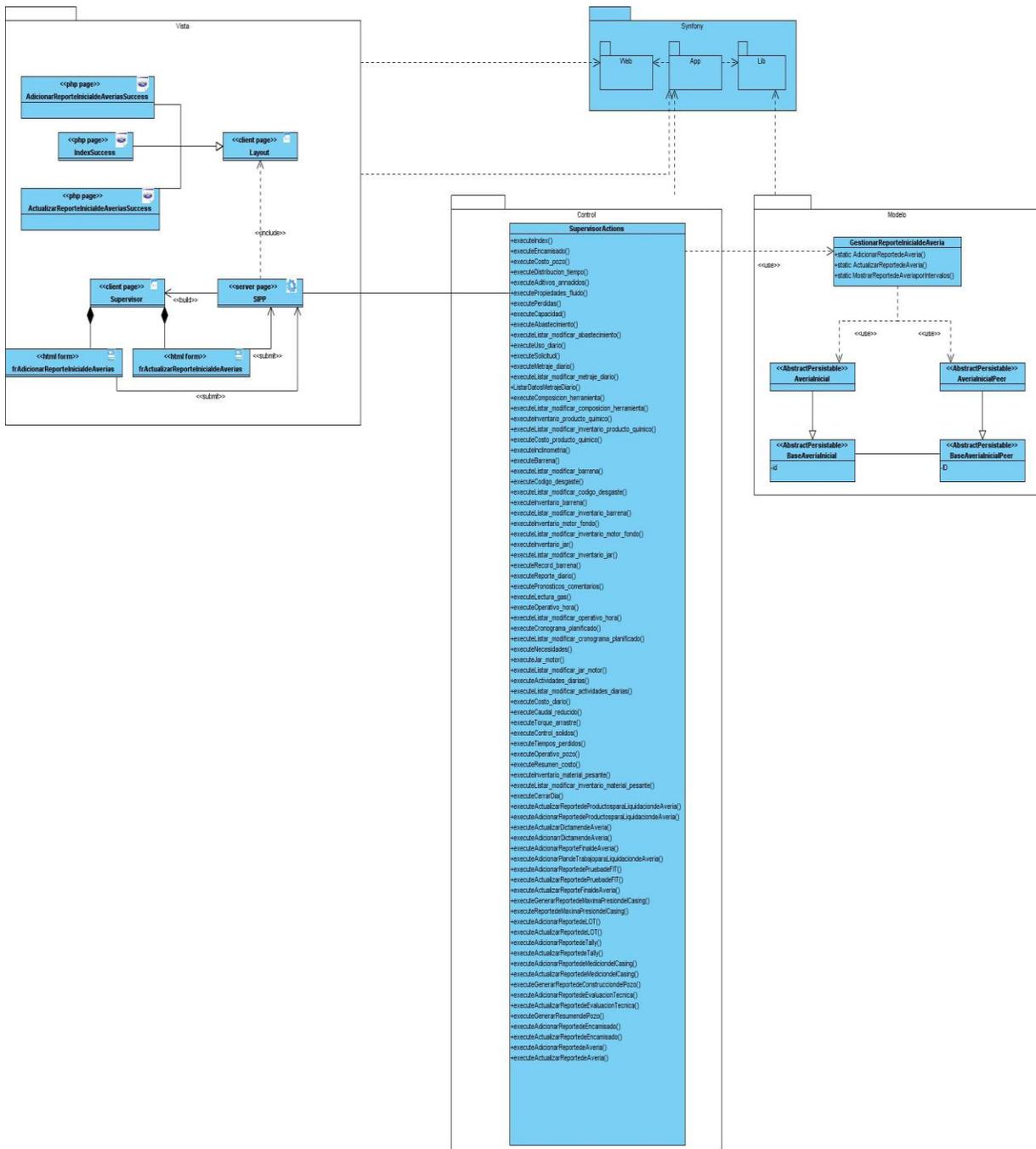
Los diagramas de clases del diseño describen las especificaciones de las clases, y de las interfaces de una aplicación, a continuación se presentan los diagramas de diseño pertenecientes a los casos de uso críticos en esta iteración, los que se encuentran agrupados por subsistemas:

- Subsistema Pozo.

✓ Generar Reporte de Máxima Presión del Casing.



✓ Gestionar Reporte Inicial de Avería.



4.3.1.2 Patrones de Diseño.

Los patrones de diseño se refieren a las descripciones de comunicación de las clases y los objetos que pueden personalizarse para resolver un problema de diseño general en un contexto particular. Symfony, al igual que la mayoría de los framework sigue las mejores prácticas y patrones de diseño para la web. (34)

A continuación se especifican los patrones utilizados para el modelado del diseño y donde son usados.

Patrones GRASP

Son patrones generales de software para asignación de responsabilidades, aunque más bien se consideran buenas prácticas de aplicación, recomendable en el diseño de software.

Experto: El patrón de diseño Experto está ideado para que la responsabilidad de realizar una labor sea de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras se consideren los mismos aspectos del sistema:

- ✓ Lógica de negocio
- ✓ Persistencia a la base de datos
- ✓ Interfaz de usuario. (15)

Controlador: Está diseñado para asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. (34)

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con la que conoce y con que concurre a ellas. El bajo acoplamiento posibilita que los cambios locales en una clase, no ocasione cambios en otras y que estas sean más fáciles de entender cuando estén aisladas, por lo que son más fáciles de reutilizar. (15)

Alta Cohesión: El uso de este patrón se evidencia en la implementación de la clase Action, la cual tiene la responsabilidad de definir las acciones para las plantillas y, paralelo a esto colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las

propiedades, expresado de otra manera, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas. (34)

Decorator (Envoltorio): Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora el contenido de la plantilla. (34)

Front-Controller: Para aplicaciones Web se recomienda utilizar este patrón que obliga a que todas las peticiones hechas a la aplicación pasen por un servlet Controlador.

- El controlador proporciona un punto de entrada único que controla y gestiona las peticiones Web realizadas por los clientes.
- Teniendo este único punto de entrada se evita tener que repetir la misma lógica de control en todos los .jsp.
- Normalmente se utiliza junto con un Dispatcher que es el responsable de redirigir el flujo de ejecución hacia el jsp adecuado. Este Dispatcher puede ser realizado por el propio controlador o estar en una clase a parte.

4.4 Conclusiones Parciales.

Luego de terminado el capítulo donde se presenta el resultado del trabajo de diploma se puede llegar a la siguiente conclusión: El modelo de diseño es imprescindible para lograr un sistema con calidad. El diseño de clases usando la estructura del funcionamiento del framework resulta engorroso y no existe una estandarización en el modelado de un sistema basado en symfony.

Conclusiones

Partiendo de la investigación realizada donde se evidencia la necesidad de diseñar los artefactos ingenieriles para el segundo ciclo de desarrollo del Sistema de Manejo Integral de Perforación de Pozo, se llegan a las siguientes conclusiones:

- Se logran identificar nuevos procesos de negocio en los pozos de petróleo en perforación.
- Se definieron los requisitos funcionales y no funcionales teniendo en cuenta las necesidades del cliente.
- La correcta identificación de la metodología para continuar el proyecto permitió agilizar el proceso hacia el diseño sin sacrificar calidad.
- Mediante el estudio del arte se evidenció la ventaja que representa el software libre sobre el propietario y el punto de partida para el continuo desarrollo del sistema.
- La correcta utilización de las herramienta VP 6.4 EE for UML permitió la representación del sistema a través de modelos sustentados por UML.
- Se logra un diseño que cumple con las restricciones del framework, aplicando patrones para la óptima construcción del mismo.

Recomendaciones

Las siguientes recomendaciones constituyen una recopilación de las principales actividades con el objetivo de realizar mejoras a la aplicación en siguientes versiones:

- Continuar con la investigación de los procesos de negocio para identificar nuevas funcionalidades, principalmente en el modulo DIPP.
- Investigar sobre el uso de nuevas metodologías que permitan disminuir tiempo sin sacrificar calidad.
- Investigar sobre nuevas tecnologías de modelado que sean independientes de la metodología y mejoren en cuanto a rapidez.
- Identificar nuevas funcionalidades para la construcción de componentes que logren hacer del sistema un producto para el mercado internacional.
 - Configuración del Proceso Gestión de Información.
 - Generador de Reportes
 - Internacionalización
 - Configuración de Interfaces
 - Historial de Usuario
 - Comunicación

Referencias Bibliográficas.

1. Tecna. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.tecna.com>.
2. Inc, RCLAB S.R.L. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.rclabsrl.com.ar/>.
3. Shlumberger. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.slb.com/>.
4. SAP. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.sap.com/spain/sme/whysap/industries/oil-gas/index.epx>.
5. Inc, Neuralog. [En línea] [Citado el: 14 de 12 de 2009.] http://www.neuralog.com/www_backup_files/espanol/index.htm.
6. (PLS), Petrokem Loggin Services. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.petrokemls.com/>.
7. InfOil. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.infoil.com.ar/>.
8. cee. [En línea] [Citado el: 10 de 02 de 2010.] <http://cee.com/products/sampling/WellWizard/WellWizard.html>.
9. Free Download Manager. [En línea] [Citado el: 10 de 02 de 2010.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/..
10. WellSight. [En línea] [Citado el: 10 de 02 de 2010.] <http://www.wellsight.com>.
11. Peloton. Drilling & Well Data Software Solutions. [En línea] [Citado el: 10 de 02 de 2010.] <http://www.peloton.com/es/..>
12. Ramírez, Alexandra. [En línea] [Citado el: 11 de 03 de 2010.] http://s3.amazonaws.com/lcp/elida-nieves/myfiles/AS_Clase-231008-1.doc..
13. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. La Habana : Felix Varela, 2005.
14. Jacobson, I., Booch, G., Rumbaugh J. *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
15. Cuevas, David Tavares. *ANÁLISIS Y DISEÑO DEL SISTEMA DE MANEJO INTEGRAL DE PERFORACIÓN DE POZOS*. Ciudad de la Habana : s.n., 2009.
16. Infosintesis Solutions Group, S.L. . Formación, asesoramiento y desarrollo de proyectos informáticos . [En línea] [Citado el: 13 de 03 de 2010.] www.infosintesis.net.
17. Suarez, Jose Gregorio. [En línea] 28 de 11 de 2009. [Citado el: 13 de 03 de 2010.] www.Los-patrones-de-diseno.htm.
18. UNAP. [En línea] 22 de 03 de 2004. [Citado el: 5 de 3 de 2009.] <http://www.unap.cl/~setcheve/Metrica/m/index.html>.

-
19. Sanchez, María A. Mendoza. [En línea] [Citado el: 15 de 03 de 2010.] http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.
 20. Castillo, Danis Rego. *Análisis y Diseño del Sistema para el análisis petrofísico*. 2009.
 21. [En línea] 2005. [Citado el: 15 de 03 de 2010.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
 22. ProyectosAgiles.org. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.proyectosagiles.org/que-es-scrum>.
 23. —. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.proyectosagiles.org/beneficios-de-scrum>.
 24. José H. Canós, Patricio Letelier y M^a Carmen Penadés. [En línea] [Citado el: 16 de 03 de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
 25. [En línea] 23 de 5 de 2008. [Citado el: 16 de 03 de 2010.] <http://crystalmethodologies.blogspot.com/>.
 26. Java Mexico-Comunonad de Desarrolladores Mexicanos. [En línea] 27 de 02 de 2010. [Citado el: 16 de 03 de 2010.] http://www.javamexico.org/blogs/carraro/que_es_dsdm.
 27. Huarachi, Maritza. *Ingernieria de Softeare*. EL ALTO – BOLIVIA : s.n., 2009.
 28. INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA △. *Herramientas Case*. 1999.
 29. Construccion de un DFD paso a paso. [En línea] [Citado el: 16 de 03 de 2010.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100010/Lecciones/Cap5/HtasCASE.htm>.
 30. Martínez, Gerardo Moreno. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
 31. *Sistemas Operativos y Plataformas de Hardware Apropriadas*. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
 32. Entorno Virtual de Aprendizaje (Universidad de las Ciencias Informáticas). [En línea] [Citado el: 10 de 03 de 2010.] <http://teleformacion.uci.cu/mod/resource/view.php?id=11569>.
 33. *Introducoftware*. Habana : UCI, 2009.
 - 34 Propuesta de diseño para proyectos de desarrollo de software que utilizan symfony. [Citado el 5 de 05 de 2010.]. Habana: UCI, 2010.
-

Bibliografía.

1. Tecna. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.tecna.com>.
2. Inc, RCLAB S.R.L. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.rclabsl.com.ar/>.
3. Shlumberger. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.slb.com/>.
4. SAP. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.sap.com/spain/sme/whysap/industries/oil-gas/index.epx>.
5. Inc, Neuralog. [En línea] [Citado el: 14 de 12 de 2009.] http://www.neuralog.com/www_backup_files/espanol/index.htm.
6. (PLS), Petrokem Loggin Services. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.petrokemls.com/>.
7. InfOil. [En línea] [Citado el: 14 de 12 de 2009.] <http://www.infoil.com.ar/>.
8. cee. [En línea] [Citado el: 10 de 02 de 2010.] <http://cee.com/products/sampling/WellWizard/WellWizard.html>.
9. Free Download Manager. [En línea] [Citado el: 10 de 02 de 2010.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/..
10. WellSight. [En línea] [Citado el: 10 de 02 de 2010.] <http://www.wellsight.com>.
11. Peloton. Drilling & Well Data Software Solutions. [En línea] [Citado el: 10 de 02 de 2010.] <http://www.peloton.com/es/..>
12. Ramírez, Alexandra. [En línea] [Citado el: 11 de 03 de 2010.] http://s3.amazonaws.com/lcp/elida-nieves/myfiles/AS_Clase-231008-1.doc..
13. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. La Habana : Felix Varela, 2005.
14. Jacoboson, I., Booch, G., Rumbaugh J. *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000.
15. Cuevas, David Tavares. *ANÁLISIS Y DISEÑO DEL SISTEMA DE MANEJO INTEGRAL DE PERFORACIÓN DE POZOS*. Ciudad de la Habana : s.n., 2009.
16. Infosintesis Solutions Group, S.L. . Formación, asesoramiento y desarrollo de proyectos informáticos . [En línea] [Citado el: 13 de 03 de 2010.] www.infosintesis.net.
17. Suarez, Jose Gregorio. [En línea] 28 de 11 de 2009. [Citado el: 13 de 03 de 2010.] www.Los-patrones-de-diseno.htm.
18. UNAP. [En línea] 22 de 03 de 2004. [Citado el: 5 de 3 de 2009.] <http://www.unap.cl/~setcheve/Metrica/m/index.html>.

19. Sanchez, María A. Mendoza. [En línea] [Citado el: 15 de 03 de 2010.] http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.
20. Castillo, Danis Rego. *Análisis y Diseño del Sistema para el análisis petrofísico*. 2009.
21. [En línea] 2005. [Citado el: 15 de 03 de 2010.] <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
22. ProyectosAgiles.org. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.proyectosagiles.org/que-es-scrum>.
23. —. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.proyectosagiles.org/beneficios-de-scrum>.
24. José H. Canós, Patricio Letelier y M^a Carmen Penadés. [En línea] [Citado el: 16 de 03 de 2010.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
25. [En línea] 23 de 5 de 2008. [Citado el: 16 de 03 de 2010.] <http://crystalmethodologies.blogspot.com/>.
26. Java Mexico-Comunonad de Desarrolladores Mexicanos. [En línea] 27 de 02 de 2010. [Citado el: 16 de 03 de 2010.] http://www.javamexico.org/blogs/carraro/que_es_dsdm.
27. Huarachi, Maritza. *Ingernieria de Softeare*. EL ALTO – BOLIVIA : s.n., 2009.
28. INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA. *Herramientas Case*. 1999.
29. Construccion de un DFD paso a paso. [En línea] [Citado el: 16 de 03 de 2010.] <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4100010/Lecciones/Cap5/HtasCASE.htm>.
30. Martínez, Gerardo Moreno. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.monografias.com/trabajos5/insof/insof.shtml>.
31. *Sistemas Operativos y Plataformas de Hardware Apropriadas*. [En línea] [Citado el: 15 de 03 de 2010.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
32. Entorno Virtual de Aprendizaje (Universidad de las Ciencias Informáticas). [En línea][Citadoel:10 de 03 de 2010.] <http://teleformacion.uci.cu/mod/resource/view.php?id=11569>.
33. *Introduoftware*. Habana : UCI, 2009.
- 34 Propuesta de diseño para proyectos de desarrollo de software que utilizan symfony. [Citado el 5 de 05 de 2010.]. Habana: UCI, 2010.

Glosario de Términos

A

ArcView: Sistemas de información geográfica.

AutoCad: Programa de diseño asistido por ordenador para dibujo 2D y 3D.

B

C

CUPET: Unión Cuba-Petróleo.

CEINPET: Centro de Investigaciones de Petrónimo.

CAD: Estilo para el tratamiento de datos gráficos.

D

DBR: Brinda datos según sus reportes diarios los cuales son consultados en la toma de decisiones.

DST: Genera informes relacionados con la Dirección de Servicios Tecnológicos.

E

EPEPC: Empresa de Producción y Extracción de Petrónimo del Centro.

F

G

GeoQuest: Empresa que brinda productos y servicios de exploración.

H

I

IDE: Integrated Development Environment (Entorno Integrado de Desarrollo).

J

K

L

Landmark: Integración visual y orientada a la secuencia de tareas utilizando la misma visión 3D para todos los datos sísmicos y de pozo, interpretaciones, modelado y simulación de reservorios y planeamiento de perforaciones.

M

N

Ñ

O

P

Petrosoft: Polo Informático para el desarrollo de soluciones informáticas para la industria del petróleo.

Q

R

RUP: Proceso Unificado de Desarrollo. Metodología de desarrollo de software.

RCLAB: Laboratorio de estudios petrofísicos.

S

SIPP: Sistema de Manejo Integral de Perforación de Pozos.

Schlumberger: Empresa proveedora de productos y servicios para compañías petroleras.

T

TIC: Tecnologías de la Información y la Comunicación.

TECNA: Empresa asociada al estudio de proyectos vinculados a empresas petroleras.

U

UCI: Universidad de la Ciencias Informáticas.

V

W

WITSML: Wellsite Information Transfer Standard Markup Language. Lenguaje para marcar normas para transferir información del sitio del pozo.

X

Y

Z