

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



ANÁLISIS Y DISEÑO DEL SISTEMA DE MANEJO INTEGRAL DE PERFORACIÓN DE POZOS

Autor: David Tavares Cuevas

Tutor: Ing. Dennis Meriño Menadier

“Año del 50 Aniversario de la Revolución”

Ciudad de la Habana. 2009

Dedicatoria

Le dedico este trabajo:

- *A mi mamá por siempre estar ahí para mí y ser mi amiga y confidente en los momentos más difíciles.*
- *A mi papá, que mi guía y ejemplo a seguir, y haber luchado conmigo todos estos años para que llegue a ser alguien en la vida*
- *A mi abuelita, donde quiera que esté.*
- *A mami, siempre te recuerdo y me hubiera gustado que estuvieras aquí conmigo hoy.*
- *A mi tía, mi segunda madre, siempre pendiente de mi e incondicional.*
- *A mi hermano por haberme acompañado y siempre ser el punto que te hace recordar que es la verdadera familia.*
- *A mi abuelo, gracias por siempre quererme.*
- *A mi primo-hermano Yadier, gracias por ayudarme durante estos 5 años.*
- *A mis otros hermanos: Jose, Carlos y Loiret.*
- *A mis hermanas de la infancia: Sucel y Yenis M.*
- *A la gente de la cuadra.*
- *A Susy por el tiempo pasado, el presente y el futuro, gracias por darme la oportunidad de aprender de ti y contigo.*

- A los Samurays, los hermanos de toda la vida: Abel, Levian, Yoandri, Alberto, Erly, Sergito, Pepe, Adnys, Dunier, Carlos, Lazaro.
- A Bros-FEU: Frank, Yidian, Nilo, Osvaldo, Luiver, Krisna, y a todos los demás
- A mis nuevos Bros: Michel y Maikel, gracias por siempre ayudarme.
- A la gente de la aula, de la vieja y de la nueva: Ariagna, Belinda, Noraisy, Nieleidys, Kenia, Henrry, Ines, Alejandro, Alegna, Angel, Elisabeth, Gisel, Darlin, Manley, Norbelys, Karel, Douglas, Maite, Yaneisys, Pimienta, y todos los demás, muchas gracias por darme la posibilidad de ser su amigo.
- A mi viejos y fieles amigos: Elier, Alejandro, Jonny, Vladimir, Tarafa, Iduanis, Fabian, Lautaro.
- A todos mis profesores, gracias a sus enseñanzas he logrado llegar hasta aquí.
- A Saily, a pesar del corto tiempo, apareciste para enseñarme que detrás del final de algo bueno, puede comenzar algo mejor.
- A todas las personas que de he conocido en esta universidad, todas son especiales para mí.

Agradecimientos

Le agradezco:

- A mi tutor por haber confiado en mí y darme todo su apoyo en todo momento. Le deseo lo mejor en su vida personal y éxitos en su vida profesional.
- A mi papa por ser mi consultor personal.
- A los profes Armando y Yancy por siempre ayudarme.
- A Maikel y a Michel por ser incondicionales y sacrificarse en todo momento.
- A mi equipo de Trabajo: Milton, Roselí, Yelina, Reinel, DVD, HI, Jorge, Leo, Camilo, Aniuvis, Jordan , Noel y Eduardo.
- A la gente del Lab-103: Alejandro, Pimienta, Helimay y todos los que me ayudaron.
- Al profesor Febe, por contribuir en que este trabajo tenga la calidad que hoy tiene.
- Al Profesor Jorge Infante, por ayudarme en estos últimos días.

Resumen

La gestión y el control de la información manipulada por las empresas, constituye uno de los mayores problemas en el mundo actual, la información, unido al conocimiento de los especialistas, son factores que influyen principalmente en el desarrollo productivo de las mismas. Por tanto la eficiencia y calidad con que se tenga acceso a la información actualizada constituye un pilar fundamental en el propio desarrollo de las organizaciones.

El negocio de la Industria del Petróleo no escapa de este fenómeno, una industria donde se corren riesgos económicos millonarios, el acierto o desacierto de estas inversiones depende en gran medida en la cantidad de información que se tenga previa a la toma de decisiones cruciales para la vida de estas empresas.

La presente investigación consiste en el análisis y diseño de un sistema que permita la gestión y el control de la información manipulada desde los pozos de petróleo en perforación hacia la Dirección de Intervención y Perforación de Pozos (DIPP). Para ello se realiza un estudio exhaustivo de los procesos que generan información, que influyen en el correcto funcionamiento de los pozos, así como en la toma de decisiones en la DIPP. También se propone un análisis de las herramientas de desarrollo de software a utilizar, quedando finalmente plasmados los resultados del Análisis y Diseño del sistema propuesto.

Palabras Claves: Gestión, control, industria, petróleo, análisis y diseño.

Índice

Introducción	8
Capítulo I: Fundamentación Teórica	11
1.1 Introducción	11
1.2 Conceptos asociados al dominio del problema	11
1.3 Objeto de Estudio	12
1.4 Análisis de otras soluciones existentes.....	12
1.5 Análisis y Diseño.....	19
1.6 Patrones	21
1.7 Conclusiones Parciales	23
Capítulo II: Tendencias y Tecnologías Actuales.....	24
2.1 Introducción.....	24
2.2 Metodologías de Desarrollo de Software	24
Robustas.....	26
Ágiles.....	29
2.2.1 Fundamentación de la Metodología Utilizada.....	41
2.3 Lenguaje para Modelado de Objetos	43
2.4 Herramientas de CASE.....	43
2.5 Conclusiones Parciales	50
Capítulo III: Características del Sistema	51
3.1 Introducción.....	51
3.2 Modelo de Negocio.....	51
3.3 Requisitos de Software	69
3.4 Modelo de Sistema	75
3.5 Patrones de Casos de Uso	109
3.6 Complejidad del Sistema	111
3.7 Conclusiones Parciales	112
Capítulo IV: Análisis y Diseño del Sistema	113
4.1 Introducción	113
4.2 Análisis	113
4.3 Diseño	122
4.4 Patrones de Diseño	134
4.5 Conclusiones Parciales.....	136

Conclusiones	137
Recomendaciones	138
Bibliografía	143
Anexos.....	147
Glosario de Términos	158

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) ocupan un lugar fundamental en el desarrollo de la sociedad y la economía mundial, sin ellas la especie humana no hubiera, ni estuviera, desarrollándose de manera tan vertiginosa. El concepto de las TIC nace con la convergencia tecnológica de la electrónica, el software y las infraestructuras de las telecomunicaciones y proveen herramientas que ofrecen la posibilidad de encontrar soluciones novedosas ante los desafíos que se impone el hombre en su bregar diario.

La interacción de la tecnología con la sociedad, como factor de desarrollo humano, debe y tiene que contribuir a la solución de los problemas de la sociedad, en especial en los países y regiones más atrasadas. La brecha tecnológica, como se le conoce a las desigualdades en el uso de las TICs entre los países desarrollados y subdesarrollados, no está dada realmente por el uso en mayor medidas de las tecnologías en los países del primer mundo sino, por el injusto orden mundial que solo incentiva el enriquecimiento desmedido, sin pensar antes en las necesidades básicas que debe cumplir cada ser humano durante la vida; no es la tecnología la que hace mayor la distancia entre ricos y pobres, sino el uso que le dan los hombres. Por lo que se puede afirmar que la utilización de las TICs se presenta como una necesidad para el desarrollo económico y social de cualquier país.

En Cuba desde 1987 con la creación de los Joven Club de Computación idea impulsada por el Comandante en Jefe Fidel Castro Ruz, se inició el proceso de llevar a todas las esferas de la sociedad el uso de las TICs como medio de enseñanza, lo que se conoce como el paso inicial para el proceso de informatización de la sociedad cubana, en el cual se encuentra inmerso el país. Cuba ha desarrollado una política consecuente con el modelo social, donde el acceso a las tecnologías es un derecho del pueblo cubano.

Todas las esferas sociales y económicas de nuestro país están insertadas en este proceso, el cual está estrechamente vinculado al perfeccionamiento empresarial que lleva a cabo la economía cubana. Por tanto las entidades vinculadas a la Industria del Petróleo no están exentas de esto. CUPET es la empresa encargada de dirigir y controlar los diferentes procesos del petróleo en Cuba.

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. La DIPP(Dirección de Intervención y Perforación de Pozos), entidad única en el país, que dirige la perforación e intervención de pozos de petróleo 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, por lo que se identifica como **situación problemática**, el hecho de que en la actualidad el proceso de gestión, control y flujo de información se realiza manualmente, lo que trae consigo, redundancia en la información, así como errores y des actualización en la misma, además de que no existe un orden y control para el almacenamiento de la información.

Teniendo en cuenta la situación problemática planteada, se propone diseñar un sistema que permita gestionar la información generada 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 más rápido y simple a la información. Por tanto el **problema a resolver** sería la inexistencia de un sistema capaz de gestionar el flujo de información en los pozos de petróleo en perforación. Para la solución de este problema se realiza un análisis de los procesos de negocio en la gestión de información en los pozos de petróleo en perforación, lo que constituye el **objeto de estudio** del presente trabajo. Por consiguiente se define como **objetivo general** el Análisis y Diseño del Sistema de Manejo Integral de Perforación de Pozos.

Se plantea como **hipótesis** que si se logran determinar correctamente las principales funcionalidades, entonces se podrá modelar un sistema capaz de dar respuestas a ellas.

Para lograr el objetivo propuesto, se definen las siguientes **tareas de la investigación**:

1. Caracterizar a nivel mundial y nacional las soluciones que en la actualidad implementan y soportan los procesos de gestión de información en los pozos de petróleo.
2. Identificar las metodologías de desarrollo más utilizadas para sistemas de gestión de información en los pozos en perforación.
3. Identificar los procesos de negocio en cada uno de los pozos de petróleo en perforación y la DIPP.
4. Identificar las principales funcionalidades del sistema.
5. Modelar los casos de uso definidos como críticos en el sistema.

6. Diseñar los casos de uso definidos como críticos en el sistema.

Para un mejor entendimiento de este trabajo se decide estructurar el mismo de la siguiente manera:

- **Capítulo I:** Se expone el estado del arte, aspectos generales del negocio del petróleo. Al mismo tiempo se describe el objeto de estudio, los procesos fundamentales y además se abordan los aspectos teóricos referentes al Análisis y Diseño y el uso de patrones.
- **Capítulo II:** Se exponen las tendencias y tecnologías utilizadas para el modelado de la aplicación, además se justifican las herramientas, lenguaje de modelado y metodología utilizada.
- **Capítulo III:** Se describe el negocio y se hace el análisis del sistema a desarrollar. Se definen las funcionalidades del sistema y se describen detalladamente, además de la estimación de la complejidad del sistema a modelar.
- **Capítulo IV:** Se enfoca en la construcción de la solución mediante diagramas de clases de análisis y diseño, se plantean los principios para el diseño, así como los patrones utilizados.

Capítulo I: Fundamentación Teórica

1.1 Introducción

En este capítulo se expone los aspectos generales del negocio del petróleo, adentrándose más específicamente en el proceso de perforación de pozos y un análisis del estado del arte de sistemas que realizan funciones similares a las del que se propone en este trabajo. Además se abordan los aspectos teóricos referentes al Análisis y Diseño Orientado a Objetos, el enfoque según RUP y el uso de patrones.

1.2 Conceptos asociados al dominio del problema

- **CUPET:** Unión de Empresas de Cuba, dentro de la Industria del Petróleo.
- **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.
- **EPEPC:** Empresa de Producción y Extracción de Petróleo del Centro, entidad con la finalidad principal de regular la explotación de los yacimientos del centro del país.
- **Procesos de la Industria del Petróleo:** Exploración, Perforación, Producción, Refinación y Comercialización.
- **Perforación:** Proceso más importante dentro de la Industria del Petróleo, ya que es que proporciona al 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.
- **Pozo de Petróleo&Gas:** Entidad donde se realiza la perforación, la cual agrupa un gran número de entidades y trabajadores, como son las compañías de servicio de lodo, direccionales, contratistas, logísticas, entre otras.
- **DIPP:** Dirección de Intervención&Perforación de Pozos, única entidad en Cuba responsable de, como su nombre lo indica, de dirigir el proceso de perforación e intervención.
- **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)

- **Resumen de Tareas:** Donde se recogen las principales tareas y necesidades de los pozos de petróleo en perforación.
- **Reporte de Tiempo Perdido:** Recogen la actividad que se demoró, con el tiempo, y la empresa responsable.
- **Resumen por combustible:** Consumo diario en litro del combustible que utiliza y el pronóstico de cada pozo.
- **Resumen de las principales necesidades:** Las necesidades las piden por teléfono por la tarde sobre las 2:00 pm y llenan la planilla del Pedido de Logística, donde 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 am, 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 Ministra con un resumen de la semana.

1.3 Objeto de Estudio

El objeto de estudio de la presente investigación lo constituyen los principales Procesos de Negocio en la Gestión de Información en los Pozos de Petróleo en Perforación, enmarcado en el Sistema de Manejo Integral de Perforación de Pozos. La DIPP, institución encargada de dirigir el proceso de perforación en Cuba, no posee un sistema que gestione y controle la información generada en los pozos en perforación. Para obtener resultados satisfactorios se hace necesaria una modelación exitosa del sistema propuesto.

1.4 Análisis de otras soluciones existentes

En la actualidad existen una gama de producciones diferenciadas de software para el sector del Petróleo, encaminadas a resolver disímiles problemáticas dentro de esta esfera. A nivel internacional están muy desarrolladas las aplicaciones relacionadas con distintas fases de desarrollo del crudo.

La segunda etapa en la búsqueda de petróleo, es la perforación, proceso hacia el cual se encamina esta investigación acerca de los sistemas que se destinados a automatizar parte o totalmente este proceso. Durante la perforación se toman registros eléctricos que ayudan a conocer los tipos de formación y las características físicas de las rocas, tales como densidad, porosidad, contenidos de agua, de petróleo y de gas natural, por tal razón se

necesitan sistemas capaces de manejar grandes cantidades de información, sumamente sensible, para las compañías perforadoras(contratistas). Estos sistemas deben permitir almacenar toda esta información, de manera tal que el acceso a la misma sea rápido y efectivo, por lo cual los sistemas desarrollados poseen en un alto nivel de desarrollo y experticia, aplicando siempre las tecnologías más modernas para la construcción de software.

1. 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. (1). Ver Anexo 1

2. 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. (2)

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. (2)

3. STRATER

Es una potente e innovadora herramienta para el registro de pozos y el trazado de perforaciones. Con esta herramienta los geólogos ya no tendrán que invertir más tiempo y dinero para crear registros profesionales de pozos. Strater es una herramienta potente, sencilla de usar y con un precio muy asequible.

Con Strater se puede visualizar gráficamente:

- Profundidad o elevación
- Notas, comentarios y otros datos de texto
- Petrología, % petrología y descripciones litológicas.
- Potencial espontáneo, rayos gamma, calibrador, neutrónica, DST, densidad aparente, resistividad, ritmo de perforación, gas total, calidad de los gases y datos sínicos
- Contabilidad de impactos, número y tipo de muestras, permeabilidad, RQD, lecturas OVM, %recuperación
- Concentración de contaminantes, contenido de humedad y detalles de construcción de pozos.
- Datos de ensayos, petrología de mineralización o alteración, valores BTU (Unidades Térmicas Británicas) y datos de contenido de cenizas.
- Virtualmente cualquier tipo de dato de profundidad o intervalo

Strater ofrece una flexibilidad insuperable para el diseño y formateo de registros. Su avanzada interfaz de usuario permite que el diseño y la visualización de sus datos sean más fáciles que nunca. Strater incluye 13 tipos de registro muy usuales para visualizar sus datos

gráficamente: profundidad, línea/símbolo, gráfico cruzado, petrología, barras de zona, barras, porcentajes, postes, postes por clase, gráficos, textos complejos, y registros de construcción de pozos. Cada uno de los registros puede ser modificado para ajustarse a las necesidades del usuario. Entre algunas de las personalizaciones posibles podemos citar:

Visualizar registros basados en profundidad o elevación

- Visualizar profundidad y/o líneas de rejilla variable
- Añadir barras de escala y títulos
- Configurar diferentes estilos de líneas de contacto entre unidades litológicas
- Utilizar curvas para crear perfiles litológicos como perfiles de desgaste
- Crear registros continuos sencillos o de múltiples páginas
- Formatear diferentes tipos de registro para obtener presentaciones lo más informativas posible

Strater proporciona varias funcionalidades para simplificar la tarea de importar datos, crear el diseño de la perforación exacta que el usuario requiere y obtener la salida en el formato necesario, ya sea impreso o exportado a formato electrónico para incluir en un informe o presentación. Su robusta base de datos que soporta importación de datos de muchas fuentes diferentes, incluyendo ficheros de texto ASCII, ficheros LAS, ficheros XLS y prácticamente cualquier fichero de base de datos. El usuario puede importar datos referentes a múltiples perforaciones en una tabla de datos e importar múltiples tablas de datos en un proyecto. También puede crear múltiples vistas de perforaciones en un proyecto. Cada vista de perforación puede contener un diseño de perforación diferente, permitiendo así una flexibilidad inigualable en la presentación de sus datos. El almacenamiento de todas las vistas y tablas de datos de las perforaciones en un único proyecto mantiene toda la información fácilmente accesible.

La edición y manipulación de objetos con Strater es muy fácil. Sólo hay que seleccionar algún objeto de la vista de perforación mediante el Gestor de Objetos (Object Manager). El Object Manager organiza de forma intuitiva y conveniente todos los objetos en una vista en forma de árbol jerárquico. Una vez se ha seleccionado el objeto, editar todas sus propiedades es sencillo gracias al nuevo Property Inspector. Además cualquier cambio se puede ver de forma inmediata. Su versatilidad y facilidad de uso le hace un software ideal para numerosas industrias, entre las que se incluyen las petrolíferas, geofísicas, mineras, medioambientales, geotécnicas y muchas otras.

Software como el que se va a desarrollar por nuestro equipo de trabajo existen muchos en el mundo pero han sido desarrollados la mayoría en plataformas privativas lo cual imposibilita

el uso del mismo para empresas que de cierta forma no pueden acceder a dicho producto.
(3)

4. WellView

WellView® es un archivo de pozo corporativo completo. Desde la petición de perforar un pozo hasta el abandono, registra todos los cambios y operaciones a través del ciclo de vida del pozo. Con su poderoso estado mecánico, reportes y sistemas de búsqueda, WellView coloca la información en las manos de la gente que mas la necesita sin las limitantes de archivos de papel tradicionales o reportes diarios.

Peloton comenzó el desarrollo de WellView en 1991, enfocados en las operaciones de producción del pozo y estado mecánico histórico. Para abarcar mas en el ciclo de vida de un pozo, Peloton expandió este software para manejo de datos de perforación en 1997. Nuestra versión actual crece con la fortaleza de versiones pasadas mejorando el modelo de datos, herramientas de administración y capacidades analíticas.

El estado mecánico de WellView le permite visualizar los equipos en el pozo y eventos (lecciones aprendidas, fallas, etc.). Esto le provee una historia visual del pozo. La base de datos de WellView captura todos los datos relacionados con el pozo y sus operaciones así como del agujero y equipo de superficie.

Trabajando con MasterView® de Peloton, WellView es parte de una solución con análisis de ciclo de vida de pozo integrado y capacidad de visualización.

Administración de Datos de Perforación - Personalice y actualice los reportes diarios rápidamente, administre y encuentre datos del pozo, visualice datos de fondo de hoyo, analice datos de perforación y su ejecución, produzca órdenes de trabajo desde la información de costo en WellView, analice datos de tiempo.

Administración de Datos de Operaciones de Pozo - Personalice y actualice los reportes diarios rápidamente, administre y encuentre datos del pozo, vea los estados mecánicos del hoyo y las intervenciones del pozo, modifique y organice el archivo electrónico del pozo, revise ejecución, costo y fallas.

Solución WellView-Pason - La solución WellView Pason es una colección de datos de perforación y un sistema de reportes implementado como una adición integrada al Grabador de Perforación Electrónico (EDR). (4)

Peloton. La Empresa

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

Hasta la fecha, Peloton se ha auto-financiado y ha administrado su crecimiento cuidadosamente mientras mantiene los estándares de calidad que son fundamentales para nuestro futuro. Hemos crecido de una pequeña operación de dos personas en 1995 a sesenta personas con oficinas en Aberdeen, Ámsterdam, Calgary, Houston, Yakarta, Moscú y Dubái. (4)

Todos estos sistemas estudiados, resultan sumamente competitivos y probados a nivel mundial, tienen como características comunes el hecho que son softwares sobre la plataforma de escritorio, bajo licencias de software propietario. Todas estas empresas poseen una alta experiencia en la construcción de sistemas para la industria del petróleo, con alto grado de experticia sobre los diferentes procesos de la industria petrolera.

Partiendo del estudio realizado anteriormente, se arriban las siguientes conclusiones:

- Estos sistemas pueden darle respuesta parcial al [negocio](#), que presenta la entidad cliente del sistema que propone este trabajo.
- Para utilizar estos software, el país se debe pagar la licencia, lo cual implica un gasto de divisas.
- Estos sistemas pertenecen a empresas, Canadienses y de EE.UU, y debido a la política del bloqueo económico aplicado por EE.UU a nuestro país, la adquisición,

de alguno de éstos puede tener un costo mucho más alto que lo normal o que simplemente sea imposible adquirirlos.

- Nos enfrentamos a un negocio complicado que para lograr sistemas que sean comercializables a nivel global, debemos antes todo lograr un alto nivel de capacitación referente a la industria y alta confiabilidad en los clientes nacionales.

1.5 Análisis y Diseño

1.5.1 Análisis y Diseño Orientado a Objetos

El Análisis Orientado a Objetos (AOO) se basa en conceptos: objetos, atributos, clases y miembros, todos y partes. El porqué nos ha llevado tanto tiempo aplicar estos conceptos al análisis es una interrogante aun por responder. Se basa en cinco principios básicos: 1. Modela el Dominio de la Información 2. Describe la Función 3. Se representa el comportamiento 4. Los modelos de datos, funcional y de comportamiento dividen para mostrar más detalles 5. Los modelos iniciales representan la esencia del problema mientras que los últimos aportan detalles de la implementación; siendo así el propósito del AOO es definir todas las clases que son relevantes al problema a resolver, las operaciones, atributos, relaciones, y comportamientos asociados al mismo.

Luego de culminar el AOO, este transforma y da paso a Diseño Orientado a Objetos (DOO). El DOO es bastante difícil y el diseño reusable lo es aún más. Se deben definir los objetos, clasificarlos dentro de las clases en la granularidad correcta, definir interfaces de clases, jerarquías de herencia y establecer relaciones clave entre ellos. El DOO debe ser específico al problema que se tiene, pero suficientemente general para adaptarse a problemas y requerimientos futuros, además deberá evitar el rediseño o al menos minimizarlo. En el caso del diseño reusable y flexible es difícil, si no imposible de obtener de una sola vez.

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, 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. (5)

1.5.2 Análisis y Diseño según RUP

El flujo de trabajo de Análisis y Diseño que define la Metodología RUP (Ver acápite 2.2), contenido dentro de la Fase de Elaboración principalmente, tiene como objetivos transformar los requerimientos en un diseño de cómo va a ser implementado el sistema, evolucionar hacia una arquitectura del software robusta y adaptar el diseño para que coincida con el ambiente de implementación, diseñando el sistema con un enfoque hacia el rendimiento. A pesar de que RUP agrupa el Análisis y el Diseño en un solo flujo, no se puede ver como procesos totalmente ligados, ya que cada uno de ellos genera artefactos diferentes, en el caso del Análisis el Modelo de Análisis y en el Diseño el Modelo de Diseño, pero para profundizar más sobre las características se presenta la siguiente tabla comparativa: (6)

Modelo de Análisis	Modelo de Diseño
Modelo conceptual, porque es una abstracción del sistema y permite aspectos de la implementación	Modelo físico, porque es un plano de la implementación
Genérico respecto al diseño(aplicable a varios diseños)	No genérico, específico para una implementación
Tres estereotipos conceptuales sobre las clases: Interfaz, Control y Entidad	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	Da forma al sistema mientras que intenta preservar la estructura definida por el modelos de análisis lo más posible

1.6 Patrones

“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”.

Christopher Alexander

Los Patrones son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de software. También se pueden ver como una descripción de un problema y la solución a la que le da el nombre y que se puede aplicar en nuevos contextos. Existen varias definiciones de que es un patrón:

- Un patrón es 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. (7)
- Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. (8)
- Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas (9)

Pero los patrones no aparecieron por simple arte del destino, el porqué viene dado por:

- El desarrollo de software es una tarea complicada, la cual depende en gran medida de la experiencia de las personas involucradas, en particular de los desarrolladores. El 80% de los aportes vienen del 20% del personal
- La comprensión del software es uno de los problemas más complicados en la tarea de mantenimiento y evolución
- El hombre durante su historia ha dominado cierta técnica pasando por un proceso
 - Se realizan las operaciones de una manera artesanal. Los expertos aprenden por un proceso de ensayo y error y por transmisión de otros expertos
 - Se crea una ciencia alrededor de la tarea
 - Se desarrollan las técnicas generalmente aceptadas en el área
 - Se logra un conocimiento común sobre cómo aplicar las técnicas. Hay un metalenguaje común entre los expertos

- La sociedad requiere sistemas más complejos y más grandes. Los recursos para desarrollarlos cada vez son más escasos... Debe existir un mecanismo de reutilización. El 80% del esfuerzo esta en el 20% del código desarrollado

El objetivo principal del uso de los patrones está dado por crear un lenguaje común a una comunidad de desarrolladores para comunicar experiencia sobre los problemas y sus soluciones.

El desarrollo de los patrones posee antecedentes desde lo años 70 por Christopher Alexander. Este conocimiento es trasportado al ámbito del desarrollo de software orientado por objetos y se aplica al diseño. (7)

Generalmente, cuando se habla de patrones, siempre se escucha hablar de "Patrones de Diseño", ¿hay otros tipos de patrones?

Hoy, se puede encontrar literalmente miles de patrones definidos. Resulta imposible para un programador conocerlos todos, ni mucho menos probarlos o valorarlos. Así que necesitamos una clasificación sencilla (aunque incompleta) para al menos poder situarlos, reduciéndolo sólo al ámbito del software se tiene:

- **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.

Realizando un acercamiento a los patrones de diseño se puede agrupar los patrones de diseño según el problema que abordan:

- **Creacionales:** Abordan problemas de creación de objetos. Entre estos patrones están la factoría, la factoría abstracta, el singleton, etc.

- **Estructurales:** Abordan problemas sobre relaciones entre entidades. Entre ellos: la fachada, el adaptador, el decorador, etc.
- **De Comportamiento:** Abordan problemas de comunicación entre entidades. Por ejemplo observador, estrategia, plantilla, etc. (10)

En los capítulos siguientes se aborda específicamente los patrones de Casos de Uso y de Diseño utilizados para el modelado del sistema propuesto.

1.7 Conclusiones Parciales

En este capítulo se han abordado los conceptos asociados al problema, además se aborda con más amplitud el objeto de estudio. Se realiza un estudio del arte, de los sistemas similares al que se propone en este trabajo, además se realiza un resumen referente al Análisis y Diseño orientado a Objetos, así como los fundamentos teóricos del Análisis y el Diseño según RUP. Además se realiza un breve acercamiento a los aspectos teóricos sobre patrones.

Capítulo II: Tendencias y Tecnologías Actuales

2.1 Introducción

En este capítulo se abordaran las metodologías de desarrollo de software más utilizadas en el mundo en este tipo de sistemas, mas la fundamentación de porque la utilización de RUP para el desarrollo de este sistema. Además del Lenguaje y Herramienta de modelado, Tecnologías, Arquitectura así como Lenguaje de programación y Herramientas de desarrollo que deben ser utilizadas para la futura implementación del sistema.

2.2 Metodologías de Desarrollo de Software

Un proceso define quién está haciendo qué, cuando y como alcanzar un determinado objetivo. (6)

Desde que se inició el proceso de construcción de software, las empresas encargadas de la producción de software a nivel global o local, comenzaron la búsqueda de formulas para que los sistemas solicitados, tuvieran la calidad requerida, cumplieran las especificaciones del cliente y fueran económicamente factibles. Este proceso trajo consigo la aparición de un nuevo concepto, *Metodologías de Desarrollo de Software*.

Según *Alistair Cockburn*, una metodología se compone de partes interconectadas:

- Roles: arquitecto, programador ...
- Habilidades (utilización de determinadas herramientas, planificación ...)
- Técnicas (programación Java, refactorización, CRC Cards)
- Equipos
- Herramientas (para una técnica determinada o realizar un entregable que cumpla el estándar)
- Entregables (casos de uso, diagramas de clases ...)
- Estándares (normativas de notación, codificación, convenciones...)
- Actividades
- Calidad (aspectos a gestionar y tener en cuenta en los entregables)

[Ver Anexo 2](#)

Desde otra perspectiva, *Graham, Brian Henderson-Sellers, y Houman Younessi*, exponen en su *The OPEN Process Specification* que toda metodología tiene que cubrir el ciclo de vida completo y proporcionar como mínimo:

- Un proceso de ciclo de vida completo.

- Un conjunto de conceptos y modelos.
- Un conjunto completo de técnicas (reglas, guías de estilo ...).
- Un conjunto delimitado de entregables.
- Un lenguaje de modelado de notación intuitiva.
- Un conjunto de métricas.
- Un conjunto de técnicas para asegurar la calidad.
- Estándares de codificación.
- Guías para la gestión de los proyectos.
- Un conjunto de herramientas de soporte.

Existen dos extremos diferenciados de metodología según Highsmith: el extremo de la burocracia, con procesos rígidos y definidos al máximo nivel de detalle, y la adhocracia, con un desarrollo caótico sin ningún proceso definido. El nivel máximo de burocracia no es aceptado de buen grado por los integrantes de la organización, mientras que el nivel de adhocracia tiene un alto riesgo de fracaso.

Si ponemos en una escala del 1 al 8 el nivel de burocracia: [Ver Anexo 3](#)

Según el gráfico, el modelo en cascada es el más burocrático de todos, mientras que la codificación (propia de programadores rebeldes que no siguen ningún proceso definido) se encuentra en la denominada adhocracia o caos. (11)

Aglutinando estos y otros conceptos podemos resumir que una metodología de desarrollo no es más que el conjunto de métodos que se utilizan para desarrollar una actividad con el objetivo de formalizarla y optimizarla (12).

Por tal razón en la actualidad existen varias metodologías de desarrollo, las cuales se dividen en dos categorías, robustas (pesadas, también llamadas tradicionales) y ágiles.

[Ver Anexo 4](#)

Existen varios modelos de desarrollo y cada metodología adopta el suyo propio, ya puede ser tomando uno solo o combinándolos y/o adaptándolos a las condiciones de la entidad, logrando un nuevo modelo. Los modelos existentes en la actualidad son:

- Modelo en Cascada
- Modelo Basado en Prototipos
- Modelo Incremental o Evolutivo

- Modelo Espiral
- Modelo Orientado a Objetos(OO)
- Modelo Cascada con Sub-Proyectos
- Modelo Entrega por Etapas. (13)

Como bien se expresa anteriormente en este documento 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), Ágiles y de Doble Personalidad.

Robustas

Métrica V3

Es una metodología de desarrollo elaborada por el *Consejo Superior de Informática del Ministerio de Administraciones Públicas*. (14)

La metodología MÉTRICA Versión 3 ofrece a las Organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software dentro del marco que permite alcanzar los siguientes objetivos:

- Proporcionar o definir Sistemas de Información que ayuden a conseguir los fines de la Organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la Organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de Sistemas y Tecnologías de la Información y las Comunicaciones, permitiendo una mayor capacidad de adaptación a los cambios y teniendo en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos software obtenidos.

La nueva versión de MÉTRICA contempla el desarrollo de Sistemas de Información para las distintas tecnologías que actualmente están conviviendo y los aspectos de gestión que aseguran que un Proyecto cumple sus objetivos en términos de calidad, coste y plazos.

Su punto de partida es la versión anterior de MÉTRICA de la cual se han conservado la adaptabilidad, flexibilidad y sencillez, así como la estructura de actividades y tareas, si bien las fases y módulos de MÉTRICA versión 2.1 han dado paso a la división en Procesos, más adecuada a la entrada-transformación-salida que se produce en cada una de las divisiones del ciclo de vida de un proyecto. Para cada tarea se detallan los participantes que intervienen, los productos de entrada y de salida así como las técnicas y prácticas a emplear para su obtención.

En la elaboración de MÉTRICA Versión 3 se han tenido en cuenta los métodos de desarrollo más extendidos, así como los últimos estándares de ingeniería del software y calidad, además de referencias específicas en cuanto a seguridad y gestión de proyectos.

También se ha tenido en cuenta la experiencia de los usuarios de las versiones anteriores para solventar los problemas o deficiencias detectados.

En una única estructura la metodología MÉTRICA Versión 3 cubre distintos tipos de desarrollo: estructurado y orientado a objetos, facilitando a través de interfaces la realización de los procesos de apoyo u organizativos: Gestión de Proyectos, Gestión de Configuración, Aseguramiento de Calidad y Seguridad.

La automatización de las actividades propuestas en la estructura de MÉTRICA Versión 3 es posible ya que sus técnicas están soportadas por una amplia variedad de herramientas de ayuda al desarrollo.

Además, para facilitar la utilización de MÉTRICA Versión 3 se ha desarrollado una herramienta software, Gestor Metodológico, de ayuda a la aplicación de la metodología en cada proyecto concreto y que permite adaptar la estructura de MÉTRICA Versión 3 de acuerdo a las características del mismo, permitiendo el seguimiento y control de sus actividades y tareas realizadas por distintos perfiles de usuario asignados a los participantes por el jefe de proyecto

Se ha desarrollado también un software, Selector de Herramientas, que ayuda a seleccionar entre las CASE del mercado la que mejor se adapta a las necesidades de cada proyecto teniendo en cuenta las características de cada organización. (15)

Todo proyecto Métrica v3 consta de un conjunto de fases que se desglosan en múltiples puntos cuya cronología hay que seguir con claridad para ir avanzando en el desarrollo del proyecto. El siguiente gráfico muestra el esquema general de estas fases: [Ver Anexo 5](#) (16)

Métrica debería ser obligatoria en cuantos contratos de desarrollo de software realiza la administración pública, sin embargo su utilización aún no está suficientemente universalizada. Todavía es frecuente ver concursos públicos donde no se establece la obligatoriedad para el contratante de seguir las pautas de métrica a la hora de llevar a cabo un proceso de construcción de software. (14)

MERISE

Las bases de MERISE comenzaron en 1.972 por un equipo universitario de ingenieros de Aix-en-Provence. La primera versión salió a finales de 1.976.

El proyecto partió del Centre Technique Informatique del Ministerio de Industria Francés en Septiembre de 1.977, para cubrir las necesidades tanto de la administración como de las empresas. El proyecto finalizó en mayo de 1.978 dando lugar a MERISE como metodología de Análisis y Diseño de Sistemas de Información.

Esta metodología aporta un ciclo de vida más largo a los existentes hasta entonces que se materializa en un conjunto definido de etapas. Introducen dos ciclos complementarios: ciclo de abstracción y ciclo de decisión. El ciclo de abstracción se basa en la percepción de tres niveles de abstracción: conceptual, organizativo y físico. Además se definen dos niveles para cada nivel: un modelo de datos y otro de tratamientos. (17)

Las fases de la metodología MERISE son:

1. Estudio preliminar.
2. Estudio detallado.
3. Implementación.
4. Realización y puesta en marcha.

SSADM v4 (Structures Systems Analysis and Design Method)

El gobierno británico plantea la necesidad de crear una metodología y se desarrolló entre el Central Computing and Telecommunications Agency (CCTA) y Learmonth and Burchett Management Systems (LBMS), dando como resultado la metodología SSADM.

Los aspectos claves de SSADM v 4 son:

- Énfasis en los usuarios: sus requisitos y participación.
- Definición del proceso de producción: qué hacer, cuándo y cómo.
- Tres puntos de vista: datos, eventos, procesos.
- Máxima flexibilidad en herramientas y técnicas de implementación.

SSADM proporciona un conjunto de procedimientos para llevar a cabo el análisis y diseño, pero no cubre aspectos como la planificación estratégica ni entra en la construcción del código. (17)

Estructura (Fases del Proceso): (18)

1. Estudio de Factibilidad
2. Análisis de Necesidades
3. Especificación de Requisitos
4. Lógica del Sistema. Especificación
5. Diseño Físico

El uso de un proceso de desarrollo muy riguroso se ha mostrado efectivo en proyectos de gran tamaño, pero en entornos muy cambiantes donde se exigen tiempos de desarrollo muy ajustados y con una buena calidad, estas metodologías no cumplen los objetivos esperados. A causa de estas restricciones, los equipos de desarrollo suelen resignarse y prescindir del uso de la metodología. Para llenar este vacío, las metodologías ágiles son una solución que permiten simplificar el proceso con el mismo objetivo de calidad, costes y tiempo que los procesos tradicionales. Esta filosofía tiene como objetivo la satisfacción del cliente, la obtención de entregas incrementales de software de forma prematura, equipos pequeños muy motivados, métodos informales, y simplicidad en el desarrollo. (11)

Ágiles

ExtremeProgramming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP en [2] sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

- **Las Historias de Usuario**

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

- **Roles XP**

1. Programador
2. Cliente
3. Encargado de pruebas (*Tester*)
4. Encargado de seguimiento (*Tracker*)
5. Entrenador (*Coach*)
6. Consultor
7. Gestor (*Big boss*)

- **Proceso XP**

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede

entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

- **Fase I: Exploración**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- **Fase II: Planificación de la Entrega**

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Esta fase dura unos pocos días.

- **Fase III: Iteraciones**

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.

- **Fase IV: Producción**

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

- **Fase V: Mantenimiento**

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.

- **Fase VI: Muerte del Proyecto**

Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

- **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.

1. El juego de la planificación
2. Entregas pequeñas
3. Metáfora
4. Diseño simple
5. Pruebas
6. Refactorización (*Refactoring*)
7. Programación en parejas
8. Propiedad colectiva del código.
9. Integración continua
10. 40 horas por semana
11. Cliente in-situ
12. Estándares de programación
13. Comentarios respecto de las prácticas (19)

Relación entre las Prácticas XP. [Ver Anexo 8](#)

Scrum

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitivo). Cada ciclo o iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Las iteraciones en general tienen una duración entre 2 y 4 semanas. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

En Scrum, el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la gestión de un proyecto Scrum se focaliza en definir cuales son las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos que sea posible.

Scrum tiene un conjunto de reglas muy pequeño y muy simple y está basado en los principios de inspección continua, adaptación, auto-gestión e innovación. El cliente se entusiasma y se compromete con el proyecto dado que ve crecer el producto iteración a iteración y encuentra las herramientas para alinear el desarrollo con los objetivos de negocio de su empresa. Por otro lado, los desarrolladores encuentran un ámbito propicio para desarrollar sus capacidades profesionales y esto resulta en un incremento en la motivación de los integrantes del equipo. (20)

Como funciona Scrum: [Ver Anexo 6](#) (21)

Los principales beneficios que proporciona Scrum son:

- Entrega mensual (o quincenal) de resultados (los requisitos más prioritarios en ese momento, ya completados) lo cual proporciona las siguientes ventajas:
- Gestión regular de las expectativas del cliente y basada en resultados tangibles.
- Resultados anticipados (*time to market*).

- Flexibilidad y adaptación respecto a las necesidades del cliente, cambios en el mercado, etc.
- Gestión sistemática del Retorno de Inversión (ROI).
- Mitigación sistemática de los riesgos del proyecto. (22)

DSDM

DSDM por sus siglas en inglés (**Dynamic System Development Method**), es un método que nace al igual que por decir algo el Método Espiral de un modelo antecesor heredando sus ventajas, desventajas y características principales, este Método de Desarrollo para Sistemas Dinámicos se basó en el modelo RAD (**Rapid Application Development**), en esta investigación conoceremos acerca de Requisitos Previos para el uso de DSDM, el Ciclo de vida del proyecto, Core Técnicas de DSDM, Funciones de DSDM entre otras. (23)

Y así el DSDM consortium liderado por Tony Mobbs, Jennifer Stapleton, Gary Hodsdon, Paul Herzlich y Peter Constable, publicó en Febrero del 95 la 1ª versión de DSDM. Las siguientes versiones, es decir, la 2ª en el 96, la 3ª en el 97 y la 4ª en el 2001. Mejoraron mucho gracias al énfasis que se puso en obtener feedback de los usuarios. La versión actual es la 4.1 y es el método más usado en el Reino Unido y comienza a extenderse por Europa y Estados Unidos.

Se centra en ayudar a la gente del equipo de desarrollo y a los usuarios a trabajar juntos de forma más eficiente y así evitar producir sistemas que no cumplan los requerimientos, no funcionen correctamente o que caigan en desuso. DSDM es un proceso iterativo e incremental donde en cada paso se implementan los requerimientos que son más importantes en el momento actual para la organización. Así se consigue satisfacer a tiempo los verdaderos requerimientos del negocio.

Principios Fundamentales

1. Participación del usuario activo
2. El equipo toma decisiones. Esto es necesario para motivar al equipo y mantener una alta velocidad de desarrollo.
3. Frecuentes entregas del producto.
4. Ajustarse a los objetivos del negocio.
5. Desarrollo iterativo e incremental.
6. Cambios reversibles.

7. Especificar requerimientos globales.
 8. Pruebas integradas durante todo el ciclo de vida.
 9. La colaboración y cooperación entre el equipo, usuarios y stakeholders es esencial.
- (24)

Crystal Methodologies

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn.

El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros). (25)

3-8	10-20	25-50	50-100	100-200	200-500	800+
-----	-------	-------	--------	---------	---------	------

Crystal aconseja que el tamaño del equipo sea reducido (Pocos componentes).

La mejora de la comunicación entre los miembros del equipo del proyecto:

- Mismo lugar de trabajo → Disminuye el coste de la comunicación
- Mejora Individual → Mejora global del equipo (26)

Existen dos reglas comunes que todos las metodologías Crystal deben seguir:

1. El proyecto debe utilizar el modelo incremental, con iteraciones de 4 meses o menos, preferentemente de 1 a 3 meses
2. El equipo de trabajo deben asegurar

Otras de la restricciones de la metodología Crystal es que el equipo de trabajo este ubicado en un solo lugar (dirección), así esta metodología no aplica la distribución de equipos en diferentes localizaciones, ya que la principal premisa de Crystal según Cockburn es a diferencias de otras que el equipo de trabajo esté todo en una misma localización. (27)

Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:

- Aspecto humano del equipo
- Tamaño de un equipo (número de componentes)
- Comunicación entre los componentes
- Distintas políticas a seguir
- Espacio físico de trabajo (26)

ASD

Adaptive Software Development, su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. (19)

- Especulación
 1. Inicio para determinar la misión del proyecto.
 2. Determinación del marco temporal del proyecto.
 3. Determinación del número de iteraciones y la duración de cada una.
 4. Determinación del objetivo de cada una.
 5. Asignación de funcionalidad a cada iteración.
- Colaboración
 1. Desarrollo concurrente del trabajo de construcción y gestión del producto
- Aprendizaje

En cada iteración se revisa:

 1. Calidad, con criterios de cliente.
 2. Calidad, con criterios técnicos.
 3. Funcionalidad desarrollada
 4. Estado del proyecto

No proporciona el tipo de prácticas detalladas como lo hace XP, pero proporciona la base fundamental de por qué el desarrollo adaptable es importante y las consecuencias a los más profundos niveles de la empresa y la gerencia.

Ciclo de Vida. [Ver Anexo 7](#)

Las características básicas de ASD son:

- Trabajo orientado y guiado por la misión del proyecto.
- Basado en la funcionalidad
- Desarrollo iterativo
- Desarrollo acotado temporalmente
- Guiado por los riesgos
- Trabajo tolerante al cambio. (28)

Doble Personalidad

Esta terminología de Doble Personalidad no se encuentra estandarizada para caracterizar o tipificar a las metodologías de desarrollo actualmente, se utiliza para definir a metodologías que son altamente adaptables, ya que poseen comportamientos y componentes de metodologías ágiles y pesadas.

Microsoft Solutions Frameworks (MSF)

Microsoft Solutions Framework (MSF) es una flexible e interrelacionada serie de conceptos, modelos y prácticas de uso que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos, se ha convertido posteriormente en un modelo práctico que facilita el éxito de los proyectos tecnológico

MSF se compone de varios modelos y disciplinas encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto:

- Modelo de Arquitectura del Proyecto
- Modelo de Equipo [Ver Anexo 11](#)
- Modelo de Proceso
- Modelo de Gestión del Riesgo
- Modelo de Diseño de Proceso
- Modelo de Aplicación (29)
- Disciplina de Administración de Riesgos
- Disciplina de Administración de Proyectos (30)

Principios

1. Promover comunicaciones abiertas
2. Trabajar para una visión compartida
3. Fortalecer los miembros del equipo
4. Establecer responsabilidades claras y compartidas
5. Focalizarse en agregar valor al negocio
6. Permanentemente ágil, y esperar los cambios
7. Invertir en calidad
8. Aprender de todas las experiencias (31)
9. Partners con clientes.
10. Siempre crear productos entregables. (30)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Todo proyecto es separado en cinco principales fases: [Ver Anexo 12](#)

Visión y Alcances: La fase de visión y alcances trata uno de los requisitos más fundamentales para el éxito del proyecto, la unificación del equipo detrás de una visión común. El equipo debe tener una visión clara de lo que quisiera lograr para el cliente y ser capaz de indicarlo en términos que motivarán a todo el equipo y al cliente. Se definen los líderes y responsables del proyecto, adicionalmente se identifican las metas y objetivos a alcanzar; estas últimas se deben respetar durante la ejecución del proyecto en su totalidad, y se realiza la evaluación inicial de riesgos del proyecto.

Planificación: Es en esta fase es cuando la mayor parte de la planeación para el proyecto es terminada. El equipo prepara las especificaciones funcionales, realiza el proceso de diseño de la solución, y prepara los planes de trabajo, estimaciones de costos y cronogramas de los diferentes entregables del proyecto.

Desarrollo: Durante esta fase el equipo realice la mayor parte de la construcción de los componentes (tanto documentación como código), sin embargo, se puede realizar algún trabajo de desarrollo durante la etapa de estabilización en respuesta a los resultados de las pruebas. La infraestructura también es desarrollada durante esta fase.

Estabilización: En esta fase se conducen pruebas sobre la solución, las pruebas de esta etapa enfatizan el uso y operación bajo condiciones realistas. El equipo se enfoca en priorizar y resolver errores y preparar la solución para el lanzamiento.

Implantación: Durante esta fase el equipo implanta la tecnología base y los componentes relacionados, estabiliza la instalación, traspasa el proyecto al personal soporte y operaciones, y obtiene la aprobación final del cliente. (32)

Resumiendo:

- Microsoft Solution Framework versión 4.0 es el último framework de la serie de basado en las mejores prácticas de Microsoft y sus clientes.
- Presenta un metamodelo descriptivo y dos templates de procesos prescriptivos que implementan MSF 4.0 en Team System.
- Juntas son herramientas muy poderosas para equipos de todo tipo. MSF proporciona la guía de proceso y Team System un conjunto de herramientas integradas y extensibles. (30)

Proceso Unificado de Desarrollo (RUP)

RUP pertenece a la familia de metodologías "pesadas", a causa de la gran cantidad de procesos y documentación que requiere. Se ha convertido en la metodología estándar "de facto" para el desarrollo de proyectos en entornos orientados a objetos. RUP es un proceso iterativo e incremental, el que facilita que sea un proceso planificado y gestionado, que:

- Se adapta a los cambios de los requerimientos con pocas alteraciones.
- Involucra al usuario/cliente durante el proceso.
- Permite detectar y gestionar los riesgos durante todo el ciclo de vida del proyecto.
- Se basa en la construcción de prototipos ejecutables. (11)

Además de las características propias de los procesos iterativos se caracteriza por:

- Está dirigido durante el proceso por los casos de uso
- Centrado en la arquitectura
- Proporciona una guía de cómo usar de forma efectiva UML.
- Es un proceso configurable. (6)

RUP es una metodología muy extensa y en la mayoría de casos en el momento de su implantación se considera un proceso demasiado costoso para la cantidad de actividades y entregables que se definen. Hay que tener en cuenta que no estamos obligados a hacer uso de todas las actividades y entregables definidos, sino que podemos configurar nuestro proceso para adaptarnos únicamente a aquellas partes que se consideran necesarias.

El proceso RUP se repite en una serie de ciclos. Cada ciclo concluye con una versión del producto (release) y cada ciclo está dividido en 4 fases: inicio, elaboración, construcción y transición. Cada una de las fases está a su vez dividida en iteraciones, y en cada iteración se realizan 5 procesos o etapas principales: requerimientos, análisis, diseño, implementación, pruebas y entrega. [Ver Anexo 9](#)

RUP define por lo tanto el proceso mediante dos dimensiones:

- Dinámica o en el tiempo. Se expresa en términos de ciclos, fases, iteraciones y echas límite o hitos (milestones).
- Estática. Se describe en términos de actividades, entregables, roles y flujos de trabajo.

RUP organiza las actividades del proceso en 9 disciplinas totales que se dividen entre 2 grupos: 6 disciplinas básicas y 3 disciplinas de soporte:

- Disciplinas básicas: Modelo del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue (o Entrega).
- Disciplinas de soporte: Configuración y Gestión de Cambios, Gestión de Proyectos y Entorno. (11)

EUP

Existe otra variante estandarizada de RUP, llamada EUP (Enterprise Unified Process), la cual nace debido a la necesidad de tomar en cuenta aspectos cruciales para el desarrollo sobre todo de sistemas empresariales. Este define nuevas fases y 8 nuevas disciplinas. Sobre esta metodología no se considera necesario abundar ya que es una variante de RUP que no se ajusta a las condiciones del negocio por lo cual no se valora como posible metodología a utilizar.

[Ver Anexo 10.](#) (11)

RUP Agilizado

RUP no fue creado para ser una metodología pesada, aunque la gran cantidad de actividades y entregables opcionales que define ha llevado a la impresión o mala interpretación que sí lo es. En realidad fue definido para ser adaptado y aplicado en cada empresa de la forma que se crea necesario.

Realmente, podemos utilizar RUP como si de una metodología ligera se tratara si seguimos las siguientes prácticas:

- Utilizar un conjunto pequeño de actividades y entregables. Los entregables de RUP son opcionales, por lo cual evitaremos crear aquéllos que no nos proporcionen valor. Podemos enfocarnos en programar de forma previa y no en documentar de forma previa.
- RUP es una metodología iterativa, en la cual los requerimientos y el diseño no se completan antes de la implementación. Éstos se van definiendo a medida que avanzamos en las iteraciones, según el feedback obtenido del usuario o cliente.
- Aplicar modelos ágiles a UML. (11)

En relación con el sistema que modela este trabajo, se han aplicado varias de estas buenas prácticas para agilizar el proceso de desarrollo de software:

1. Se predefinieron las actividades principales para modelar el sistema, obviando actividades y artefactos que no fueran relevantes.
2. Aprovechando el modelo incremental de RUP, se trabajaban paralelamente en varios flujos de trabajo al mismo tiempo.
3. Se mantiene una constante interacción con el cliente, lo cual posibilita detectar con mayor rapidez cambios en los requerimientos.
4. Debido a la necesidad de presentar resultados se documenta paralelamente a la implementación.

2.2.1 Fundamentación de la Metodología Utilizada. Proceso Unificado de Desarrollo

RUP sin duda alguna da una amplia gama de posibilidades para el desarrollo de un sistema (proyecto) ya sea a largo o a corto plazo. Para la construcción del sistema usamos RUP por las siguientes razones

1. Metodología estándar “de facto” a nivel mundial.
2. Alta adaptabilidad a las condiciones reales del desarrollo del sistema. Es decir que podemos hacerla más ágil según se necesite.
3. Metodología para entornos de desarrollo orientado a objetos.
4. Alta capacidad de generar documentación relevante del negocio.
5. Alta retroalimentación (feedback) con el cliente. (Iterativo e Incremental).
6. Dominio básico de la metodología por el personal del proyecto.

Todas estas ventajas, están altamente unidas a las características del negocio, del sistema en construcción y del entorno en que se desarrolla el proyecto.

El Polo Productivo PetroSoft posee poco tiempo de experiencia en la construcción de sistemas para la industria del petróleo, por lo cual se necesita una metodología que genere documentación útil sobre el negocio. Además que al ser iterativo e incremental posibilita tener al cliente en constante retroalimentación, lo cual brinda un desarrollo más rápido y efectivo del sistema. Todo esto unido a las posibilidades de desarrollo de esta metodología en la actualidad, así como su alta adaptabilidad a las condiciones reales, se fundamenta el porqué de la utilización del *Proceso Unificado de Desarrollo* como metodología de desarrollo para el Sistema de Manejo Integral de Perforación de Pozos.

A continuación se presenta una tabla comparativa entre las diferentes metodologías de desarrollo de software valoradas para guiar el proceso de desarrollo en el sistema propuesto, lo cual fundamenta el porqué se decide RUP como metodología.

Escala: 0 a 5. Mientras mayor el número más elegible sería.

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

Como se aprecia en la tabla se han utilizado varios criterios de evaluación, fundamentados en las características del negocio, que enfrenta el sistema propuesto, por lo cual esta comparación arroja que la metodología correcta es RUP, aplicando varias buenas prácticas para agilizar su funcionamiento.

2.3 Lenguaje para Modelado de Objetos

El Lenguaje Unificado de Modelado UML – OMG(Object Management Group), por sus siglas en ingles) es el más utilizado en la especificación, y la manera en que el mundo no sólo los modelos de solicitud de la estructura, comportamiento, y la arquitectura, sino también de procesos de negocio y estructura de datos.

UML, proporciona una clave para la fundación de la OMG-Modelo Impulsado Arquitectura, que unifica todas las fases del desarrollo y la integración de las empresas de modelación, a través de la arquitectura y la aplicación de modelado, al desarrollo, despliegue, mantenimiento, y la evolución. La última versión liberada por OMG de UML es la 2.2 en febrero el año en curso (2009). (33)

UML no requiere un proceso de desarrollo en particular. Aunque UML puede usarse con varios procesos de desarrollo, fue diseñado para usarse con un proceso iterativo, incremental, guiado por casos de uso y centrado en la arquitectura, el tipo de proceso que consideramos más apropiado para el desarrollo de sistema complejos modernos. El Proceso Unificado de Desarrollo de Software (RUP) describe el tipo de proceso que complementa UML y que ayuda mejor al desarrollo de software. (34)

UML es el lenguaje de modelado de objetos escogido para el modelado des sistema que se propone en este trabajo, debido a sus características, probada eficacia y perfecta complementación con la metodología de desarrollo RUP.

2.4 Herramientas de CASE

La ingeniería de sistemas asistida por ordenador (Computer Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las

metodologías propias de desarrollo de sistemas su objetivo es acelerar el proceso para el que han sido diseñadas, específicamente para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

La tecnología CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información. Para mejorar la calidad y la productividad de los sistemas de información a la hora de construir software se plantean los siguientes objetivos:

- Permitir la aplicación práctica de metodologías estructuradas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

De una forma esquemática se puede decir que una herramienta CASE se compone de los siguientes elementos:

- Repositorio (diccionario) donde se almacenan los elementos definidos o creados por la herramienta, y cuya gestión se realiza mediante el apoyo de un Sistema de Gestión de Base de Datos (SGBD) o de un sistema de gestión de ficheros.
- Metamodelo (no siempre visible), que constituye el marco para la definición de las técnicas y metodologías soportadas por la herramienta.
- Carga o descarga de datos, son facilidades que permiten cargar el repertorio de la herramienta CASE con datos provenientes de otros sistemas, o bien generar a partir de la propia herramienta esquemas de base de datos, programas, etc. que pueden, a su vez, alimentar otros sistemas. Este elemento proporciona así un medio de comunicación con otras herramientas.
- Comprobación de errores, facilidades que permiten llevar a cabo un análisis de la exactitud, integridad y consistencia de los esquemas generados por la herramienta.
- Interfaz de usuario, que constará de editores de texto y herramientas de diseño gráfico que permitan, mediante la utilización de un sistema de ventanas, iconos y menús, con la ayuda del ratón, definir los diagramas, matrices, etc. que incluyen las distintas metodologías.

Las herramientas CASE evolucionan hacia tres tipos de integración:

1. La integración de datos permite disponer de herramientas CASE con diferentes estructuras de diccionarios locales para el intercambio de datos.
2. La integración de presentación confiere a todas las herramientas CASE el mismo aspecto.
3. La integración de herramientas permite disponer de herramientas CASE capaces de invocar a otra herramientas CASE de forma automática. (17)

A continuación se presentan varias de las herramientas CASE más utilizadas en la actualidad.

- **BoUml**

Es una herramienta CASE gratuita que utiliza [UML](#) 2.0 para permitirte especificar y generar código en [C++](#), [Java](#), [Python](#) y [PHP](#), multiplataforma ([Unix](#), [Windows](#) y [MacOS](#)), no requiere de mucha memoria para manejar una gran cantidad de clases, utilizar un modelo extensible y está liberada bajo la licencia [GNU](#). (35)

- **Visible Analyst 2008**

Perteneciente a la compañía Visible Enterprise, integra en una sola herramienta la planificación, el modelado de datos, modelado de procesos (BPMN), modelado UML, y análisis y Diseño estructurado de modelado. Especializado en diseño de aplicaciones de e-business, el desarrollo de data warehouse y en la integración de sistemas heredados. Facilita el procesamiento de documentación, y el almacenamiento de estos y de datos, procesos requisitos, y/o objetos que pueden ser o no compartidos para los desarrolladores. Esta liberado bajo licencia comercial. (36)

- **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 2.0](#), Java, [C#](#), C++, [VB.Net](#), [Delphi](#), [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. (37)

- **Rational Rose**

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte UML 1.x. *Rational Rose Enterprise* es una de las más eficientes herramientas para el ambiente de modelado que soporta la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Como, todos los demás productos, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Características

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 inversa 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

- **Visual Paradigm for UML 6.1 Enterprise Edition**

El Visual Paradigm for 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.

- 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, [Objeto Definition Language \(ODL\)](#), 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 XML 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. (38)

A continuación se presenta una tabla en la cual se compara a tres de las herramientas, sobre las cuales se investigó en este trabajo, ya que se considera que son las más potentes en la actualidad.

Aspectos	VP-EE	EA	Rational Rose
Modelado UML	Si.UML 2.1	Si. UML 2.1	Si. UML 1.0
Modelado de Base de Datos	Si	Si	Si
Metodología de Desarrollo	Independiente de la Metodología	Independiente de la Metodología	Orientado a RUP
Integración con el IDE	Alto	Alto	Bajo
Trabajo en Equipo	Si	Si	Si

Generación de Código	Alto	Alto	Bajo
Generación de Documentación	Si	Si	SI
Licencia	Comercial (Privada). Versión Community Gratis	Comercial (Privada). Versión de Prueba por 30 días	Comercial (Privada)
Soporte	Multiplataforma	Multiplataforma	Multiplataforma
Conocimiento de la Herramienta	Básico	Ninguno	Medio

Luego de haber analizado todas estas herramienta, se escoge como herramienta CASE: **Visual Paradigm for UML 6.1 Enterprise Edition**. Como se pudo apreciar en la tabla anterior, tanto EA, como Rational Rose, poseen características muy similares, pero sin dudas VP, es mucho más competente para el trabajo que se desea realizar lo cual posibilitará que se logren los resultados esperados de este trabajo.

Es importante abordar brevemente a las herramienta BOUML, una herramienta sumamente interesante, ya que logra lo que ninguna de las demás que es el poder instalarse en sistemas operativos MacOS, además de ser una herramienta, para la cual no se necesitan estaciones de trabajo tan potentes y estar liberado bajo licencia GNU, lo cual aporta una gran gama de oportunidades hacia el futuro.

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.
5. Genera código directo e inverso para Postgree SQL

Por estas razones esta herramienta se complementa perfectamente con el IDE y con el Gestor de Base de Datos, además, la metodología de desarrollo utilizada RUP (ver [epígrafe](#)

[anterior](#)), donde se utilizan buenas prácticas para agilizar el desarrollo, el VP contribuye a que se pueda programar y documentar al unísono, por todas estas razones se ratifica la utilización de **Visual Paradigm for UML 6.1 Enterprise Edition**.

2.5 Conclusiones Parciales

En este capítulo se analizaron las tendencias y tecnologías actuales utilizadas, dando los conceptos necesarios para comprender el por qué de las tecnologías que serán utilizadas en el desarrollo de la aplicación. Se realiza un amplio estudio de las principales metodologías de desarrollo que se utilizan en la actualidad, fundamentando la metodología a utilizar. Se hace referencia al Lenguaje Unificado de Modelado, también se realiza un estudio de las Herramientas CASE, más usadas actualmente, abordando su gran potencialidad y facilidades que brindan a la hora de desarrollar las aplicaciones, fundamentando así el porqué se utiliza la Visual Paradim for UML Enterprise Editon.

Capítulo III: Características del Sistema

3.1 Introducción

En este capítulo se realiza el análisis del negocio, luego el proceso de la captura de requerimientos del sistema, tanto los funcionales como los no funcionales, la construcción de los casos de uso del sistema, los diagramas que representan la solución propuesta del sistema y para finalizar la descripción textual de cada uno de ellos.

3.2 Modelo de Negocio

3.1.1 Reglas del Negocio

- A partir de las 12:00am y hasta las 6:00am los supervisores del pozo elaboran el Parte Diario y el Cierre Operativo Diario del Pozo.
- A las 6:00am llega a la DIPP el Parte Diario de Perforación y el Cierre Operativo Diario del Pozo que envían los supervisores de pozos, con estas partes se realiza un resumen general, el cual se denomina el Reporte Diario de Perforación de Pozos que es enviado por la oficina de despacho DIPP a CUPET, el mismo se elabora teniendo en cuenta varias partes.

3.1.2 Actores y Trabajadores del Negocio

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

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

CEINPET	Se encarga de recepcionar la información geológica enviada de los pozos en perforación y supervisar el la construcción del Parte Diario de Geología.
----------------	--

Trabajador: Representa a personas, o sistemas (software) dentro del negocio que son las que realizan las actividades que están comprendidas dentro de un caso de uso.

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 este elabore el reporte diario de perforación del pozo.
Secretaria de Despacho DIPP.	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.
Direccional	Encargado de elaborar el Reporte del Direccional, a partir de los resultados obtenidos del sistema del Mago del Pozo. Este es entregado al supervisor del pozo para que posteriormente este elabore el

	reporte diario de perforación del pozo.
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.

Los actores y trabajadores del negocio, son artefactos generados en el trabajo, del flujo de trabajo, de Negocio. Su correcta identificación es de suma importancia para lograr un modelo de sistema(artefacto del flujo de requerimientos) efectivo, ya que los trabajadores del negocio, son posibles actores del sistema y de no identificarse correctamente podría existir confusión y podría colocarse a los actores del negocio como actores del sistema o viceversa, por solo citar un ejemplo.

3.1.3 Casos de Uso del Negocio

Caso de Uso del Negocio: Es un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.

Casos de Uso del Negocio :

1. Modulo_Negocio_CEINPET

- Elaborar Parte Diario de Geología

2. Modulo_Negocio_DIPP

- Elaborar Parte Diario Operativo de Perforación.
- Elaborar Plan de Transportación
- Elaborar Plan Operativo Mensual

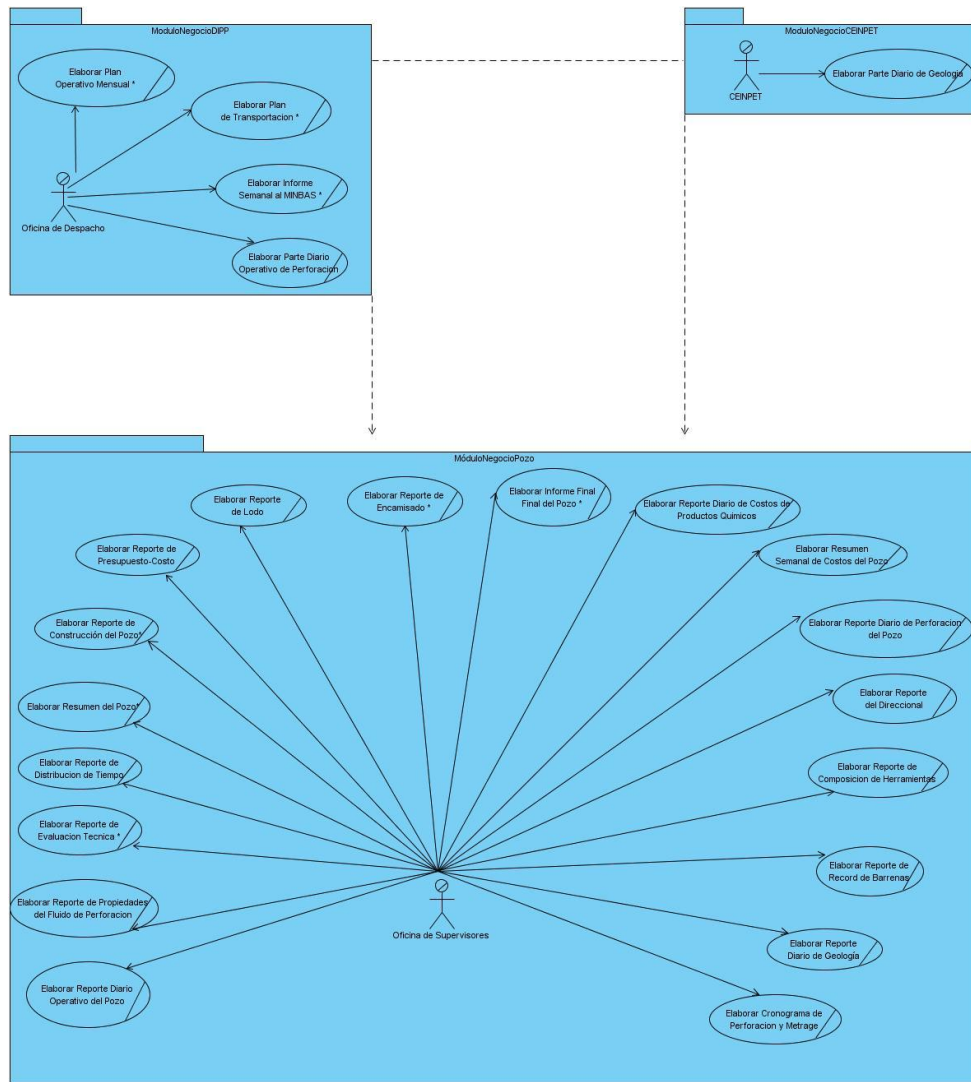
- Elaborar Informe Semanal al MINBAS

3. Módulo_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 Direccional
- 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

Los Casos de Uso del Negocio (CUN) constituye un artefacto generado por el flujo de trabajo de Negocio. La identificación correcta de estos posibilita una correcta determinación de los requisitos funcionales del sistema, ya que no debe existir ninguna funcionalidad del sistema que no tenga un proceso de negocio asociado. De no realizarse esta actividad con la calidad y seriedad que se requiere, podría mal interpretarse por completo las necesidades del cliente, así como el sistema a modelar lo que traería la pérdida de tiempo volviendo a realizar el levantamiento de negocio e identificación de los requerimientos o en el peor de los casos la pérdida del cliente.

3.1.4 Diagrama de Casos de Uso del Negocio

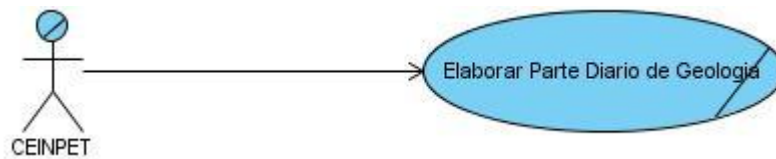


El Diagrama de CUN es el artefacto donde se integran actores y CUN, mostrando así la relación que existe entre ellos. Es importante destacar, que si un diagrama de CUN está correcto el cliente debe ser capaz de entender lo que a él se le muestra, en caso de que se decida mostrárselo al cliente, una actividad que puede ser muy efectiva para medir la calidad del Modelo de Negocio que se elabora, ya que el DCUN es el principal artefacto a generar en el flujo de trabajo de negocio.

En el caso específico del DCUN que se genera para la efectiva modelación del sistema propuesto, se decide agrupar, utilizando el estereotipo de paquete que propone UML, los CUN con sus respectivos actores en Módulo de Negocio. Aunque

RUP no indica que sea necesario, agrupar los artefactos generado es un diagrama de paquetes, se considera efectivo realizarlo para la modelación del sistema en cuestión, ya que el conocimiento que posee el equipo de trabajo acerca del negocio es de nivel básico, por tanto esta agrupación ayudará a la trazabilidad del sistema por cada uno de los flujos de trabajo. Se agrupó los procesos de negocio en Módulos según las entidades que interactúan con el negocio en cuestión.

Modulo de Negocio CEINPET

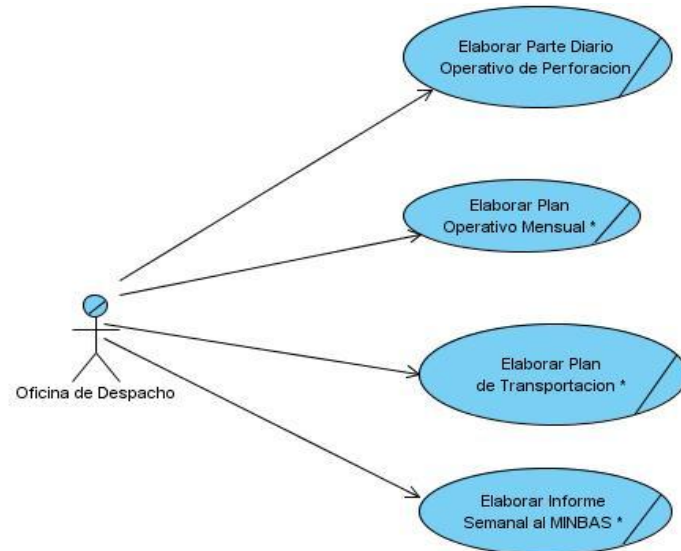


Este módulo agrupa los procesos de negocio pertenecientes a al CIENPET.

Caso de uso del negocio	Elaborar Parte Diario de Geología
Actor	CEINPET
Resumen	El caso de uso se inicia cuando el CEINPET ordena la confección del Parte Diario de Geología a las 8:00 AM, a partir de la información consultada de los Reportes Diario de Perforación y Reporte Diario de Geología. Este Parte recoge toda la información Geológica de los pozos que en ese momento están siendo perforados
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. Ordena realizar el Parte Diario de Geología	a. Consulta el Reporte Diario de Perforación b. Consulta el Reporte Diario de Geología c. Confecciona el Parte Diario de Geología d. Almacena el parte e. Envía por correo a las instancias superiores (DIPP, CUPET y MINBAS)

Mejoras propuestas:	

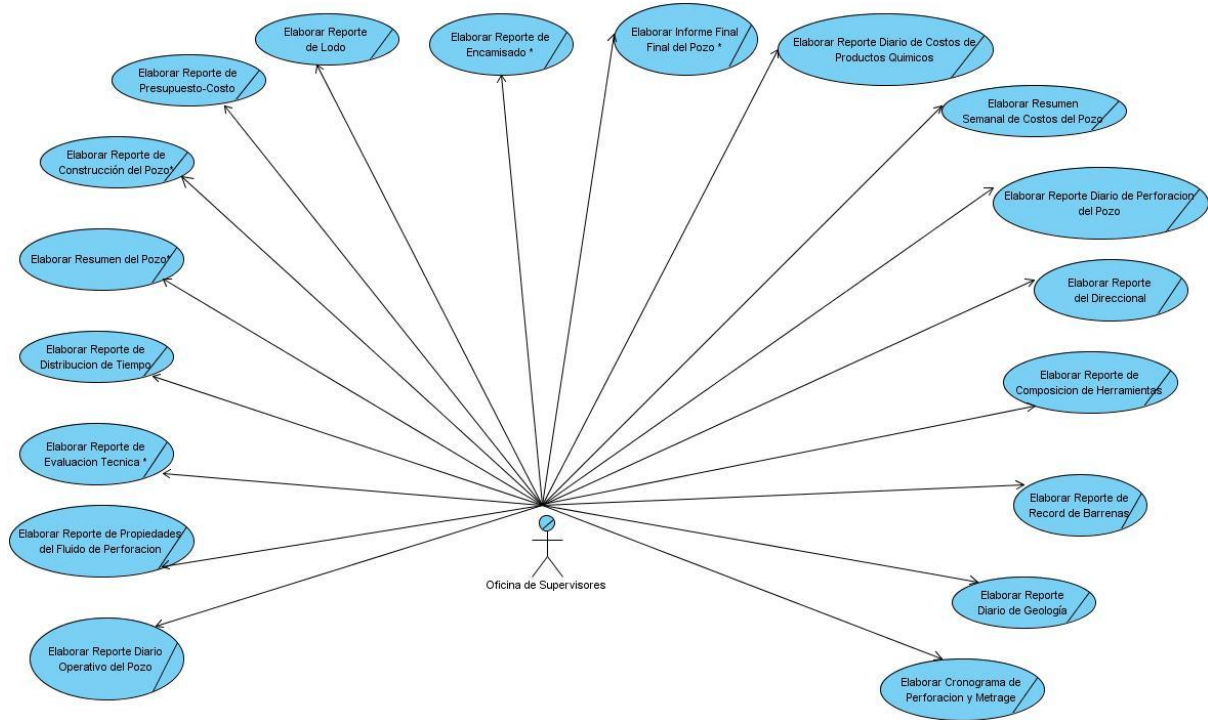
Modulo_Negocio_DIPP



Caso de uso del negocio	Parte Diario Operativo de Perforación
Actor	Oficina de la DIPP.
Resumen	El caso de uso se inicia cuando en la Oficina de la DIPP realiza a las 8.00 a.m el Parte Diario operativo de Perforación, por lo cual la Secretaria de Despacho se da la tarea de consultar los datos de los partes operativos de perforación de cada uno de los pozos y elabora el Parte Diario Operativo de Perforación, lo almacena y lo 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 la DIPP solicita el Parte Diario de Perforación.</p>	<p>1.1 Consulta el Reporte Diario Operativo del Pozo.</p> <p>1.2 Elabora el Parte Diario Operativo de Perforación.</p> <p>1.3 Almacena el Parte Diario Operativo de Perforación.</p> <p>1.4 Envía por correo el Parte Diario Operativo de Perforación a la Dirección de la DIPP</p>
<p>Mejoras propuestas:</p>	<p>Que sea generado automáticamente a partir de la información consultada.</p>

Módulo_Negocio_Pozo



<p>Caso de uso del negocio</p>	<p>Elaborar Cronograma de Perforación y Metraje.</p>
<p>Actor</p>	<p>Oficina de Supervisores</p>

Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores consulta el reporte Cronograma de Perforación y Metraje, y ordena al Supervisor del Pozo a actualizar el Cronograma de perforación y metraje este va introduciendo diariamente la información en este reporte y posteriormente lo almacena.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena actualizar el Cronograma de Perforación y Metraje.	<p>1.1 El supervisor del pozo consulta la información contenida en el Cronograma de Perforación y Metraje para actualizar los datos.</p> <p>1.2 El supervisor del pozo introduce los datos correspondientes al día en curso de la perforación:</p> <ul style="list-style-type: none"> • Metraje diario en proyecto. M(m\d) • Metraje diario real. <p>1.3 El supervisor del pozo actualiza la información y la almacena.</p> <p>1.4 Automáticamente es generado en Excel un gráfico con los datos almacenados hasta la fecha.</p>
Mejoras Propuestas	

Caso de uso del negocio	Elaborar Reporte Record de Barrenas.
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores ordena sacar la barrena. El

	Supervisor de Pozo consulta el Catalogo de la Barrena, además de que extrae información visual, cuando se saca la barrena, una vez que se vaya a comenzar en un nuevo intervalo de la perforación del pozo de petróleo, luego es almacenada para su posterior uso.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena un cambio de barrena	<p>1.1 El supervisor del pozo consulta el reporte para introducir los datos según el intervalo de perforación que sea :</p> <ul style="list-style-type: none"> • Tipo de barrena • Diámetro • Intervalo <p>1.2 El supervisor del pozo almacena este reporte para su posterior consulta.</p>
Mejoras propuestas:	Brindar la posibilidad en el sistema de que el actor consulte según varios criterios (deben ser suministrados por los especialistas); los distintos tipos de barrenas y las características que ofrecen en determinado tipo de suelo, así como su tiempo de duración y efectividad en el proceso de perforación del pozo de petróleo.

Caso de uso del negocio	Elaborar Reporte de Composición de Herramientas.
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores realiza un cambio de

	herramienta, por cada uno de los intervalos de la perforación del pozo se va registrando las herramientas utilizadas, luego es almacenado para su posterior uso; aunque en un intervalo se puede utilizar más de una composición de herramientas.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores realiza un cambio en la herramienta utilizada en el pozo.	<p>1.1 El supervisor del pozo consulta el reporte para introducir los datos correspondientes a cada herramienta por cada intervalo de la perforación:</p> <ul style="list-style-type: none"> • Descripción • Diámetro externo • Diámetro interno. • Longitud • Intervalo <p>1.2 El supervisor del pozo almacena este reporte para su posterior consulta.</p>
Mejoras propuestas:	

Caso de uso del negocio	Elaborar Reporte de Fluidos de Perforación.
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores recibe el Reporte de Lodo y ordena elaborar el Reporte de Fluidos de Perforación, al supervisor de pozo.

Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena elaborar el Reporte de Fluidos de Perforación.	<p>1.1 El supervisor del pozo consulta el Reporte de Lodo introduce los datos de los fluidos correspondientes a cada intervalo de la perforación del pozo según la profundidad:</p> <ul style="list-style-type: none"> • Intervalo • Profundidad • Fecha • Api • PH <p>1.2 El supervisor del pozo almacena este reporte para su posterior consulta.</p>
Mejoras propuestas:	Se plantea que en el sistema propuesto sean generados varios datos automáticamente a partir de otros reportes anteriormente generados.

Caso de uso del negocio	Reporte Diario de Perforación del Pozo
Actor	Oficina de Supervisores.
Resumen	El caso de uso se inicia cuando a las 12.00 a.m el supervisor del pozo realiza el reporte diario de perforación, éste consulta la información entregada por las compañías de servicio al pozo para confeccionar su reporte, que posteriormente es enviado a la DIPP de perforación y a las distintas áreas del pozo.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores	1.1 El supervisor del pozo consulta los

<p>ordena realizar el Reporte Diario de Perforación</p>	<p>reportes emitidos por las compañías de servicio al pozo.</p> <ul style="list-style-type: none"> • Reporte de Lodo • Reporte Direccional • Reporte Record de Barrenas • Reporte Composición de Herramientas. <p>1.2 El supervisor del pozo realiza el reporte Diario de Perforación del pozo.</p> <p>1.3 El supervisor del pozo almacena este reporte según el día de perforación.</p> <p>1.4 El supervisor del pozo envía este reporte a la DIPP de perforación.</p>
<p>Mejoras propuestas:</p>	<p>Que a la hora de realizar este reporte, se obtengan automáticamente los datos que son tomados de otros reportes anteriormente elaborados, lo que propiciaría que gran parte del mismo se genere automáticamente.</p>

<p>Caso de uso del negocio</p>	<p>Elaborar Reporte Diario Operativo del Pozo.</p>
<p>Actor</p>	<p>Oficina de Supervisores.</p>
<p>Resumen</p>	<p>El caso de uso se inicia cuando a las 12.00 a.m el supervisor del pozo realiza el Reporte Diario Operativo, éste consulta información del reporte diario de perforación para confeccionar su reporte, que posteriormente es enviado a la DIPP de perforación y a las distintas áreas del pozo.</p>
<p>Casos de uso asociados</p>	<p>Ninguno</p>

Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena realizar el Reporte Diario de Perforación.	<p>1.1 El supervisor del pozo consulta el Reporte Diario de Perforación del pozo.</p> <p>1.2 El supervisor del pozo realiza el reporte diario operativo a partir de los datos consultados.</p> <p>1.3 El supervisor del pozo almacena este reporte.</p> <p>1.4 El supervisor del pozo envía este reporte a la DIPP de perforación.</p>
Mejoras propuestas:	Que sea generado este reporte automáticamente a partir de los reportes consultados.

Caso de uso del negocio	Elaborar Resumen Semanal de Costos del Pozo.
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando a las 12.00 a.m el lunes, el supervisor del pozo realiza el Reporte Resumen Semanal de Costos del Pozo, éste consulta información de los reportes diarios entregados por las compañías de servicios y de los costos a pagar por otros servicios prestados al pozo.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena elaborar el Reporte Semanal de Costos del Pozo	<p>1.1 El supervisor del pozo consulta los reportes de las empresas de servicio para tomar los costos.</p> <p>1.2 El supervisor del pozo elabora el Reporte</p>

	Resumen Semanal de Costos del Pozo. 1.3 El supervisor del pozo almacena esta información.
Mejoras propuestas:	Que sea generado este reporte automáticamente a partir de los reportes consultados.

Caso de uso del negocio	Elaborar Reporte Diario de Costos de Productos Químicos.
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando a las 12.00 a.m el supervisor del pozo realiza el Reporte Diario de Costos de Productos Químicos, éste consulta información del reporte de Lodo entregado por el químico del pozo para confeccionar su reporte, que posteriormente es almacenado.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena elaborar el Reporte Diario de Costos de Productos Químicos.	1.1 El supervisor del pozo consulta el Reporte de Lodo para tomar los costos de los productos químicos 1.2 Elabora el Reporte Diario de Costos de Productos Químicos. 1.3 El supervisor del pozo almacena este reporte.
Mejoras propuestas:	Que sea generado parcial o totalmente de manera automática.

Caso de uso del negocio	Elaborar Reporte Direccional
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores solicita el Reporte Direccional, el cual es elaborado por el Direccional del Pozo y entregado al Supervisor del Pozo.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores solicita el Reporte Direccional.	1.1 El direccional confecciona el reporte 1.2 Entrega el Reporte al Supervisor del Pozo
2. La Oficina de Supervisores almacena la información.	
Mejoras propuestas:	

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 Reporte Direccional y 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

<p>1. La Oficina de Supervisores solicita el Reporte Diario Geología.</p>	<p>1.1 El geólogo del pozo consulta los datos entregados por los químicos y los direccionales del pozo.</p> <p>1.2 El Geólogo del Pozo utiliza estos datos y elabora el Reporte Diario de Geología.</p> <p>1.3 El Geólogo del Pozo Almacena Reporte.</p> <p>1.4 El Geólogo del Pozo envía por correo el Reporte Diario de Geología al CEINPET.</p>
<p>2. La Oficina de Supervisores recepciona este reporte</p>	
<p>Mejoras propuestas:</p>	

Caso de uso del negocio	Elaborar Reporte de Lodo
Actor	Oficina de Supervisores
Resumen	El caso de uso se inicia cuando la Oficina de Supervisores solicita el Reporte Lodo. El Químico del Pozo, procesa los datos y elabora el Reporte de Lodo, almacena el reporte. Lo imprime y lo entrega a la Oficina de Supervisores.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
<p>1. La Oficina de Supervisores solicita el Reporte Lodo</p>	<p>1.1 El Químico del Pozo realiza las pruebas al lodo y elabora el reporte.</p> <p>1.2. El Químico del Pozo almacena Reporte.</p> <p>1.3 El Químico del Pozo imprime el Reporte</p> <p>1.4 El Químico del Pozo entrega el Reporte</p>

2. La Oficina de Supervisores recepciona este Reporte	
Mejoras propuestas:	

Caso de uso del negocio	Elaborar Reporte Distribución de Tiempo
Actor	Oficina de Supervisores.
Resumen	El caso de uso se inicia cuando el actor Oficina de Supervisores ordena a las 12.00 a.m elaborar el Reporte de Distribución de Tiempo.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. La Oficina de Supervisores ordena elaborar el Reporte de Distribución de Tiempo	1.1 El supervisor del pozo confecciona el Reporte Distribución de Tiempo a partir de las actividades realizadas durante el día en el pozo. 1.2 El supervisor del pozo almacena este reporte.
Mejoras propuestas:	

Caso de uso del negocio	Elaborar Reporte de Presupuesto-Costo
Actor	Oficina de Supervisores.
Resumen	El caso de uso se inicia cuando se comienzan las actividades en el pozo, se consulta el presupuesto de la inversión del pozo. Se confecciona y se almacena el reporte.
Casos de uso asociados	Ninguno
Acción del actor	Respuesta del proceso de negocio
1. Inicio de las Actividades en	1.1 El supervisor del pozo consulta el

el Pozo	<p>presupuesto asignado para la inversión.</p> <p>1.2 El supervisor del pozo confecciona Reporte de Presupuesto-Costo a partir de las actividades realizadas durante el día en el pozo.</p> <p>1.3 El supervisor del pozo almacena este reporte.</p>
Mejoras propuestas:	

3.1.5 Modelo de Objetos. [Ver Anexo 13](#)

El modelo de objetos es un artefacto que agrupa, a los trabajadores del negocio con las entidades del negocio. Aunque no la misma importancia que los artefactos abordados anteriormente, pero a pesar de ello, posee gran relevancia para el modelado de las Base de Datos con que va a interactuar el sistema, además al igual que el DCUN, la calidad con que se realice este artefacto, el cliente puede valorar si es correcto o no la visión que se obtuvo del levantamiento de negocio que se realizó, lo cual avalaría la calidad del mismo.

3.3 Requisitos de Software

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.

3.3.1 Requisitos Funcionales

Son capacidades o condiciones que el sistema debe cumplir.

RF 1 Gestionar Reporte Diario de Perforación

RF 1.1 Insertar Datos del Reporte Diario de Perforación

RF 1.2 Actualizar Datos del Reporte Diario de Perforación

RF 2 Generar Reporte Diario Operativo

RF 3 Gestionar Cronograma Diario de Perforación y Metraje

RF 3.1 Insertar Datos del Datos del Cronograma Diario de Perforación y Metraje

RF 3.2 Actualizar Datos del Datos del Cronograma Diario de Perforación y Metraje

- RF 4 Generar Reporte de Record de Barrenas
- RF 5 Gestionar Reporte Composición de Herramientas
 - RF 5.1 Insertar Datos de Composición de Herramientas
 - RF 5.2 Actualizar Datos del Composición de Herramientas
- RF 6 Gestionar Reporte de Propiedades de Fluido de Perforación
 - RF 5.1 Insertar Datos de Propiedades de Fluido de Perforación
 - RF 5.2 Actualizar Datos de Propiedades de Fluido de Perforación
- RF 7 Generar Reporte de Costo de Productos Químicos
- RF 8 Generar Reporte de Distribución del Tiempo Diario
- RF 9 Generar Reporte de Distribución del Tiempo por Etapas
- RF 10 Gestionar Reporte de Presupuesto Diario
 - RF 10.1 Insertar Datos del Reporte de Presupuesto Diario
 - RF 10.2 Actualizar Datos del Reporte de Presupuesto Diario
- RF 11 Generar Resumen de Costo
- RF 12 Gestionar Reporte de Encamisado de Superficie
 - RF 12.1 Insertar Datos del Reporte de Encamisado de Superficie
 - RF 12.2 Actualizar Datos del Reporte de Encamisado de Superficie
- RF 13 Gestionar Reporte de Encamisado Técnico.
 - RF 13.1 Insertar Datos del Reporte de Encamisado Técnico
 - RF 13.2 Actualizar Datos del Reporte de Encamisado Técnico
- RF 14 Gestionar Reporte de Encamisado Liner
 - RF 14.1 Insertar Datos del Reporte de Encamisado Técnico
 - RF 14.2 Modificar Datos del Reporte de Encamisado Técnico
- RF 15 Generar Reporte de Evaluación Técnica.
 - RF 15.1 Insertar Datos del Reporte de Encamisado Técnico
 - RF 15.2 Actualizar Datos del Reporte de Encamisado Técnico

RF 16 Gestionar Resumen de Pozo.

RF 16.1 Insertar Datos del Resumen del Pozo

RF 16.2 Actualizar Datos del Resumen del Pozo

RF 17 Gestionar Reporte de Construcción de Pozo

RF 17.1 Insertar Datos del Reporte de Construcción de Pozo

RF 17.2 Actualizar Datos del Reporte de Construcción de Pozo

RF 18 Gestionar Reporte Diario de Geología

RF 18.1 Insertar Datos del Reporte Diario de Geología

RF 18.2 Actualizar Datos del Reporte Diario de Geología

RF 19 Gestionar Reporte de Lodo

RF 19.1 Insertar Datos del Reporte de Lodo

RF 19.2 Actualizar Datos del Reporte de Lodo

RF 20 Gestionar Reporte Direccional

RF 20.1 Insertar Datos del Reporte Direccional

RF 20.2 Actualizar Datos del Reporte Direccional

RF 21 Generar Informe Semanal al MINBAS

RF 22 Gestionar Plan Operativo Mensual.

RF 22.1 Insertar Datos del Plan Operativo Mensual

RF 22.2 Actualizar Datos del Plan Operativo Mensual

RF 23 Generar Parte Diario de Perforación

RF 24 Gestionar Parte Diario de Intervención

RF 24.1 Insertar Datos del Parte Diario de Intervención

RF 24.2 Actualizar Datos del Parte Diario de Intervención

RF 25 Gestionar Parte Diario Operativo de Perforación

RF 25.1 Insertar Datos del Parte Diario Operativo de Perforación

RF 25.2 Modificar Datos del Parte Diario Operativo de Perforación

RF 26 Gestionar Información Inicial de Pozo

RF 26.1 Insertar Información Inicial del Pozo

RF 26.2 Modificar Información Inicial del Pozo

RF 27 Gestionar Parte Diario de Geología

RF 27.1 Insertar Datos del Parte Diario de Geología

RF 27.2 Actualizar Datos del Parte Diario de Geología

RF 28 Construir la Columna Litológica de Pozo

RF 29 Generar Informe Final de Pozo

RF 30 Gestionar Usuario

RF 30.1 Insertar Usuario

RF 30.2 Actualizar Usuario

RF 30.3 Eliminar Usuario

RF 31 Gestionar Rol

RF 31.1 Crear Rol

RF 31.2 Actualizar Rol

RF 31.3 Eliminar Rol

RF 32 Autenticar Usuario

RF 33 Gestionar Permiso

RF 33.1 Crear Permiso

RF 33.2 Actualizar Permiso

RF 33.3 Eliminar Permiso

RF 34 Guardar Registros de Seguridad

RF 35 Consultar Registro de Seguridad

RF 36 Generar Gráfica de Reporte

RF 37 Visualizar Información de Reporte

RF 38 Consultar Información

Se define el sistema propuesto principalmente para la gestión de información, donde se debe presentar la información en forma de gráficos en dos dimensiones (2D). Además de ellos se deben presentar consultas de la información almacenada por lo cual se identifican requerimientos como Visualizar Información de Reportes y Consultar Información. También se identifican requerimientos para garantizar la seguridad mínima del sistema. Para lograr que el sistema sea interoperable con otros tipos de sistemas se identifican los requerimientos de importar y exportar archivos. Luego de identificar los requisitos funcionales apoyados en el Modelo de Negocio, las necesidades del cliente y las características que debe tener un sistema de este tipo, para poder competir con los que existen actualmente en el mercado, se identifican los requerimientos no funcionales.

Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado.

RNF 1 Usabilidad

- Preparar a los Administradores en la gestión de Roles y Permisos
- Preparar a los usuarios del sistema en la gestión con los Reportes y Partes
- Visibilidad del estado del sistema
- Control y libertad del usuario
- Consistencia y estándares
- Prevención de errores

RNF 2 Seguridad

- Seguridad del sistema (C.I.D).
- La seguridad se tratará desde la fase de diseño del sistema. Se garantizará un fuerte tratamiento de excepciones.
- Trabajo por Sesiones.

RNF 3 Rendimiento

- El sistema necesita un servidor de base de datos en una PC o servidor con (1 o superior) gigas de RAM y 2 discos duros de 160Gigas
- Tiempo de respuesta de transacción (menos de 3 segundos).
- El sistema debe permitir trabajar conectados concurrentemente 250 usuarios como mínimo
- El sistema necesita las PC clientes con 256-512 de RAM y un Navegador que soporte Java script

RNF 4 Soporte

- El soporte y/o mantenimiento del sitio no debe detener el servicio

RNF 5 Restricciones del Diseño

- El framework tiene en cuenta las posibles restricciones de diseño para la aplicación
- El diseño para la aplicación será de acuerdo a los distintos paradigmas y adecuado con la arquitectura seleccionada.

RNF 6 Reusabilidad

- La aplicación se construirá utilizando estándares internaciones y patrones, para facilitar su integración futura, con componentes desarrollados por cualquier empresa y garantizar posibilidades de un mantenimiento ágil.
- La documentación de la arquitectura debe ser reutilizable para poder documentarla como una familia de productos.
- Cada módulo del sistema se desarrollará de forma tal en la que pueda ser reutilizada.

RNF 7 Portabilidad

- El sistema deberá poder ser usado desde cualquier Sistema Operativo

RNF 8 Escalabilidad

- La propia arquitectura en que se encuentra estructurada la aplicación permite que si se necesita migrar por ejemplo de una base de datos a otra solo se vea afectada la capa de la modelo sin esto inferir cambios extremos en las vistas de la aplicación.

RNF 9 Interfaz

- Interfaz de interacción con el usuario.

RNF 10 Hardware

- Servidor Web
- Servidor de Bases de Datos
- PC Cliente por cada actor del sistema
- Conexión de Red

RNF 11 Software

- Sistema Operativo tanto Windows (win9.x o versión superior) como Linux (cualquiera de sus distribuciones).
- El Navegador Web compatible con HTML 2.0 y CSS, podrá ser Netscape 3 (o superior), Mozilla 2.0, Internet Explorer 4.2 (o superior) y compatibles.
- Servidor Web Apache 2.2.X o superior
- Servidor de Bases de Datos PostgreSQL8.2.
- Los requerimientos no funcionales complementan a los funcionales, ellos fueron presentados y aprobados por el cliente, por lo cual se comienza el modelado del sistema.

3.4 Modelo de Sistema

Siguiendo las actividades que propone RUP para el modelado del sistema, referente al flujo de trabajo de requerimientos, se dio la tarea de agrupar lo requerimientos funcionales en CUS, identificar los actores del sistema y describir los CUS desde una vista externa del sistema. Además se diseñan los prototipos de interfaz.

3.4.1 Actores del Sistema

En el Modelo de casos de uso se describe lo que hace el sistema para cada tipo de usuario. Cada uno de estos se representa mediante uno o más actores. También se representa mediante uno o más actores cada sistema externo con el que interactúa el sistema, incluyendo dispositivos externos como temporizadores, que se consideran externos al sistema. Por tanto, los actores representan terceros fuera del sistema que colaboran con el 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 una actor de alta

	importancia, ya que todos los casos de uso que inicia son críticos del sistema.
Químico de Pozo.	Inicia el caso de uso Gestionar Reporte de Lodo.
Secretaria de Despacho DIPP.	Inicia los Casos de Uso del Subsistema DIPP donde participa La Secretaria de la DIPP (Trabajador del Negocio).
Direccional	Inicia el Caso de Uso Gestionar Reporte del Direccional.
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 expuso en el epígrafe referente a los trabajadores del negocio, los actores del sistema se derivan de este artefacto del negocio. A ello se le suma el administrador del sistema, un actor que aparece por la necesidad de un actor que se encargara de interactuar con los casos de uso que agrupan lo requerimientos de seguridad.

3.4.2 Diagrama de Casos de Uso del Sistema

3.4.2.1 Subsistema CEINPET

Este subsistema de agrupar el CUS, encargado de responder al requerimiento derivado del Módulo de Negocio CEINPET, garantizando la trazabilidad del sistema, además de agrupar los CU derivados del requerimiento funcional Construir Columna

Litológica de Pozo. [Ver Anexo 14](#)

3.4.2.2 Subsistema Pozo

Este subsistema de agrupar los CUS, encargados de responder a los requerimientos derivados del Módulo de Negocio Pozo, garantizando la trazabilidad del sistema. [Ver Anexo 15](#)

3.4.2.3 Subsistema DIPP

Este subsistema de agrupar los CUS, encargados de responder a los requerimientos derivados del Módulo de Negocio DIPP, garantizando la trazabilidad del sistema. [Ver Anexo 16](#)

3.4.2.4 Subsistema Seguridad

Este subsistema agrupa los CUS, encargados de garantizar la seguridad del sistema. [Ver Anexo 17](#)

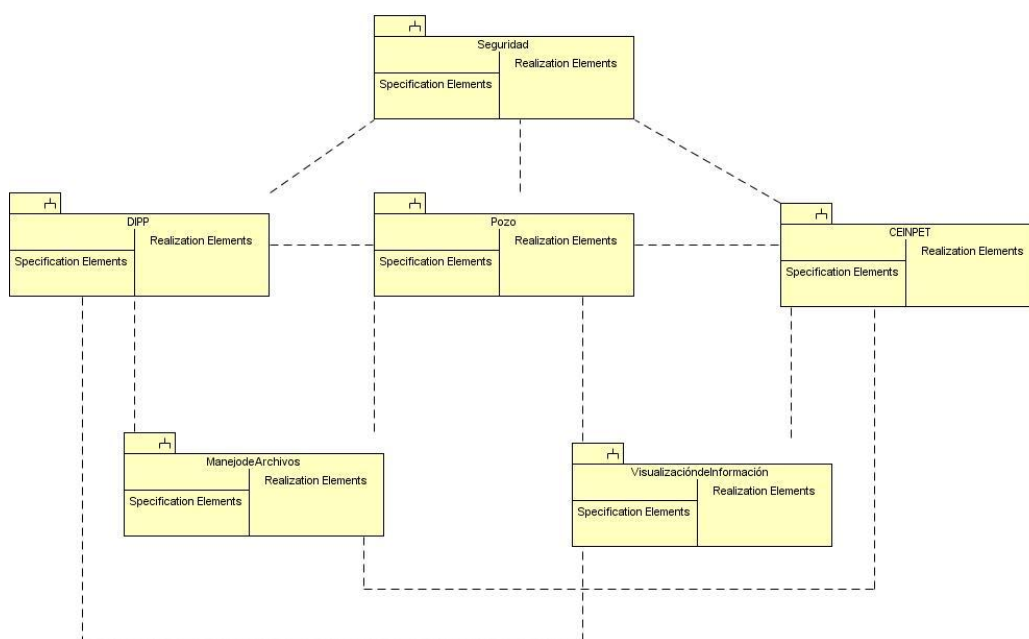
3.4.2.5 Subsistema Visualización de Información

Este subsistema agrupa los CUS, encargados de garantizar el acceso a la información almacenada en el sistema. [Ver Anexo 18](#)

3.4.2.6 Subsistema Manejo de Archivos

Este subsistema agrupa el CUS, encargado de garantizar la interoperabilidad del sistema. [Ver Anexo 19](#)

3.4.2.7 Diagrama De Casos de Uso por Subsistemas



Según la metodología RUP, no indica que necesariamente se agrupen los CUS del sistema, usando el estereotipo de subsistema de UML, pero se considera efectivo, la agrupación de los CUS en subsistemas, contribuye, a un mejor entendimiento del funcionamiento del sistema propuesto. Además esta agrupación contribuye a obtener una mejor trazabilidad de los casos de uso, a través de los flujos de trabajo.

Como se puede observar, existe una analogía, entre los Subsistemas Pozo, DIPP y CEINPET, con los Módulos de Negocio. El resto de los subsistemas son derivados de las necesidades manifestadas por el cliente. Todas las relaciones entre estos subsistemas son de asociación, ya que no existe dependencia entre ellos. Importante resaltar que la agrupación se realizó por funcionalidades.

3.4.3 Descripción Textual de los Casos de Uso del Sistema

3.4.3.1 Subsistema Pozo

Caso de Uso:	Gestionar Reporte Diario de Perforación	
Actor:	Supervisor de Pozo	
Resumen:	Este caso de uso se encarga de la gestión de la información generada diariamente en los pozos en perforación	
Precondiciones:	Se debe tener en el sistema la El Reporte de Lodo, Reporte de Direccional y el Reporte Diario de Geología; Así como la Información de las actividades realizadas por la compañía perforadora de las últimas 24 horas. Además de estar autenticados previamente en el sistema.	
Referencias	RF 2.1.1, RF 2.1.2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Sección 1: Insertar Datos del Reporte Diario de Perforación		
Acción del Actor	Respuesta del Sistema	
1. El Supervisor de Pozo selecciona la opción de Insertar Datos del Reporte Diario de Perforación	1.1 El Sistema muestra un formulario con los datos que debe contener el reporte. 1.2 El Sistema llena los campos que contienen informaciones de otros reportes. Los campos que contienen informaciones que son	

<ol style="list-style-type: none"> 2. El Supervisor de Pozo inserta los Datos 3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar 4. El Supervisor de Pozo confirma la opción de Guarda y Enviar 	<p>extraídas de otros reportes no pueden ser modificados.</p> <ol style="list-style-type: none"> 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema activa la opción de Guardar y Enviar 3.1 El Sistema muestra una venta emergente pidiendo confirmación de la operación 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 2: Actualizar Datos del Reporte Diario de Perforación

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El Supervisor de Pozo selecciona la opción de Actualizar Datos del Reporte Diario de Perforación 2. El Supervisor de Pozo actualiza los Datos. 3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar 4. El Supervisor de Pozo confirma la operación de Guardar y Enviar. 	<ol style="list-style-type: none"> 1.1 El Sistema muestra un formulario con todos los datos que se encuentran almacenados en el reporte. Los campos con datos extraídos de otros reportes no pueden ser modificados. 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una venta emergente pidiendo confirmación de la operación 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.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2a. Inserta los Datos o Actualiza los Datos	2a.1 "Datos introducidos incorrectos" El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo 2a.2 El Sistema limpia todos los caracteres extraños.
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" 4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

Caso de Uso:	Generar Reporte Diario Operativo
Actor:	Supervisor de Pozo
Resumen:	Este caso de uso se encarga de generar automáticamente el Reporte Operativo del Pozo, sin necesidad de que el usuario inserte ningún dato.
Precondiciones:	Reporte Diario de Perforación El actor debe estar autenticado previamente en el sistema.
Referencias	RF 2.2

Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo		1.1 El Sistema recopila los datos 1.2 El Sistema genera el Reporte Diario Operativo. 1.3 El Sistema visualiza el reporte. El usuario no puede tener la posibilidad de cambiar ningún dato de este reporte.
<i>Prototipo de Interfaz</i>		
Flujos Alternos		
Acción del Actor		Respuesta del Negocio
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo		1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
<i>Prototipo de Interfaz</i>		
<i>Prototipo de Interfaz</i>		
Poscondiciones	Redirecciona a la Interfaz Principal	

Caso de Uso:	Gestionar Cronograma de Perforación Planificado
Actores:	Supervisor de Pozo
Resumen:	El caso de uso se inicia cuando se necesita insertar o actualizar datos del Cronograma de Perforación.
Precondiciones:	El usuarios debe estar autenticado en el sistema.
Referencias	RF 2.3.1, RF 2.3.2
Prioridad	Crítico
Flujo Normal de Eventos	

Sección “Insertar Datos del Cronograma de Perforación Planificado”

Acción del Actor	Respuesta del Sistema
<p>1. Supervisor de Pozo selecciona la opción de Insertar Datos del Cronograma de Perforación Planificado</p> <p>2. El Supervisor Introduce los datos</p>	<p>1.1 El Sistema muestra un formulario con los datos que debe contener 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. El Supervisor de Pozo selecciona la opción de Guardar y Enviar</p>	<p>3.1 El Sistema muestra una venta emergente pidiendo confirmación de la operación</p>
<p>4. El Supervisor de Pozo confirma la operación de Guardar y Enviar</p>	<p>4.1 El Sistema comprueba que los datos no contengan errores</p> <p>4.2 El Sistema almacena la información.</p> <p>4.3 El Sistema envía un mensaje del éxito de la operación.</p>

Sección “Actualizar Datos del Cronograma de Perforación Planificado”

Acción del Actor	Respuesta del Sistema
<p>1. El Supervisor de Pozo selecciona la opción de Actualizar Datos del Cronograma de Perforación Planificado</p> <p>2. El Supervisor de Pozo actualiza la información</p>	<p>1.1 El Sistema muestra un formulario con todos 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 venta emergente pidiendo confirmación de la operación</p>
<p>El Supervisor de Pozo selecciona la opción de Guardar y Enviar</p>	<p>El Sistema comprueba que los datos no contengan errores</p>
<p>El Supervisor de Pozo confirma la opción de Guardar y Enviar</p>	<p>El Sistema almacena la información.</p> <p>El Sistema envía un mensaje del éxito de la</p>

	operación.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
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	<p>4b.1 El Sistema muestra un mensaje: "Los datos enviado son incorrectos"</p> <p>4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.</p>
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

Caso de Uso:	Generar Reporte de Record de Barrena
Actores:	Supervisor de Pozo
Resumen:	Este caso de Uso se encarga de Generar el Reporte de Record de Barrenas
Precondiciones:	Que el usuario este autenticado previamente en el sistema
Referencias	RF 2.4.1, RF 2.4.2

Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Supervisor de Pozo selecciona la opción Generar Reporte de Record de Barrenas	1.1 El Sistema busca los datos referentes al Reporte de Record de Barrenas 1.2 El Sistema muestra los datos en un formulario.(Estos datos no pueden ser modificador por el actor) 1.3 El Sistema muestra un mensaje del éxito de la operación.
<i>Prototipo de Interfaz</i>	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo	1a.1 "No existen todos los datos para generar el reporte". Muestra un mensaje de error de la operación. "Debe Gestionar antes Generar este Reporte"
<i>Prototipo de Interfaz</i>	
Poscondiciones	Muestra el Reporte

Caso de Uso:	Gestionar Reporte Composición de Herramienta
Actores:	Supervisor de Pozo
Resumen:	Este caso de Uso se encarga de Gestionar el Reporte Composición de Herramientas
Precondiciones:	Que el usuario este autenticado previamente en el sistema
Referencias	RF 2.5.1, RF 2.5.2
Prioridad	Crítico
Flujo Normal de Eventos	

Sección “Insertar Datos del Reporte de Composición de Herramienta”	
Acción del Actor	Respuesta del Negocio
<ol style="list-style-type: none"> 1. El Supervisor de Pozo selecciona la opción de Insertar Datos del Reporte de Composición de Herramienta 2. El Supervisor de Pozo inserta los Datos. 3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar 4. El Supervisor de Pozo confirma la operación de Guardar y Enviar 	<ol style="list-style-type: none"> 1.1 El Sistema muestra un formulario con los datos que debe contener el reporte. 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una venta emergente pidiendo confirmación de la operación 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 2: “Actualizar Datos del Reporte de Composición de Herramienta”	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El Supervisor de Pozo selecciona la opción de Actualizar Datos del Reporte de Composición de Herramientas 2. El Supervisor de Pozo actualiza los Datos. 3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar 4. El Supervisor de Pozo confirma la operación de Guardar y Enviar 	<ol style="list-style-type: none"> 1.1 El Sistema muestra un formulario con todos los datos que se encuentran almacenados en el reporte. 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una venta emergente pidiendo confirmación de la operación 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.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
2a. Inserta los Datos o Actualiza los Datos	2a.1 "Datos introducidos incorrectos" El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo 2a.2 El Sistema limpia todos los caracteres extraños.
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" 4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

Caso de Uso:	Generar Reporte de Distribución del Tiempo Diario.
Actor:	Supervisor de Pozo.
Resumen:	Este caso de uso tiene como función generar automáticamente el reporte de Distribución de Tiempo Diario
Precondiciones:	El usuario debe estar previamente autenticado en el sistema

Referencias	R 2.8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo selecciona la opción de Generar Reporte de Distribución de Tiempo Diario.	<p>1.1 El Sistema busca los datos relacionados con los tiempos empleados en las actividades del pozo de esa fecha</p> <p>1.2 El Sistema muestra en un formulario los datos del reporte. Estos datos no pueden ser modificados por el actor.</p> <p>1.3 El Sistema muestra un mensaje del éxito de la operación.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo	1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Prototipo de Interfaz	
Poscondiciones	Muestra el Reporte

Caso de Uso:	Generar Reporte de Distribución del Tiempo por Etapa.
Actor:	Supervisor de Pozo.
Resumen:	Este caso de uso tiene como función generar automáticamente el Reporte de Distribución de Tiempo por Etapas.
Precondiciones:	Se debe tener en el sistema los Reportes de Distribución de Tiempo Diarios.

Referencias	RF 2.9
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo selecciona la opción Generar Reporte de Distribución de Tiempo por Etapas	<p>1.1 El Sistema recopila los de distribución del tiempo en la etapa que culmina.</p> <p>1.2 El Sistema visualiza los datos en un formulario. El actor no puede tener la posibilidad de cambiar ningún dato de este reporte.</p> <p>1.3 El Sistema muestra un mensaje del éxito de la operación</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo	1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Prototipo de Interfaz	
Poscondiciones	Muestra el Reporte

Caso de Uso:	Gestionar Reporte de Presupuesto Diario
Actor:	Supervisor de Pozo.
Resumen:	Este caso de uso tiene como función la gestión de la información de los gastos diarios en el pozo
Precondiciones:	El usuario debe estar previamente autenticado en el sistema
Referencias	RF 2.10.1 RF 2.11.2

Prioridad Crítico

Flujo Normal de Eventos

Sección 1: Insertar Datos del Presupuesto Diario.

Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo selecciona la opción de Insertar Datos del Presupuesto Diario	1.1 El Sistema muestra un formulario con los datos que debe contener el reporte.
2. El Supervisor de Pozo inserta los Datos.	2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación
3. El Supervisor de Pozo selecciona la opción 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.
4. EL Supervisor de Pozo confirma la operación de Guardar y Enviar	

Sección 2: Actualizar Datos del Reporte de Presupuesto Diario.

Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo selecciona la opción de Actualizar Datos del Presupuesto Diario	1.1 El Sistema muestra un formulario con todos los datos que se encuentran almacenados en el reporte.
2. El Supervisor de Pozo actualiza los Datos	2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar
3. El Supervisor de Pozo selecciona la opción de Guardar y Enviar	3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación
4. El Supervisor de Pozo confirma la operación de Guardar y Enviar	4.1 El Sistema comprueba que los datos no contengan errores

	<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>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
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	<p>4b.1 El Sistema muestra un mensaje: "Los datos enviado son incorrectos"</p> <p>4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.</p>
Prototipo de Interfaz	
Poscondiciones	Redirecciona al Interfaz Principal

Caso de Uso:	Generar Reporte de Presupuesto Semanal.
Actor:	Supervisor de Pozo.
Resumen:	Este caso de uso tiene como función generar automáticamente un resumen de los costos semanales en el pozo.
Precondiciones:	El Usuario debe estar previamente autenticado en el sistema.

Referencias	RF 2.11
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo selecciona la opción Generar Reporte de Presupuesto Semanal	<p>1.1 El Sistema recopila datos de los costos del pozo en la semana que termina.</p> <p>1.2 El Sistema visualiza los datos en un formulario. El actor no puede tener la posibilidad de cambiar ningún dato de este reporte.</p> <p>1.3 El Sistema muestra un mensaje del éxito de la operación</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo	1a.1 "No existen todos los datos para generar el reporte". Muestra un mensaje de error de la operación. "Debe Gestionar antes Generar este Reporte"
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

Caso de Uso:	Generar Resumen de Costo.
Actor:	Supervisor de Pozo.
Resumen:	Este caso de uso tiene como función de generar automáticamente el Resumen de Costos del Pozo.
Precondiciones:	Se debe tener en el sistema el Reporte de Presupuesto Semanal.
Referencias	RF 2.12
Prioridad	Crítico

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Supervisor de Pozo Selecciona la opción Generar Resumen de Costo	1.1 El Sistema recopila los Datos del Reporte de Distribución del Tiempo por Etapas. 1.2 El Sistema genera el Reporte de Distribución del Tiempo por Etapas 1.3 El Sistema visualiza el reporte. El usuario no puede tener la posibilidad de cambiar ningún dato de este reporte.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1a. El Supervisor de Pozo selecciona la opción Reporte Diario Operativo	1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

3.4.3.2 Subsistema DIPP

Caso de Uso:	Generar Parte de Perforación
Actor:	Secretaria de Despacho
Resumen:	Este caso de uso se encarga de generar automáticamente el Parte Diario de Perforación, sin necesidad de que el usuario inserte ningún dato.
Precondiciones:	Reporte Diario Operativo del Pozo. El actor debe estar autenticado previamente en el sistema.
Referencias	RF 2.25

Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. La Secretaria de Despacho selecciona la opción de Generar Parte Diario de Perforación.		<p>1.1 El Sistema recopila los Datos del Parte Diario de Perforación</p> <p>1.2 El Sistema genera el Parte Diario de Perforación.</p> <p>1.3 El Sistema visualiza el Parte Diario de Perforación. El usuario no puede tener la posibilidad de cambiar ningún dato de este reporte.</p>
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor		Respuesta del Negocio
1a. La Secretaria de Despacho selecciona la opción Reporte Diario Operativo		1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Poscondiciones	Muestra el Reporte	

Caso de Uso:	Gestionar Parte Intervención
Actor:	Secretaria de Despacho
Resumen:	Este caso de uso se encarga de la gestión de la información generada diariamente en los pozos en intervención.
Precondiciones:	<p>La información recopilada de los pozos todos los pozos que se encuentran en intervención.</p> <p>Se debe de estar autenticados previamente en el sistema.</p>

Referencias	RF 2.26
Prioridad	Crítico
Flujo Normal de Eventos	
Sección 1: Insertar Datos del Parte Diario de Intervención	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. La Secretaria de Despacho selecciona la opción de Insertar Datos del Parte de Intervención 2. La Secretaria de Despacho inserta los Datos. 3. La Secretaria de Despacho selecciona la opción de Guardar y Enviar 4. La Secretaria de Despacho confirma la operación de Guardar y Enviar 	<ol style="list-style-type: none"> 1.1 El Sistema muestra un formulario con los datos que debe contener el reporte. 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación 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 2: Actualizar Datos del Parte de Intervención	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. La Secretaria de Despacho selecciona la opción de Actualizar Datos del Parte Diario de Intervención 2. La Secretaria de Despacho actualiza los Datos 3. La Secretaria de Despacho selecciona la opción de Guardar y Enviar 4. La Secretaria de Despacho confirma la operación de Guardar y Enviar 	<ol style="list-style-type: none"> 1.1 El Sistema muestra un formulario con todos los datos que se encuentran almacenados en el reporte. 2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar 3.1 El Sistema muestra una ventana emergente pidiendo confirmación de la operación

	<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>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
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	<p>4b.1 El Sistema muestra un mensaje: "Los datos enviados son incorrectos"</p> <p>4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.</p>
Prototipo de Interfaz	
Poscondiciones	Redirecciona a la Interfaz Principal

Caso de Uso:	Gestionar Parte Diario Operativo de Perforación
Actor:	Secretaria de Despacho
Resumen:	Este caso de uso se encarga de la gestión de la información generada diariamente en los pozos en intervención, perforación, otros partes.

Precondiciones:	La información recopilada de los pozos todos los pozos que se encuentran en intervención. Se debe de estar autenticados previamente en el sistema.
Referencias	R 2.27
Prioridad	Crítico

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. La Secretaria de Despacho selecciona la opción de Gestionar Parte de Intervención 2. La Secretaria de Despacho selecciona la opción de Generar Parte de Perforación 3. La Secretaria de Despacho selecciona la opción de Generar Partes 	

Sección 1 “Gestionar Parte de Intervención”

Acción del Actor	Respuesta del Sistema
	1.1 El Sistema invoca al CU Gestionar Parte de Intervención

Sección 2 “Generar Parte de Perforación”

Acción del Actor	Respuesta del Sistema
	2.1 El Sistema Invoca al CU Generar Parte de Perforación .

Sección 3 “Generar Partes Restantes”

Acción del Actor	Respuesta del Sistema
1. La Secretaria de la DIPP selecciona la opción de Generar Partes Restantes.	<ol style="list-style-type: none"> 1.1 El Sistema recopila los Datos de los Partes Restantes 1.2 El Sistema genera los partes 1.3 El Sistema visualiza los partes. El usuario no puede tener la posibilidad de

	cambiar ningún dato de este reporte.
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
1a. La Secretaria de Despacho selecciona la opción Reporte Diario Operativo	1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Poscondiciones	Muestra el Reporte

3.4.3.3 Subsistema CEINPET

Caso de Uso:	Generar Parte Diario de Geología
Actores:	Geólogo del CEINPET
Resumen:	La función de este requisito es la gestión del Parte Diario de Geología, el cual no es más que la recopilación de las informaciones geológicas de todos los pozos en perforación.
Precondiciones:	Haber obtenido los Reportes Diario de Perforación de Cada Pozo. El usuario debe estar previamente autenticado en el sistema
Referencias	RF 2.28
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El Geólogo del CEINPET selecciona la opción de Generar el Parte Diario de Geología	1.1 El Sistema recopila los Datos del Parte Diario de Geología 1.2 El Sistema genera el Parte Diario de

	<p>Geología</p> <p>1.3 El Sistema visualiza el Parte Diario de Geología. El usuario no puede tener la posibilidad de cambiar ningún dato de este reporte.</p>
Flujos Alternos	
Acción del Actor	Respuesta del sistema
1a. La Secretaria de Despacho selecciona la opción Reporte Diario Operativo	1a.1 “No existen todos los datos para generar el reporte”. Muestra un mensaje de error de la operación. “Debe Gestionar antes Generar este Reporte”
Prototipo de Interfaz	
Poscondiciones	

3.4.3.4 Subsistema Seguridad

Caso de Uso:	Autenticar Usuarios
Actor:	Usuario
Resumen:	Este requisito es uno de los encargados de la seguridad del sistema; es el encargado de proporcionar el acceso inicial al sistema solo a aquellos usuarios registrados.
Precondiciones:	Se Muestra un formulario solicitando usuario y contraseña
Referencias	RF 2.33
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario entra su id de usuario y la contraseña.	1.1 El Sistema Verifica que los datos introducidos no contengan errores

<p>2. El usuario selecciona la opción Entrar</p>	<p>1.2 El Sistema habilita la opción Enviar</p> <p>2.1 El Sistema comprueba que los datos estén correctos</p> <p>2.2 El Sistema consulta la base de datos para chequear si el usuario existe.</p> <p>2.3 El sistema verifica que la contraseña sea la correcta.</p> <p>2.4 El Sistema consulta el rol del usuario</p> <p>2.5 El Sistema consulta el los permisos del usuario</p> <p>2.5 El Sistema muestra un mensaje de bienvenida a la aplicación.</p>
--	--

Prototipo de Interfaz

Flujos Alternos

Acción del Actor	Respuesta del Sistema
<p>1a. Entra su id de usuario y la contraseña.</p>	<p>1a.1 El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo</p> <p>1a.2 El Sistema limpia todos los caracteres extraños.</p>
<p>1b. El usuario o las contraseña son incorrectos</p>	<p>1b.1 El Sistema muestra un mensaje: "El usuario o la contraseña son incorrectos"</p> <p>1b.2 El Sistema limpia los campos del usuario y la contraseña.</p>

Prototipo de Interfaz

<p>Poscondiciones</p>	<p>Se le muestra al usuario la interfaz correspondiente al rol asignado, así como las funcionalidades correspondientes con los permisos correspondientes.</p>
------------------------------	--

Caso de Uso:	Gestionar Usuarios
Actor:	Administrador del Sistema
Resumen:	Este requisito es uno de los encargados de la seguridad del sistema; tiene como funcionalidad el trabajo con los datos de los usuarios que autorizados a trabajar con el sistema.
Precondiciones:	El actor debe estar autenticado como administrador del sistema
Referencias	RF 2.31
Prioridad	Crítico

Flujo Normal de Eventos

Sección 1: Insertar Datos del Usuario

Acción del Actor	Respuesta del Sistema
1. El Administrador de Sistema selecciona la opción de Insertar Datos del Usuario	1.1 El Sistema muestra un formulario con los con los campos donde se va a insertar los datos del usuario.
2. El Administrador del Sistema inserta los datos.	2.1 El Sistema verifica si los datos son correctos 2.2 El Sistema habilita la opción de Guardar y Enviar
3. El Administrador del Sistema selecciona la opción de Guardar y Enviar	3.1 El Sistema muestra una ventana emergente solicitando confirmación de la operación
4. El Administrador del Sistema confirma la operación de Guardar y Enviar	4.1 El Sistema comprueba que los datos no contengan errores. 4.2 El Sistema almacena los datos en la base de datos 4.3 El Sistema muestra un mensaje del éxito

de la operación.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
2a. Inserta los Datos	2a.1 "Datos introducidos incorrectos" El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo 2a.2 El Sistema limpia todos los caracteres extraños.
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" 4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.

Sección 2: Actualizar Usuario

Acción del Actor	Respuesta del Sistema
1. El Administrador del Sistema selecciona la opción de Actualizar Usuario	1.1 El Sistema muestra un Formulario donde se selecciona el ID del Usuario
2. El Administrador del Sistema selecciona el usuario a actualizar	2.1 El Sistema muestra un formulario con todos los datos del usuario.
3. El Administrador del Sistema actualiza los datos del usuario	3.1 El Sistema habilita la opción de Guardar y Enviar
4. El Administrador del Sistema selecciona la opción de Guardar y Enviar	4.1 El Sistema muestra una ventana emergente solicitando confirmación de la operación.
5. El Administrador del Sistema confirma la operación de Guardar y Enviar	5.1 El Sistema verifica que los datos estén correctos 5.2 Almacena los Datos Actualizados

5.3 Muestra un mensaje del éxito de la operación.

Flujos Alternos

Acción del Actor	Respuesta del Sistema
3a. Actualiza los Datos	3a.1 “Datos introducidos incorrectos” El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo 3a.2 El Sistema limpia todos los caracteres extraños.
5a. No confirma la operación de Guardar y Enviar	5a.1 El Sistema no envía los datos
5b. Confirma la operación de Guardar y Enviar	5b.1 El Sistema muestra un mensaje: “Los datos enviado son incorrectos” 5a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.

Sección 3: Eliminar Usuario

Acción del Actor	Respuesta del Sistema
1. El Administrador del Sistema selecciona la opción de Eliminar Usuario 2. El Administrador del Sistema selecciona el Usuario 3. El Administrador del Sistema selecciona la opción Eliminar. 4. El Administrador del Sistema confirma la operación	1.1 El Sistema muestra un formulario donde se selecciona el Usuario 2.1 El Sistema habilita la opción de Eliminar 3.1 El Sistema muestra un ventana emergente solicitando confirmación de la operación 4.1 El Sistema Elimina todos los datos del usuario de la base de datos 4.2 El Sistema muestra un mensaje del éxito de la operación

Prototipo de Interfaz

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
4a. No confirma la operación de Guardar y Enviar	5a.1 El Sistema no envía los datos
Poscondiciones	Redirecciona a la Interfaz Principal

3.4.3.5 Subsistema Visualización de Información

Caso de Uso:	Generar Gráficas de Reportes
Actores:	Oficina de Supervisores
Resumen:	El caso de uso se inicia cuando se visualiza la información de los reportes en gráficas con la cual se dará la posibilidad de mostrar la información de otra manera.
Precondiciones:	Usuarios del Sistema ya Autenticados.
Referencias	RF 2.37
Prioridad	Secundario
Flujo Normal de Eventos	
Sección “Generar Graficas”	
Acción del Actor	Respuesta del Sistema
1. Solicita ver la información de los reportes de forma gráfica.	1.1 El sistema muestra una interfaz para seleccionar: <ul style="list-style-type: none"> Tipo de Gráfica: Lineal – Barra – Pastel Reporte a Graficar
2. Selecciona el tipo de gráfico, y el reporte	2.1 El sistema busca los datos del reporte
3. Selecciona la opción Graficar	2.2 El sistema muestra la gráfica.
Prototipo de Interfaz	

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<i>Prototipo de Interfaz</i>	
Poscondiciones	Se genera las graficas correspondiantes.

Caso de Uso:	Visualizar Información de Reportes
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el actor necesita visualizar información persistente referente a un reporte del pozo.
Precondiciones:	Usuarios del Sistema ya Autenticados.
Referencias	RF 2.38
Prioridad	Secundario

Flujo Normal de Eventos	
Sección “Visualizar Información de Reportes”	
Acción del Actor	Respuesta del Sistema
1. El usuario solicita visualizar la información de un reporte.	1.1 El sistema Muestra al usuario los reportes.
2. El usuario selecciona el reporte a visualizar.	2.1 El sistema visualiza el reporte seleccionado.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
<i>Prototipo de Interfaz</i>	
Poscondiciones	Se visualiza el reporte deseado.

Caso de Uso:	Buscar Información General	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el actor necesita buscar cualquier información persistente referente a los pozos, ya sea de los diferentes reportes y cronogramas que se generan.	
Precondiciones:	Usuarios del Sistema ya Autenticados.	
Referencias	RF 2.39	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección “Buscar Información General”		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona “Búsqueda General”	1.1 El sistema muestra una interfaz con un formulario, donde se puede introducir la cadena de caracteres.	
2. El usuario introduce la cadena de caracteres.	2.1 El Sistema verifica la cadena de caracteres introducida 2.2 El sistema habilita la opción Buscar	
3. El usuario selecciona la opción Buscar	3.1 El sistema comprueba que la cadena enviada esté correcta 3.2 El sistema busca la información 3.3 El muestra el resultado de la búsqueda.	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
2a. Introduce la cadena de caracteres.	2a.1 “Datos introducidos incorrectos” El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo	

	2a.2 El Sistema limpia todos los caracteres extraños.
3a. Selecciona la opción Buscar	3a.1 El Sistema muestra un mensaje: “Los datos enviados son incorrectos” 3a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.
Prototipo de Interfaz	
Poscondiciones	Se obtiene la información solicitada.

Caso de Uso:	Buscar Información Específica	
Actores:	Usuario	
Resumen:	El caso de uso se inicia cuando el actor necesita buscar cualquier información persistente referente a los pozos, ya sea de los diferentes reportes y cronogramas que se generan.	
Precondiciones:	Usuarios del Sistema ya Autenticados.	
Referencias	RF 2.39	
Prioridad	Secundario	
Flujo Normal de Eventos		
Sección “Buscar Información Específica”		
	Acción del Actor	Respuesta del Sistema
	1. El usuario selecciona “Búsqueda Avanzada”	1.1 El sistema muestra una interfaz con un formulario con varios criterios de búsqueda
	1. El usuario selecciona los criterios.	2.1 Muestra campos de texto, según los criterios seleccionados
	2. El usuario introduce los datos a buscar	3.1 Verifica que los datos estén correctos 3.2 Habilita la opción Buscar
	3. El usuario selecciona la opción Buscar	2.1 El Sistema comprueba que los datos estén correctos

	<p>3.1 El sistema busca la información</p> <p>3.2 El sistema muestra los resultados de la búsqueda.</p>
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3a. Introduce los datos a buscar	<p>3a.1 "Datos introducidos incorrectos" El Sistema muestra un mensaje de que no se pueden introducir letras/números/caracteres(&,% , etc) en ese campo</p> <p>3a.2 El Sistema limpia todos los caracteres extraños.</p>
4a. Selecciona la opción Buscar	<p>4a.1 El Sistema muestra un mensaje: "Los datos enviados son incorrectos"</p> <p>4a.2 El Sistema colorea en color rojo los campos que contienen datos incorrectos.</p>
Prototipo de Interfaz	
Poscondiciones	Se obtiene la información solicitada.

3.4.3.6 Subsistema Trabajo con Archivos

Caso de Uso:	Gestionar Trabajo con Archivo
Actores:	Usuario
Resumen:	Este caso de uso se encarga de Importar, Exportar y Guardar archivos de diferentes formatos.
Precondiciones:	El usuario debe estar autenticado en el sistema.

Referencias	RF 2.40
Prioridad	Secundario
Flujo Normal de Eventos	
Sección 1 : Importar Archivo	
Acción del Actor	Respuesta del Sistema
<p>1. El Usuario selecciona la opción de Importar Archivo</p> <p>2. El Usuario selecciona el tipo de archivo a importar</p> <p>3. El Usuario carga el archivo</p> <p>4. El Usuario selecciona la opción de Importar</p>	<p>1.1 El Sistema muestra una interfaz con un menú para escoger el tipo de archivo que va a importar:</p> <ul style="list-style-type: none"> • Microsoft Exel • TXT • PDF <p>2.1 El Sistema muestra una interfaz para cargar el archivo</p> <p>4.1 El Sistema importa el archivo</p> <p>4.2 El Sistema muestra el archivo importado</p> <p>4.3 El Sistema guarda los datos contenido en el archivo</p> <p>4.4 El Sistema muestra un mensaje del éxito de la operación.</p>
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Acción del Actor
3a. Error al cargar el archivo	3a.1 Muestra un mensaje: "El archivo está dañado o contiene errores"
Sección 2 : Exportar Archivo	
Acción del Actor	Respuesta del Sistema
1. El Usuario selecciona la opción de Exportar Archivo	1.1 El Sistema muestra una interfaz para seleccionar el Reporte que desea exportar a:

<ol style="list-style-type: none"> 2. El Usuario selecciona reporte 3. El Usuario selecciona el tipo, que desea exportar el reporte 4. El Usuario selecciona la ubicación en el PC 5. El Usuario selecciona la opción de exportar 	<ul style="list-style-type: none"> • Microsoft Exel • TXT • PDF <p>5.1 El Sistema exporta el Reporte</p> <p>5.2 El Sistema muestra un mensaje del éxito de la operación.</p>
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Prototipo de Interfaz	
Poscondiciones	Redirecciona hacia la interfaz principal

3.5 Patrones de Casos de Uso

Luego de haber abordado el tema de los Patrones en el primer capítulo de este trabajo, definiendo concepto, necesidad y los tipos, en este epígrafe se expondrá los Patrones de Casos de Uso, con un enfoque más práctico, centrados en los que se utilizaron para lograr el Modelo de Casos de Uso, artefacto perteneciente de al Flujo de Trabajo de Requerimientos.

Los patrones de utilizados son

- **CRUD (Creating, Reading, Updating, Deleting):** Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual. *Parcial:* El patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es

preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras. [Ver Anexo 20](#)

- **Múltiples Actores Roles Comunes:** Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso. [Ver Anexo 21](#)
- **Concordancia (Commonality):** Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

Especialización: Otro patrón de concordancia que contiene casos de uso del mismo tipo. En este caso, estos son modelados como una especialización de casos de uso de tipo de uso común. Todas las acciones en estos casos de uso son heredadas por los casos de uso hijos, donde otras acciones serán adicionadas o acciones heredadas que serán especializadas. Este patrón es aplicable cuando la utilización de los casos de uso que han sido modelados son del mismo tipo, y este tipo debe hacerse visible en el modelo. [Ver Anexo 22](#) (39)

- **Concrete Extension o Inclusion:**

Inclusion: En este modelo, se incluye una relación de la base de caso de uso a la inclusión de casos de uso. Esta última instancia puede ser por su propia cuenta. La base de caso de uso puede ser concreto o abstracto. [Ver Anexo 23](#) (40)

El sistema propuesto, está destinado principalmente para la gestión de información, por tal razón, se decide aplicar el patrón CRUD Parcial, ya que las funcionalidades de Eliminar (Deleting) y Buscar (Reading), se separaron del funcionamiento de los casos de uso. En el caso específico de Eliminar, por requerimientos del cliente ninguna información Inserte (Creating), debe poder eliminarse, esta restricción donde único no se aplica es en el Subsistema de Seguridad, en el CU de Gestionar los Usuarios donde se le da la posibilidad al Administrador del Sistema de Eliminar Usuario. En el caso de Buscar, se separó y se agrupan todas las funciones vinculadas a la búsqueda de información en el Subsistema de Visualización de Información, en dos CU Buscar Información General y en el CUS Buscar Información Específica. El segundo patrón aplicado es el Múltiples Actores con Roles Comunes, su utilización viene dada por la existencia de varios casos de uso que eran inicializados por más de un actor que

desempeñan el mismo rol, al inicializar el caso de uso; la aplicación de este patrón se evidencia en el funcionamiento del Subsistema Seguridad, Visualización de Información y Trabajo con Archivos. Luego de lograr los diagramas de casos de uso del sistema, se dispone a aplicar el patrón Concordancia el cual nos posibilita la identificación de subsecuencias dentro de un caso de uso, que pueden ser expresados como un caso de uso por separado, en el CU Gestionar Reporte de Encamisado, perteneciente al Subsistema Pozo, se aplica la propiedad de Especialización. El ultimo patrón utilizado es el Concrete Extension o Inclusion: Inclusion específicamente en el CU Gestionar Parte Diario de Perforación, perteneciente al Subsistema DIP. La aplicación de estos patrones de casos de uso se logra una mayor calidad al modelo de caso, además de convertirse en modelos más elocuentes para el cliente, ya que estos modelos representan los requisitos del sistema a modelar.

3.6 Complejidad del Sistema

Luego de culminar la especificación de los CU del sistema, se muestra prudente medir la complejidad del sistema, actividad sumamente importante que contribuye a medir el tiempo y el esfuerzo que necesitamos para construir el sistema, aunque en este trabajo no se presenta el resultado de la estimación, solo se llegará a la conclusión del nivel de complejidad que posee el sistema.

“Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoces algo sobre ello; cuando no puedas medir, cuando no puedas expresar lo que dices con números, tu conocimiento es precario y deficiente.”

Lord Kelvin

Existen varios modelos de estimación de software:

1. Basados en la Experiencia
2. Históricos
3. De Base Estadística
4. Con Base Teórica
5. Compuestos

Incluido dentro del modelo con Base Teórica se encuentran los métodos de Putnam (SLIM), COCOMO, Puntos de Función, etc. Dentro del método por Puntos de Función, encontramos el método de Estimación por puntos de Casos de Uso, el cual es el utilizado para medir la complejidad del sistema que propone este trabajo.

La utilización de este método viene dado debido a la necesidad de conocer la complejidad del sistema en una fase temprana de desarrollo del mismo, lo cual posibilita tener una idea del costo de esfuerzo y tiempo a invertir en las distintas fases del proyecto. Existen otros métodos para calcular la complejidad de un sistema, pero se basan en las líneas de código o en el diseño del mismo, las cuales no fueron valoradas para realizar la estimación de la complejidad del sistema.

Un aspecto a considerar en la estimación es que no debe hacer una sola vez en la vida del proyecto, lo recomendado es estimar varias veces utilizando métodos diversos, en las diferentes fases de trabajo lo cual le proporcionará al equipo una visión cada vez más actual del costo y esfuerzo para construir el sistema.

Análisis por Puntos de Casos de Uso

Tipo de CU	Descripción	Cantidad de CU	Factor	Peso
Simple	3 transacciones o menos	15	5	75
Promedio	de 4 a 8 transacciones	23	10	230
Complejo	8 a infinito.	0	15	0

De los 45 CU identificados fueron descritos 38, el cual fue la muestra tomada para realizar este análisis.

Dado el resultado de llega a la conclusión de que el sistema posee un grado de complejidad promedio.

3.7 Conclusiones Parciales

En este capítulo se definieron las características del sistema obteniéndose las reglas, actores, trabajadores y entidades del negocio, así como los diferentes procesos de negocio representados a través de casos de usos. Se identifican requerimientos funcionales y no funcionales, los cuales se representan a través de diagramas de casos de uso del sistema, junto a la descripción de cada uno de ellos, así como el cálculo de la complejidad del sistema utilizando la técnica de Estimación por Puntos de Casos de Uso

Capítulo IV: Análisis y Diseño del Sistema

4.1 Introducción

En este capítulo se expondrá lo referente al análisis y diseño del Sistema de Manejo Integral de Perforación de Pozos, correspondiente al flujo de Análisis y Diseño de la metodología RUP, metodología empleada para la modelación del sistema. Se realiza una vista al Modelo de Análisis, donde se muestra los diagramas de clases del análisis. También se realiza una vista del Modelo de Diseño, así como los diagramas de clases del diseño y los diagramas de interacción, además de los patrones de diseño utilizados.

4.2 Análisis

4.2.1 Modelo de Análisis

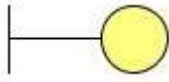
El Modelo de Análisis contribuye a refinar los requerimientos y permite razonar sobre los aspectos internos del sistema y se considera como la primera aproximación al modelo de diseño. El Modelo de Análisis puede o no realizarse para construcción de un sistema, RUP ofrece tres maneras de emplear el modelo de análisis en la construcción de un sistema:

- El proyecto utiliza el modelo de análisis para describir los resultados del análisis y mantiene la consistencia de este a lo largo del ciclo de vida del proyecto.
- El proyecto utiliza el modelo de análisis para describir los resultados del análisis pero considera a este como una herramienta intermedia (es decir de mayor interés durante la Fase de Elaboración).
- EL proyecto no utiliza el modelo de análisis para describir los resultados del análisis.

Para la modelación del sistema en cuestión se adoptado la segunda manera de emplear el modelo de análisis, por tal motivo luego de pasar a la Fase de Construcción se deja de actualizar el modelo de análisis y cualquier cambio que ocurra en el funcionamiento interno del sistema se resuelve como parte integrada al modelo de diseño.

4.2.2 Clases del Análisis

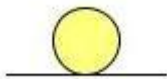
Representan abstracciones de una o varias clases y/o subsistemas del diseño del sistema, las cuales se caracterizan por centrarse en el tratamiento de los requisitos funcionales. Las clases del análisis siempre encajan con tres estereotipos básicos:



Las clases interfaz se utilizan para modelar la interacción entre el sistema y sus actores. Al modelar partes del sistema que implican a sus actores implica que clarifican y reúnen los requisitos en los límites del sistema. Representan abstracciones de de ventanas, formularios, paneles, interfaces de comunicaciones, terminales, entre otros.



Las clases controladoras (control) representan coordinaciones, secuencias, transacciones, y control de objetos y se utilizan con frecuencia para encapsular el control de un caso de uso en concreto. También se utilizan para representar derivaciones y cálculos, como la lógica de negocio, que no pueden asociarse con ninguna información concreta, de larga duración, almacenada por el sistema.



Las clases entidades se utilizan para modelar información que posee una vida larga y que es a menudo persistente. La mayoría de estas clases se derivan de un entidad del negocio.

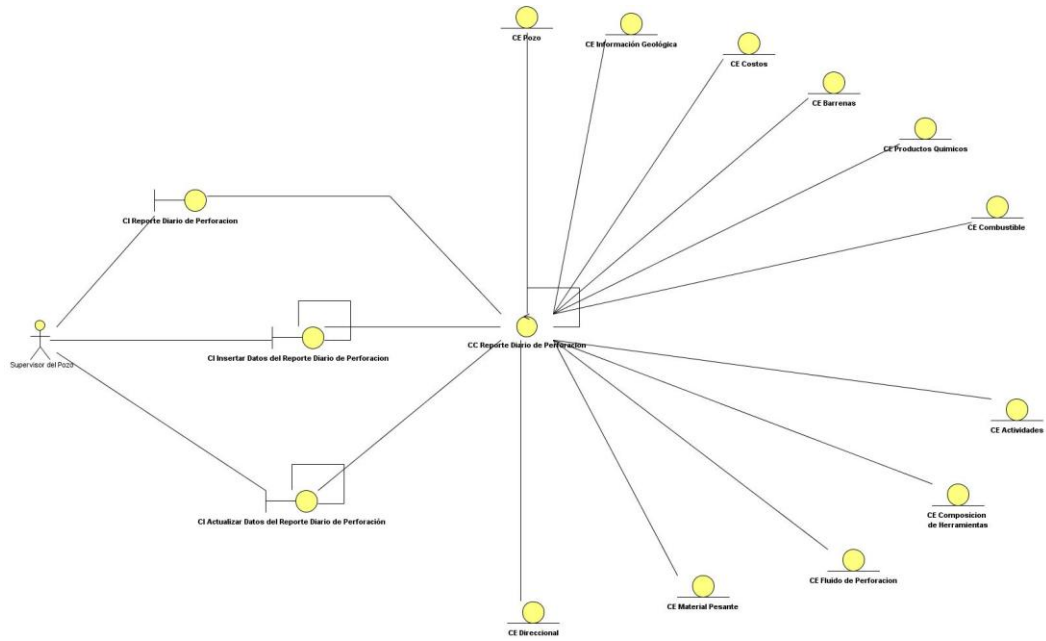
Cada estereotipo implica una semántica específica, lo cual constituye un método potente y consistente de identificar y describir las clases del análisis.

4.2.2.1 Diagramas de Clases del Análisis

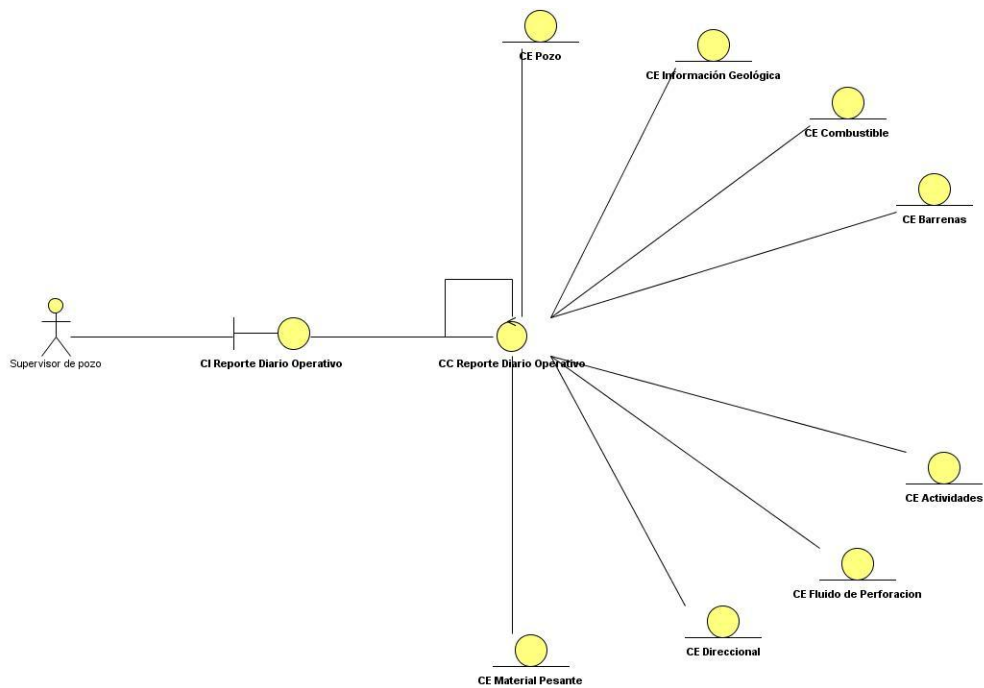
Subsistema Gestión de Reportes

- **Subsistema Pozo**

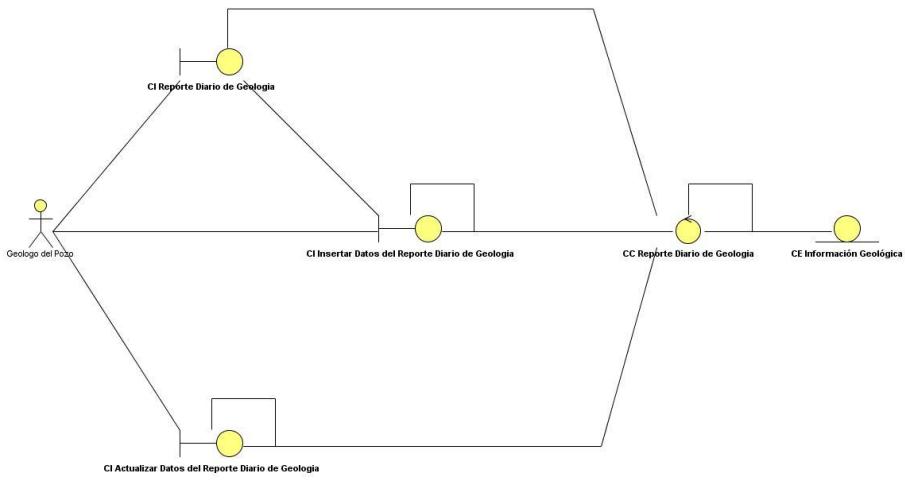
CU Gestionar Reporte Diario de Perforación



CU Generar Reporte Diario Operativo

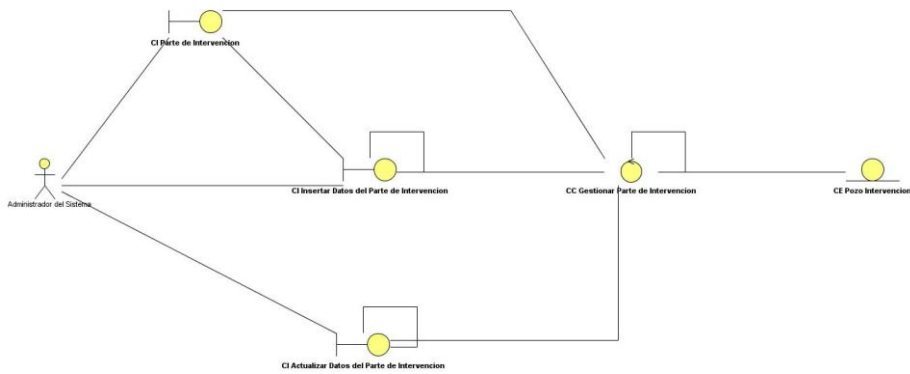


CU Gestionar Reporte de Geología

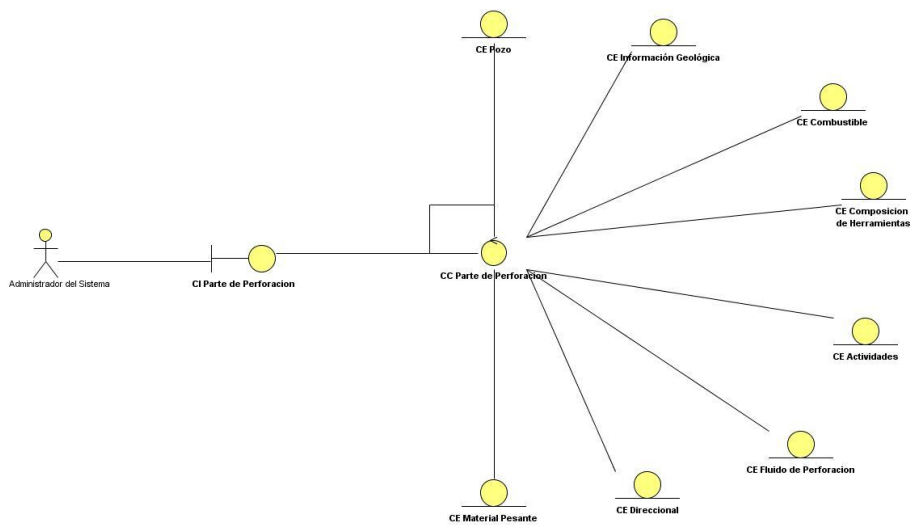


- **Subsistema DIPP**

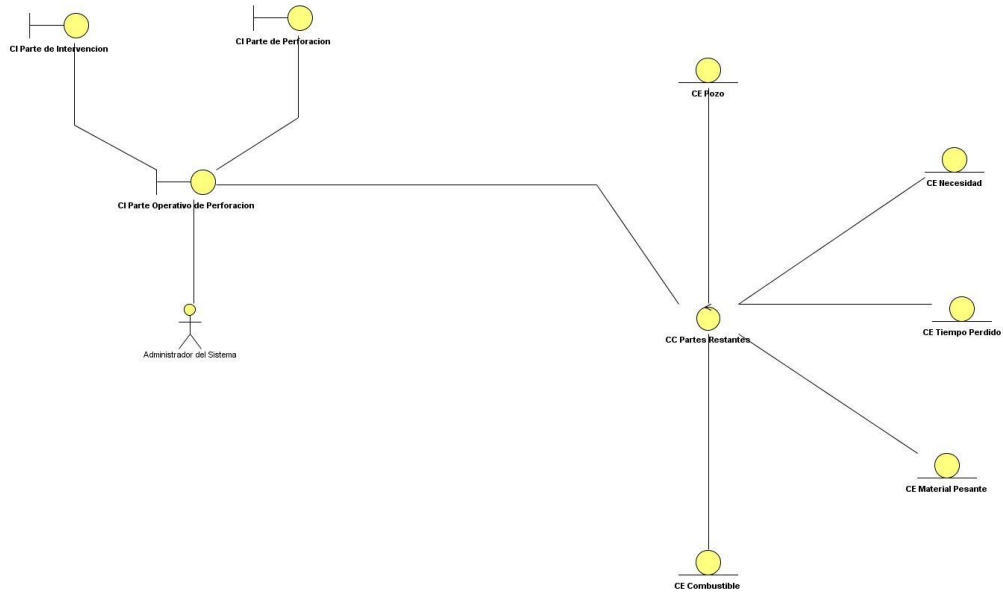
CU Gestionar Parte de Intervención



CU Generar Parte de Perforación

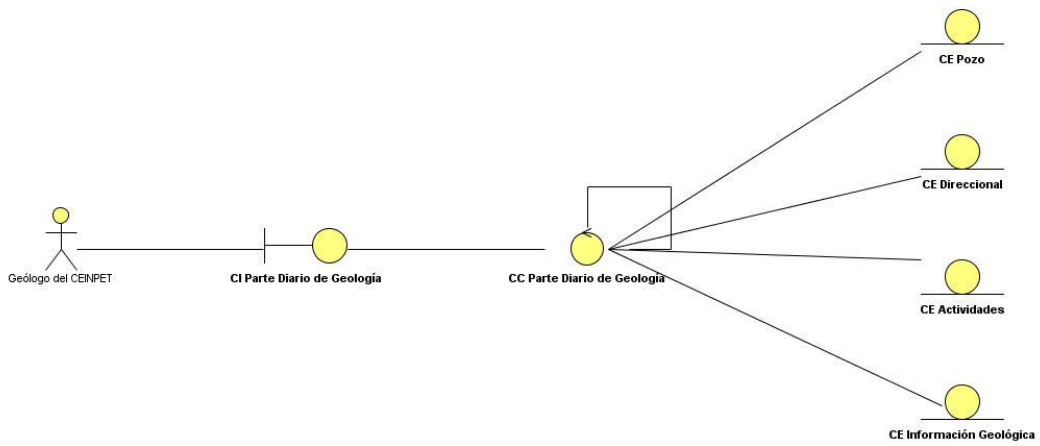


CU Gestionar Parte Operativo de Perforación



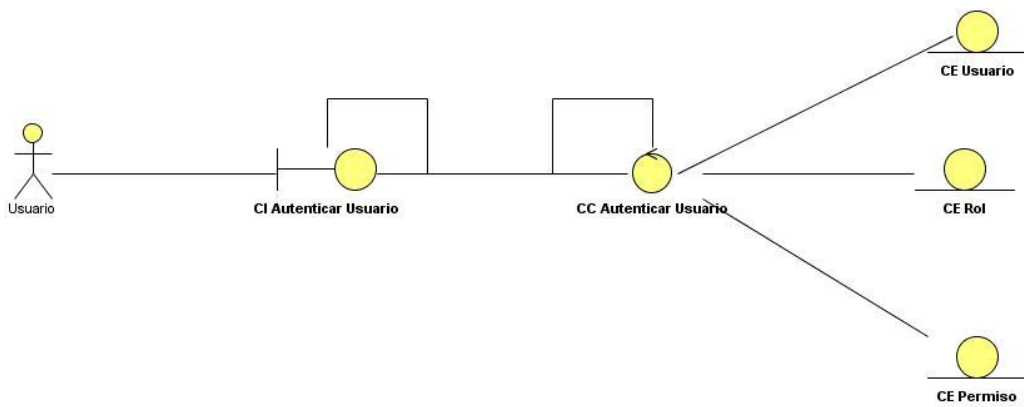
• **Subsistema CEINPET**

CU Gestionar Parte Diario de Geología

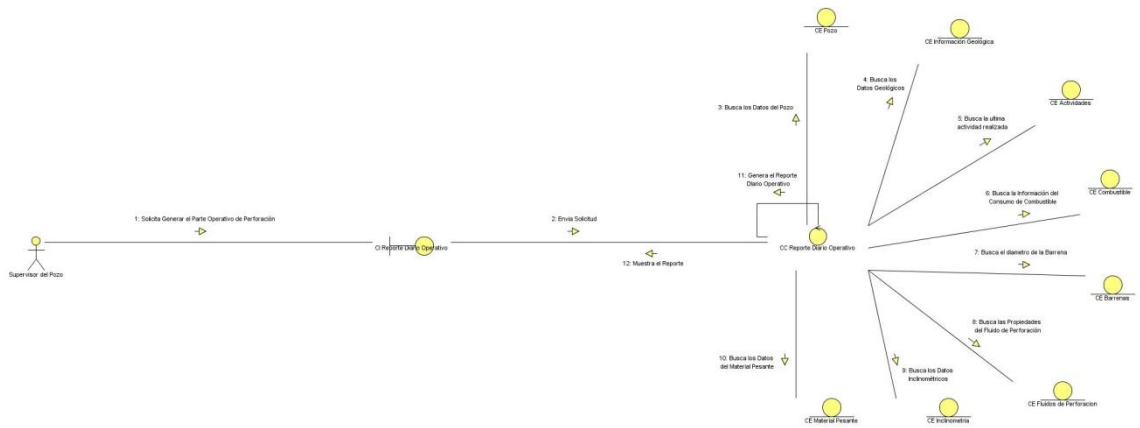


Subsistema de Seguridad

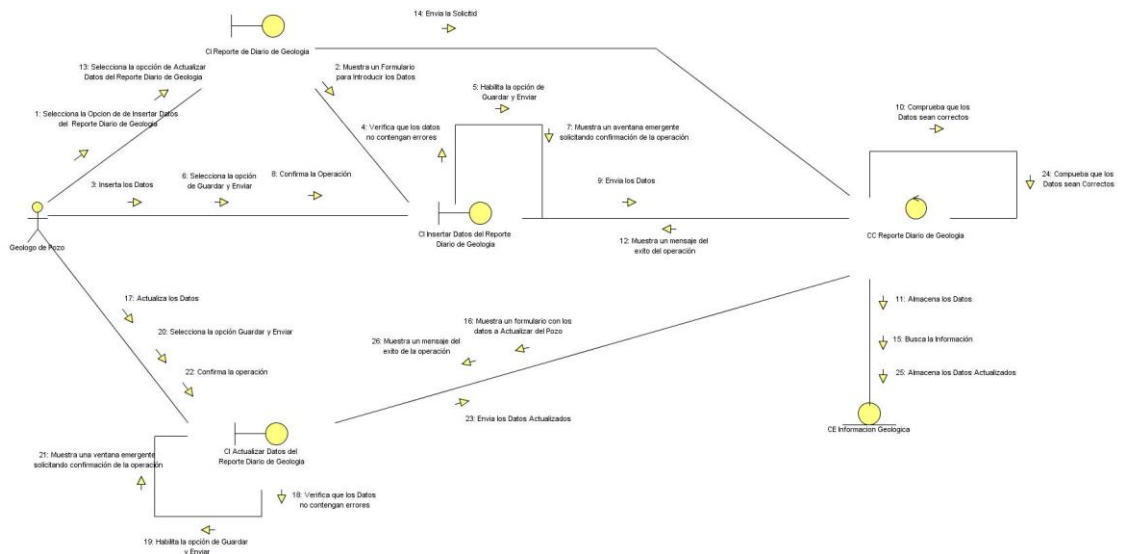
CU Autenticar Usuario



CU Gestionar Usuarios

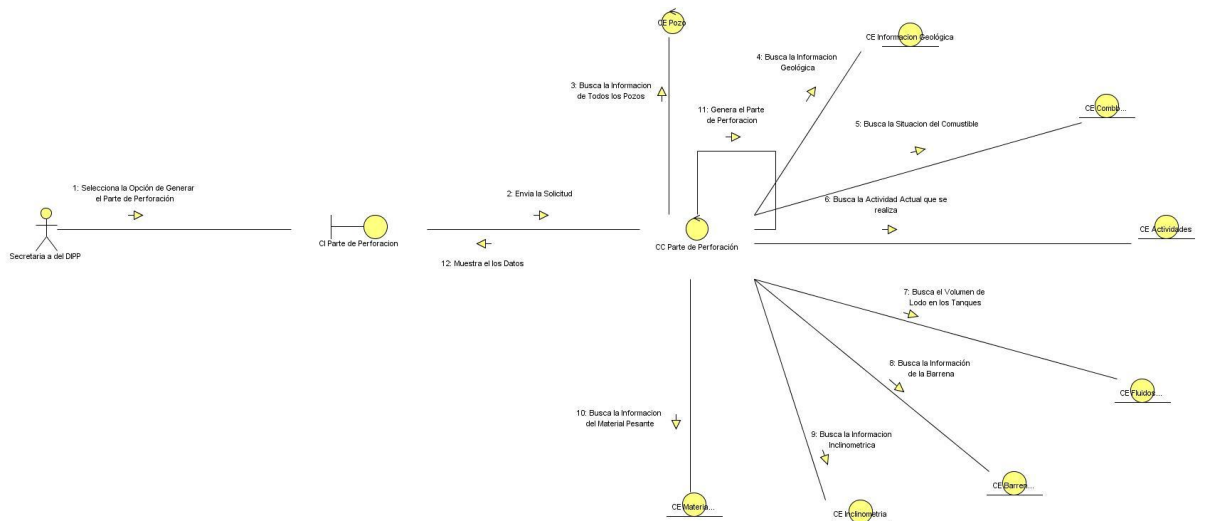


CU Generar Reporte Diario de Geología

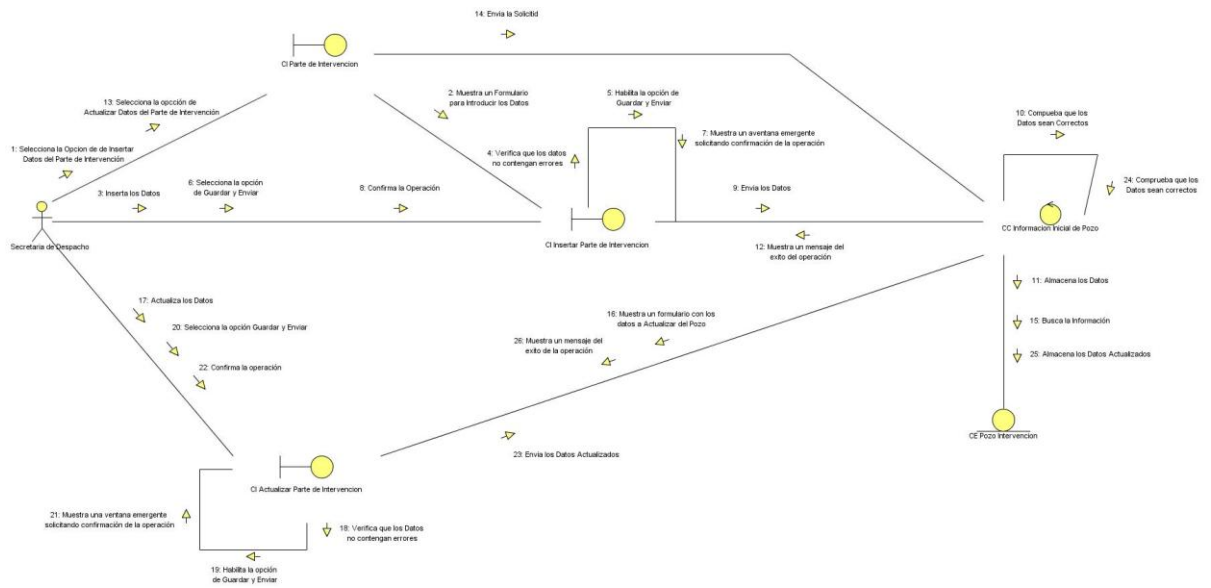


• Subsistema DIPP

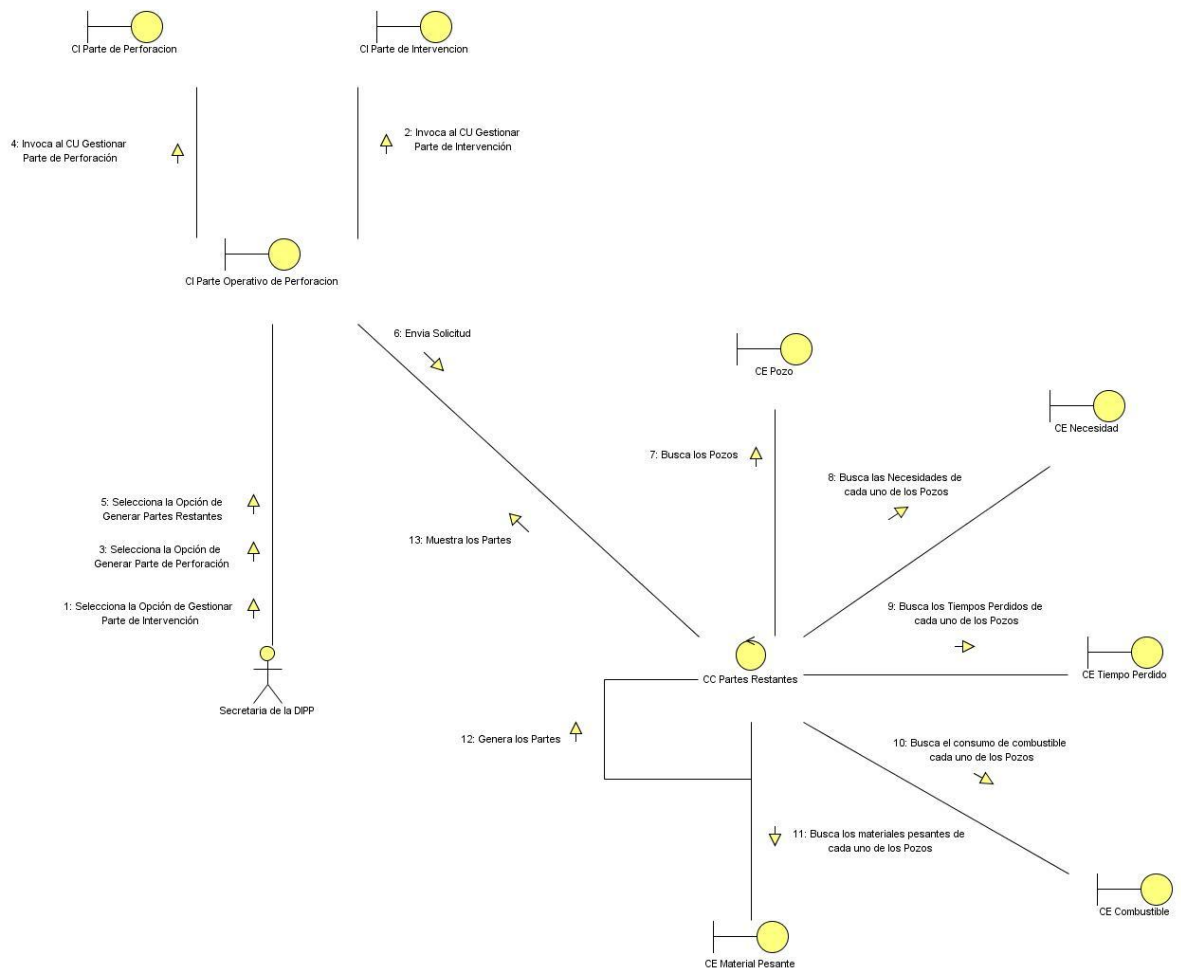
Diagrama de Colaboración del Análisis del CU Generar Parte de Perforación



CU Gestionar Parte de Intervención



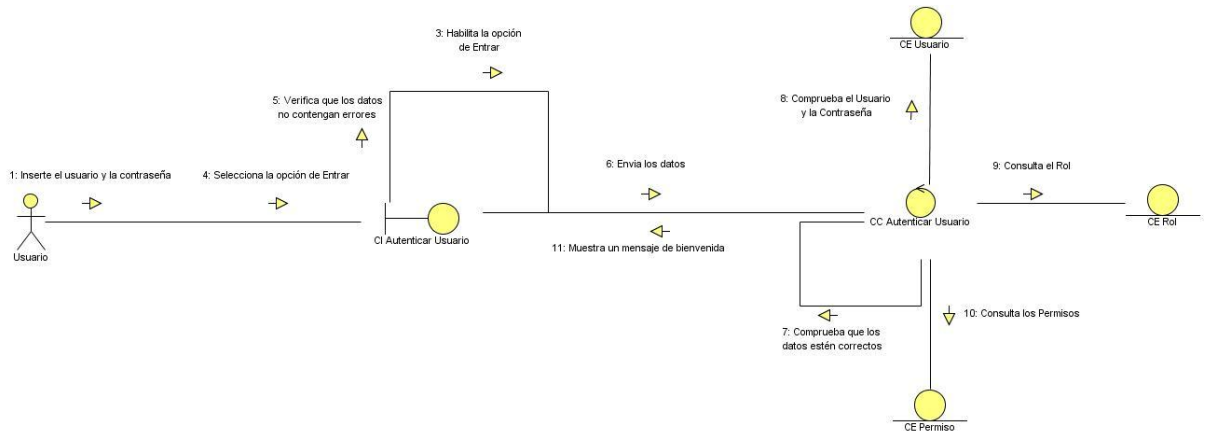
CU Gestionar Parte Diario de Perforación



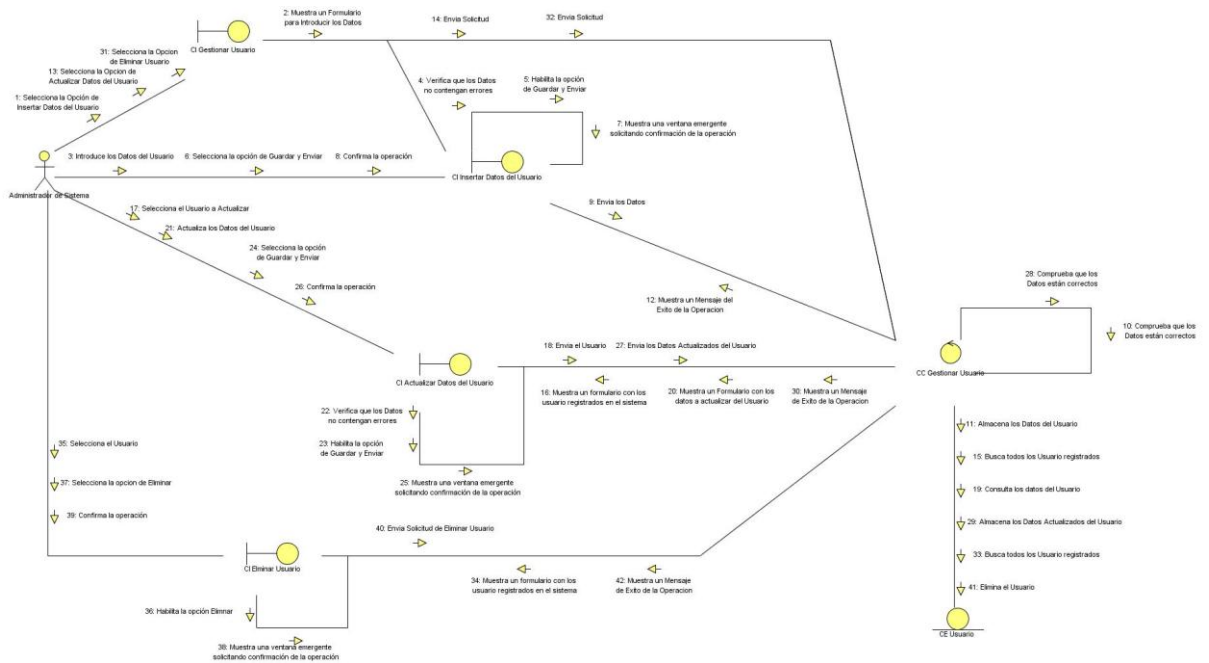
- **Subsistema CEIPET**

Subsistema de Seguridad

CU Autenticar Usuario



CU Gestionar Usuario



4.2.2.3 Subsistemas del Análisis

Los Subsistemas del Análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. Estos pueden constar de clases del análisis, realizaciones, y de otros paquetes del análisis (recursivamente). Además es necesario que sean altamente cohesivos y débilmente acoplados, es decir que las dependencias entre ellos deben minimizarse.

En el caso particular del sistema en cuestión, se han identificado seis paquetes del

análisis, donde agrupamos a los subsistemas de gestión de reportes en el Subsistemas de Gestión de Reportes.

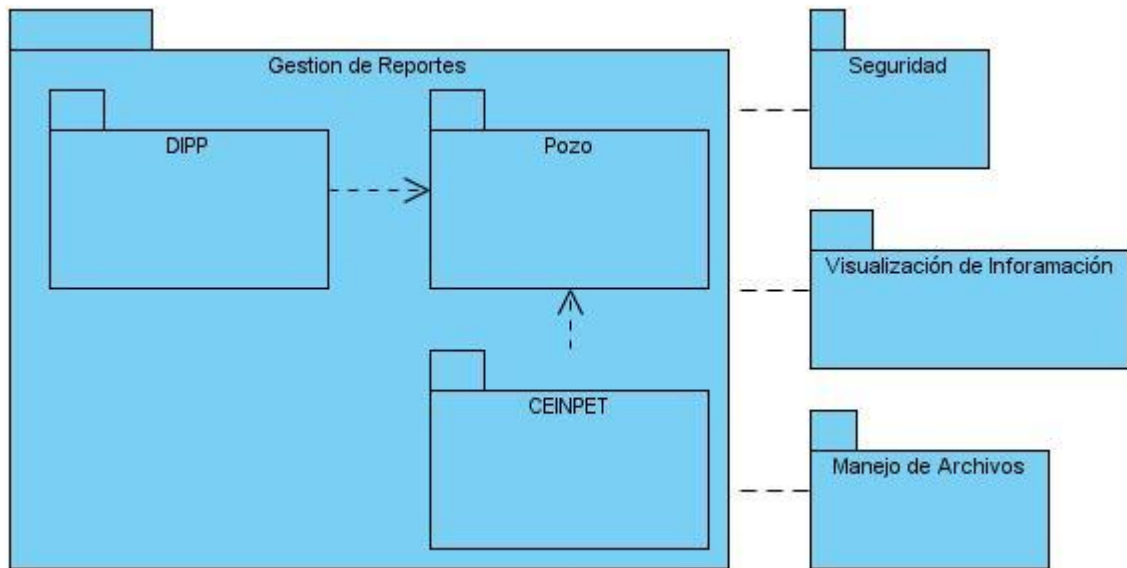


Diagrama de Paquetes del Análisis

Como se aprecia en la figura, los subsistemas derivados del flujo de trabajo de requerimientos se mantienen, solo que ahora agrupamos los artefactos del análisis. En el Subsistema pozo se agrupan todas las clases y relaciones de los casos de uso del subsistema heredado del flujo de requerimientos, lo mismo sucede para los restantes subsistemas DIPP y CEINPET. En el caso de los subsistemas Seguridad, Visualización de Información y Manejo de Archivos, de manera análoga al subsistema pozo, lo que se relacionan con el nuevo subsistema Gestión de Reportes; el cual agrupa todos los casos de uso encargados de la gestión de reportes; estos tres subsistemas son horizontales al negocio enfrentado por lo cual no poseen dependencia alguna de los demás subsistemas.

4.3 Diseño

4.3.1 Modelo de Diseño

El Modelo de Diseño es un modelo que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, unidas a ciertas restricciones del dominio del negocio y por consiguiente de la aplicación. Según RUP el modelo de diseño se comienza a construir al final de la fase de Elaboración y se culmina al comienzo de la fase de Construcción, lo cual contribuye a lograr una arquitectura estable, lo cual posibilita crear un plano del Modelo de Implementación.

Es importante aclarar que para comenzar las tareas del diseño es necesario tener definida la Línea Base de la Arquitectura, no así la el Modelo de Análisis, aunque esto puede variar según el modelo de desarrollo con que se trabaje. Además si va a trabajar usando Base de Datos, la modelación de la misma debe estar en una versión la cual no debe tener cambios, cuando iniciamos el diseño.


En el caso específico del sistema a modelar, se tomo el modelo de análisis como una herramienta intermedia (Ver Epígrafe 4.2), por lo cual se comenzó a modelar el diseño luego de tener el análisis culminado, esto puede demorar el proceso de desarrollo en la fase de elaboración pero posibilita que cuando se comience la fase de Construcción no tengamos que regresar a refinar el Modelo de Análisis y nos concentramos en los flujos de diseño e implementación.



Como se menciona en el epígrafe 2.2 para el modelado ágil de la aplicación se tomaron varias buenas prácticas de RUP para lograr este objetivo, una de ellas el programar y luego documentar. Esto se logra debido a que el Modelo de Implementación está estrechamente ligado al Modelo de Diseño, por lo cual la característica incremental de RUP te permite ir y venir entre estos modelos donde el diseño se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráficamente, y de manera análoga la implementación permite refinar el diseño a medida que avanza el desarrollo del sistema.

En el caso específico del sistema en cuestión, se realizaron los artefactos del diseño, al mismo tiempo que se implementaron las funcionalidades críticas del sistema; llámese por artefactos generados: Clases del Diseño, Realización de los Casos de Uso(Diagramas de Clases), Subsistemas de Diseño y el Modelo de Diseño, el cual agrupa todos los artefactos generados.

4.3.2 Extensiones UML para Diseño Web

Como se expuso en el capítulo dos de este trabajo, el lenguaje de modelado utilizado es UML el cual define extensiones para el modelado de aplicaciones web

De:	A: Client Page	Server Page	Form
	<i>Link</i> <i>Redirect</i>	<i>Link</i>	Aggregation

	Builds Redirect	Redirect	
	Aggregated by	Submit	

4.3.3 Artefactos del Diseño

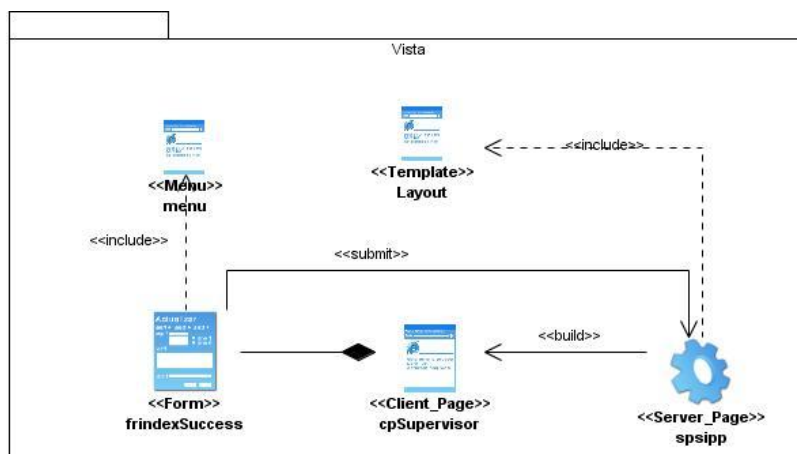
Como se expone en el epígrafe 2.2, donde se abordan varias metodologías de desarrollo, RUP es una metodología centrada en la arquitectura por lo cual al comenzar el diseño se debe estudiar con detenimiento la arquitectura propuesta y partiendo de ella y las CU, se comienzan a lograr los artefactos de los casos de uso críticos.

Para lograr un entendimiento de lo que se expondrá a continuación debe antes leer el documento anexo: Documento de Arquitectura SIPP

En la arquitectura propuesta, se escoge el patrón arquitectónico Modelo-Vista-Controladora (MVC, Ver epígrafe 1.6), plataforma web, el Framework Symfony, PHP(Lenguaje de programación) y Postgree SQL(Gestor de Base de Datos).

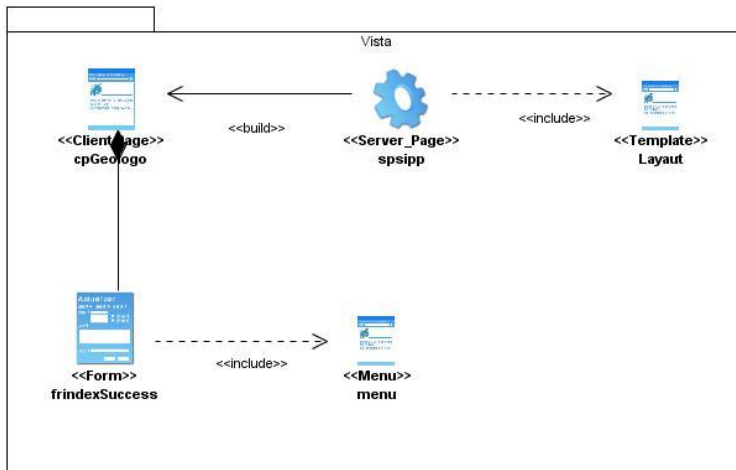
Teniendo esto se comienza el diseño del sistema: Primeramente el hecho de usar u framework de desarrollo restringe el diseño a la estructura que del framework, debido a esto se definen siete módulos(Subsistemas); de cada unos de los módulos de identificaron sus respectivas Vistas y Controladoras:

- Supervisor



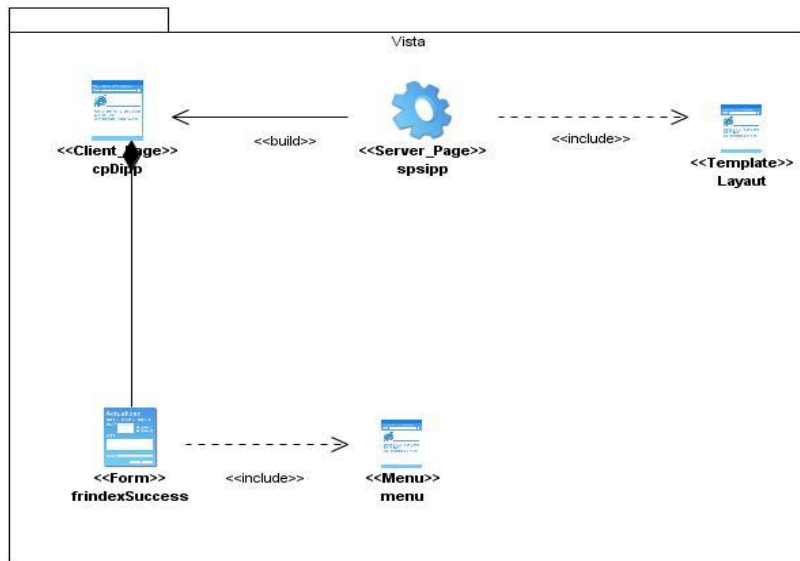
supervisorActions
+executeIndex()
+executeAdicionarReporteDiarioPerforacion()
+executeAdicionarReporteDiarioGeologia()
+executeAdicionarReporteLodo()
+executeAdicionarReporteDirreccional()
+executeAdicionarReporteComposiciondeHerramientas()
+executeAdicionarMetrageDiario()
+executeAdicionarCronogramaPlanificado()
+executeAdicionarReportePropiedadesFluidoPerforacion()
+executeAdicionarReportePresupuestoDiario()
+executeAdicionarReporteCostoProductosQuimicos()
+executeActualizarReporteDiarioPerforacion()
+executeActualizarReporteDireccional()
+executeActualizarReporteDiarioGeologia()
+executeActualizarReporteLodo()
+executeActualizarReporteComposiciondeHerramientas()
+executeActualizarMetrageDiario()
+executeActualizarCronogramaPlanificado()
+executeActualizarReportePropiedadesFluidoPerforacion()
+executeActualizarReportePresupuestoDiario()
+executeActualizarReporteCostoProductosQuimicos()
+executeGenerarReporteDiarioOperativo()
+executeGenerarDistribucionTiempoDiario()
+executeGenerarReporteRecordBarrena()
+executeGenerarReporteDistribucionTiempoEtapas()
+executeGenerarReporteCostosSemanal()
+executeGenerarResumenCosto()

- Geólogo



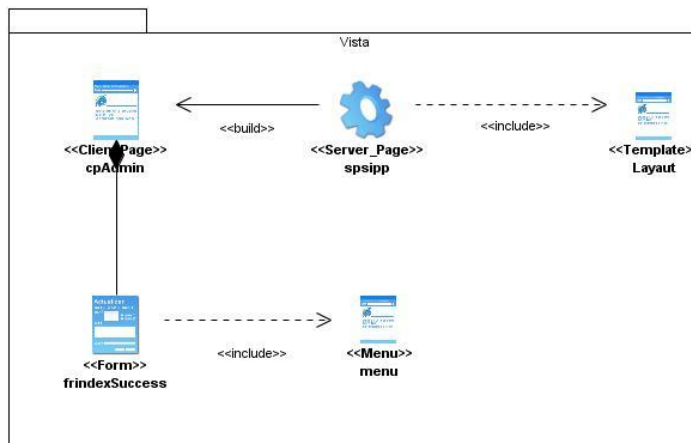
geologoActions
+executeIndex()
+executeAdicionarReporteDiarioGeologia()
+executeActualizarReporteDiarioLodo()

- DIPP



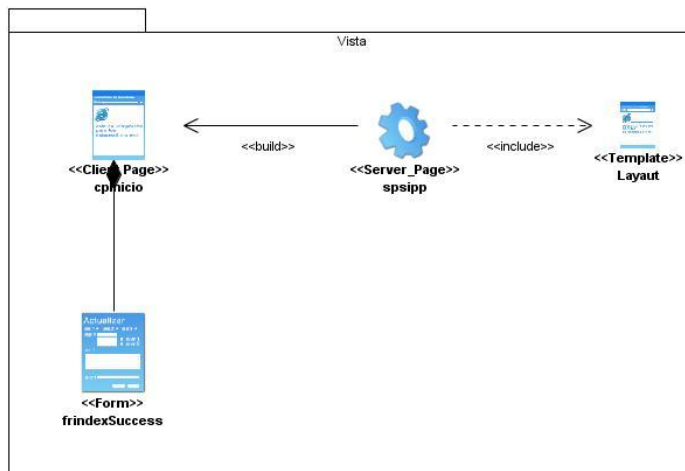
dippActions
+executeIndex()
+executeAdicionarParteOperativodePerforacion()
+executeAdicionarPartedeIntervencion()
+executeActualizarParteOperativodePerforacion()
+executeActualizarPartedeIntervencion()
+executeGenerarPartedePerforacion()

- Administración



adminActions
+executeIndex()
+executeAdicionarPozo()
+executeActualizarPozo()
+executeCerrarPozo()
+executeListadoPozos()
+executeAdicionarUsuario()
+executeActualizarUsuario()
+executeEliminarUsuario()
+executeListadoUsuarios()
+executeActualizarPermisos()
+executeAdicionarPermisos()

- Inicio



inicioActions
+executeLogin()
+executeLogout()

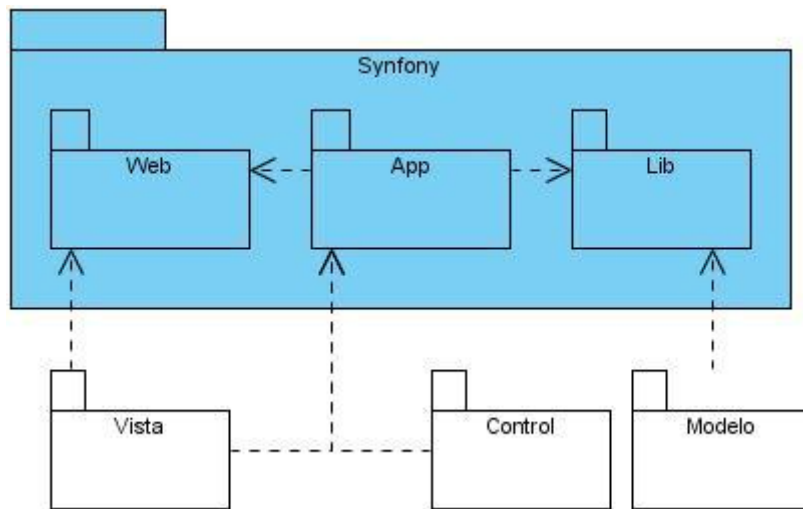
Para que pueda existir un mejor entendimiento de le presentamos como está el Framework Symfony, el cual también aplica el patrón MVC para su funcionamiento.

[Ver Anexo 24](#)

Como se aprecia el framework divide el proyecto creado en tres carpetas principales: App, Web y Lib. En App almacena todos los módulos creados, con sus success(vistas) y sus actions(controladoras), en Lib todo lo referente a la capa del modelo, la cual se agrupa específicamente en la carpeta Model, donde en la carpeta om, de almacena la abstracción de los datos y en map la abstracción generada para el mapeo de la base de datos. Por último en la carpeta Web se almacenar varios componentes referentes a la Vista, las imágenes a utilizar, los CCS y los archivos JavaScript, entre otros.

Importante destacar el tratamiento que recibe la Modelo, de parte del framework de desarrollo, el cual te crea una capa de abstracción de los datos, usando la aplicación Propel, por lo cual no es necesario modelar la Modelo y nos concentramos en modelar las Vistas y las Controladoras.

Por lo cual la estructura del diseño siguiendo el patrón MVC, aplicado en el framework de desarrollo queda de la siguiente manera:

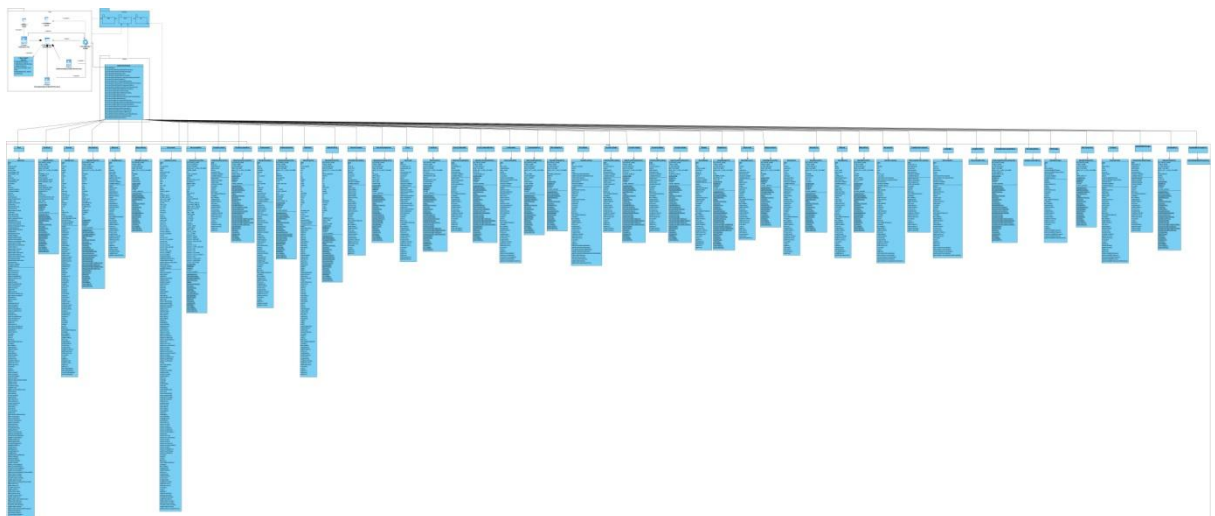


Luego de haber identificado todas las clases de diseño y la estructura a seguir, se presentan los diagramas de clases del diseño de los casos de uso críticos, así como la identificación y agrupación por subsistemas.

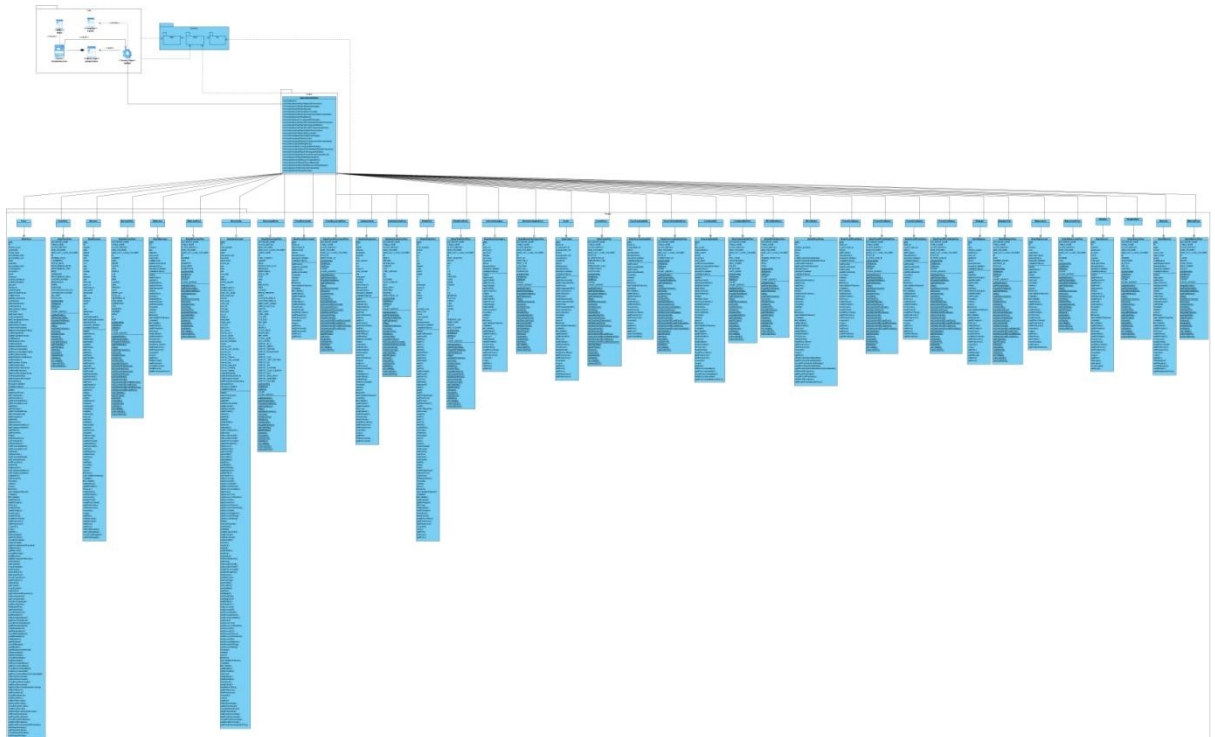
4.3.3.1 Diagramas de Clases

Subsistema Supervisor

Gestionar Reporte Diario de Perforación

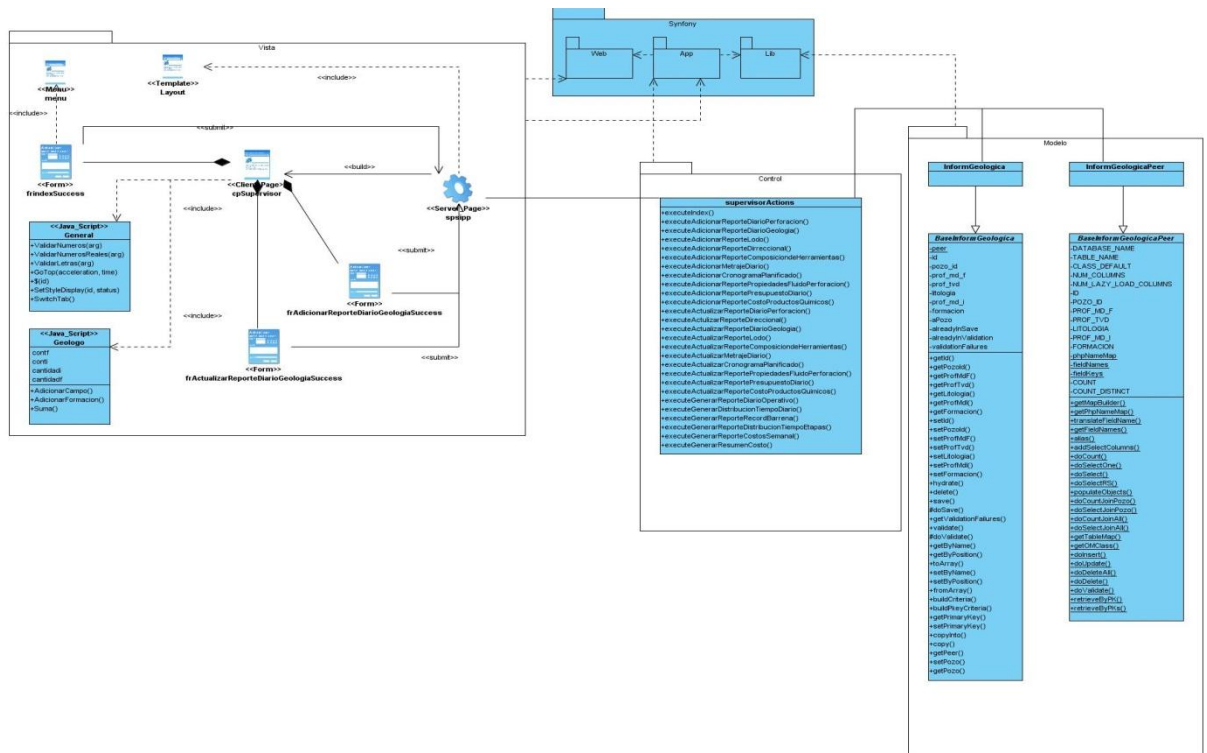


Generar Reporte Diario Operativo

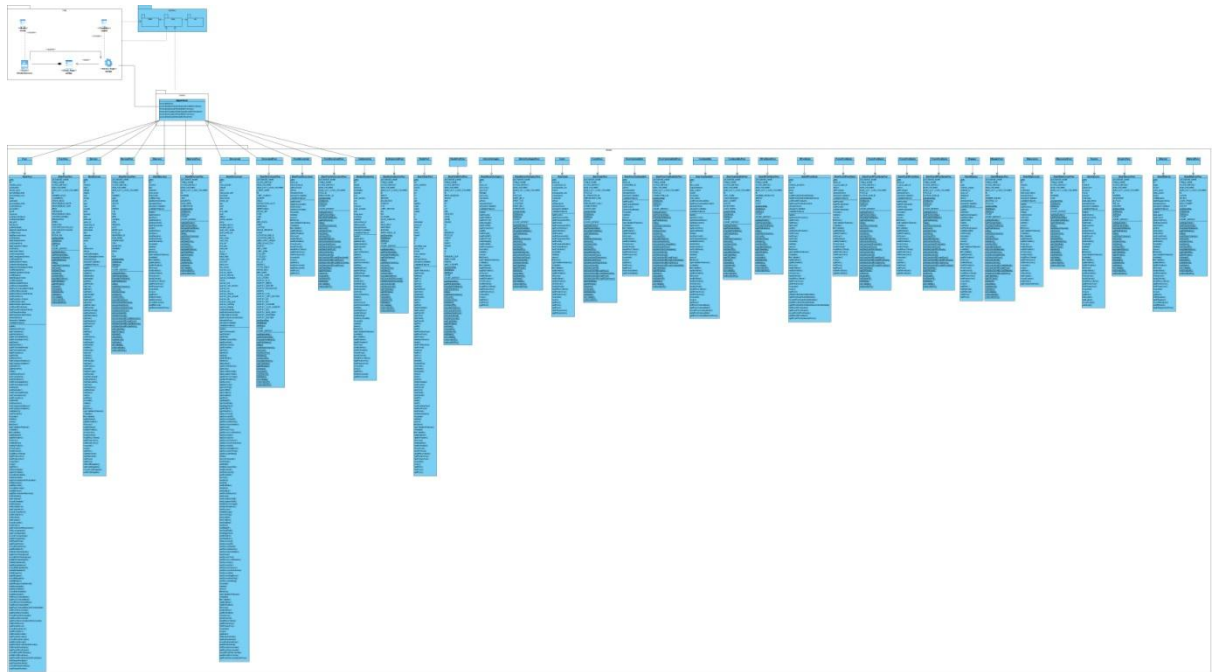


Subsistema Geólogo

Gestionar Reporte Diario de Geología

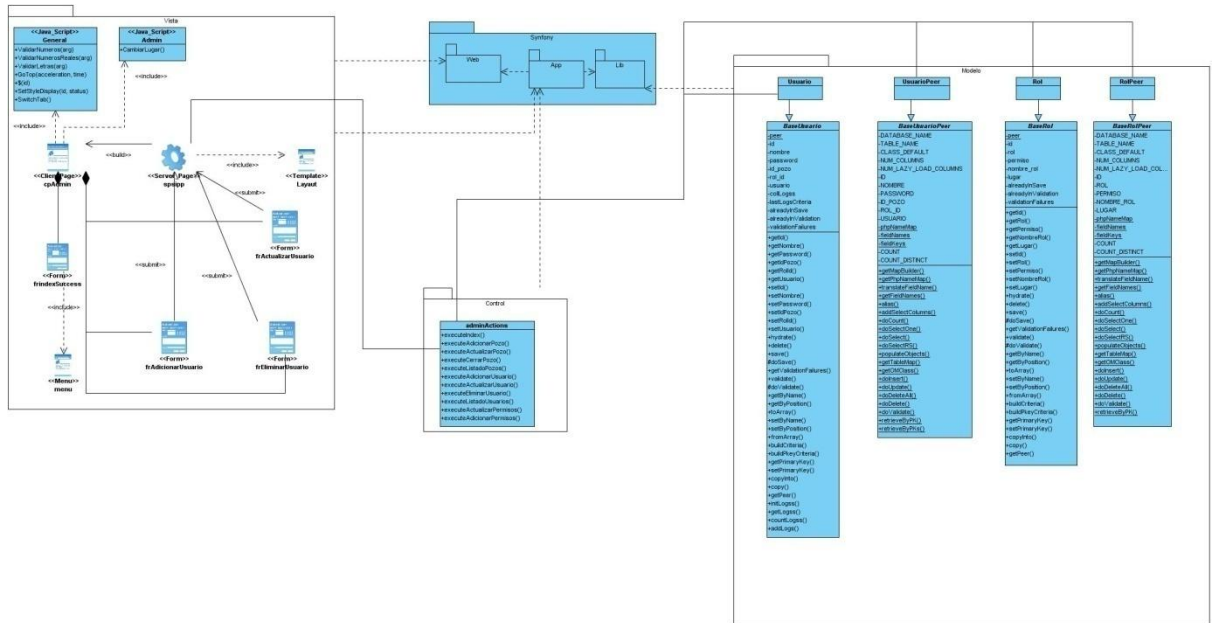


Generar Parte Perforación



Subsistema Administración

Gestionar Usuario



Los subsistemas se agruparon siguiendo la estructura que se definió dentro del framework, y luego se agruparon dentro de subsistemas ya conocidos, de otros flujos de trabajo, garantizando a través de subsistemas la trazabilidad del sistema. También es importante resaltar los cambios que a pesar que subsistemas del diseño posean un nombre similar a otros en flujos anteriores no significa que posean las mismas funcionalidad, ya que la agrupación de se realizó agrupando por actores. Debido a ello la trazabilidad del Subsistema DIPP, no es completa ya que el CU Gestionar Información General de Pozo, pasa al subsistema Administración del subsistema de Seguridad.

También se realizó una Vista Arquitectónica por Subsistemas. [Ver Anexo 25](#)

4.4 Patrones de Diseño

Como se aprecia en el epígrafe 1.6 dentro de los mucho patrones que existen, entre ellos están los patrones de diseño y en este epígrafe expondremos cuales fueron usados para el modelado de esta aplicación.

Patrones GRASP

- Experto

El patrón de diseño **Experto** está diseñado en que la responsabilidad de realizar una labor es 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 estemos considerando los mismos aspectos del sistema:

- Lógica de negocio
- Persistencia a la base de datos
- Interfaz de usuario

- Controlador

El patrón de diseño **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.

- Alta Cohesión

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase, además se caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Por tanto la aplicación de este patrón posibilita, que las clases sean más fáciles de comprender, de reutilizar y conservar

- 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; sean más fáciles de entender cuando estén aisladas, por lo que son más fáciles de reutilizar.

Importante destacar que estos cuatro patrones no se pueden ver por separado, si no como un todo que posibilita la asignación de responsabilidades.

Otros

- Front-Controller

Para aplicaciones Web se recomienda utilizar este patrón que obliga a que todas las peticiones hechas a nuestra 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.

- Decorator

Este patrón describe cómo agregar funciones a los objetos dinámicamente. Decorator es un patrón estructural que compone los objetos recursivamente para permitir una participación abierta a una serie de responsabilidades adicionales. Estas responsabilidades se añaden a los objetos en vez de añadir a las clases.

4.5 Conclusiones Parciales

En este capítulo tratamos el resultado de este trabajo, el análisis y diseño del sistema propuesto, donde se abordaron aspectos teóricos de los Modelos de Análisis y Diseño, los artefactos generados, así como los patrones de diseño empleados.

Conclusiones

Al término de este trabajo se puede arribar a las siguientes conclusiones:

- Con el estudio de los procesos de negocio en los pozos de petróleo en perforación, se obtuvo el conocimiento necesario para identificar los procesos de negocio.
- El estudio del arte y de la selección correcta de la metodología de software, permitió la modelación correcta del negocio y la captura precisa de los requisitos teniendo en cuenta las necesidades del cliente.
- La modelación efectiva del negocio permitió la construcción del de manera tal que respondiera a las características específicas del mismo.
- La correcta utilización de la herramienta CASE permitió la correcta representación del sistema a través de modelos sustentados por UML.
- La aplicación de buenas prácticas de RUP en la Fase de Elaboración y Construcción permitió al equipo de trabajo lograr implementar las funcionalidades críticas del sistema.
- El diseño de una arquitectura robusta (documento referenciado en este trabajo), permitió el cumplimiento de los requerimientos no funcionales, así como la modelación de un diseño que respondiera a las necesidades del sistema.
- Se logra un diseño que cumpla las restricciones del Framework de desarrollo, aplicando patrones de diseño para la óptima construcción del mismo.
- Se identifican de manera temprana subsistemas, que pueden convertirse en componentes reutilizables para aplicaciones de similares características.

Recomendaciones

Con el objetivo de realizar mejoras a la aplicación propuesta, se recomienda:

- Medir la calidad el Diseño realizado.
- Continuar la investigación de los procesos de negocio en torno a la perforación en la DIPP y en los Pozos en perforación.
- Investigar sobre otras de las extensiones UML para la modelación de próximas versiones
- Identificar nuevas funcionalidades para la construcción de componentes que logren redondear el sistema como un producto para el mercado:
 - Generador de Reportes
 - Internacionalización
 - Configuración
 - Historial de Usuario
 - Comunicación

Referencias Bibliográficas

1. **Free Download Manager.** Free Download Manager. [En línea] [Citado el: 20 de 8 de 2008.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/.
2. **WellSight.** Wellsight.com. [En línea] 3 de 3 de 2009. [Citado el: 6 de 11 de 2008.] <http://www.wellsight.com>.
3. **AddLink.** AddLink. [En línea] [Citado el: 22 de 8 de 2008.] <http://www.addlink.es/productos.asp?pid=430>.
4. **Peloton.** Drilling & Well Data Software Solutions. [En línea] [Citado el: 3 de 3 de 2009.] <http://www.peloton.com/es/>.
5. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* La Habana : Felix Varela, 2005.
6. **Jacobson, Ivar, Booch, Grady y Rumbauch, James.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004.
7. **Universidad de los Andes.** UNIANDES. [En línea] [Citado el: 23 de 1 de 2009.] http://agamenon.uniandes.edu.co/~pfiguero/soo/Magister_Patrones/intropatrones.html.
8. **Universidad Autonama Nacional de Mexico.** UNAM. [En línea] [Citado el: 23 de 1 de 2009.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
9. UML y Patrones. [aut. libro] Craig Larman. *UML y Patrones. Introducción a al análisis y diseño orientado a objetos.* La Habana : Felix Varela, 2004.
10. [En línea] [Citado el: 22 de 1 de 2009.] <http://migueljaque.com/index.php/patrones/43-patronesintroduccion/88-tipospatrones>.
11. **De Frutos, Evaristo y Guerra, Ignacio.** eduforge. [En línea] 12 de 12 de 2007. [Citado el: 4 de 3 de 2009.] https://eduforge.org/docman/view.php/230/3180/SUMA_Metodologia_v0.1.pdf.
12. **Martinez, Enrique A.** crisol. [En línea] Julio de 2005. [Citado el: 4 de 3 de 2009.] <http://www.crisol.cc/eamf-tec/metodologias/Metodologias.doc>.
13. **Figuroa Medina, Yenny.** [En línea] 2006. [Citado el: 5 de 3 de 2009.] http://is.umb.edu.co/aulaumb/file.php/33/METODOLOGIAS_DE_DESARROLLO_DE_SOFTWARE_V01.ppt.
14. **Auladirectiva.** Aula Directiva Online. [En línea] 2006. [Citado el: 5 de 3 de 2009.] <http://www.auladirectiva.com/curso/metrica-v3/presentacion.html>.
15. **UNAP.** [En línea] 22 de 03 de 2004. [Citado el: 5 de 3 de 2009.] <http://www.unap.cl/~setcheve/Metrica/m/index.html>.

16. **US.** us. [En línea] [Citado el: 4 de 3 de 2009.]
<http://www1.us.es/pautadatos/publico/asignaturas/39813/17161/04MetricaV3.pdf>.
17. **madritel.** [En línea] 15 de Julio de 2003. [Citado el: 5 de 3 de 2009.]
<http://web.madritel.es/personales3/edcollado/ingsw/tema2/tema2.htm>.
18. **Universidad de Glamorgan.** glam. [En línea] [Citado el: 5 de 3 de 2009.]
<http://www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4/CHAPTER4.PPT>.
19. **Canós, Carlos H., Letelier, Patricio y Penadés, Maria Carmen.** willydev. [En línea] [Citado el: 20 de 2 de 2009.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
20. **baufest.** baufest. [En línea] [Citado el: 4 de 3 de 2009.]
http://www.baufest.com/spanish/scrum/scrumconference2006/Que_es_scrum.pdf.
21. **proyectosagiles.** [En línea] [Citado el: 4 de 3 de 2009.] <http://www.proyectosagiles.org/como-funciona-scrum>.
22. **proyectosagiles.** [En línea] [Citado el: 4 de 3 de 2009.]
<http://www.proyectosagiles.org/beneficios-de-scrum>.
23. **Pérez Vásquez, Luis Miguel.** emagister. [En línea] 21 de Febrero de 2008. [Citado el: 4 de 3 de 2009.] <http://grupos.emagister.com/ficheros/vcruzada?idGrupo=1055&idFichero=16300>.
24. **Morató Moscardó, Juan.** UPV. [En línea] 4 de 2 de 2003. [Citado el: 5 de 3 de 2009.]
<http://users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/292002.ppt>.
25. **crystalmethodologies.** [En línea] [Citado el: 5 de 3 de 2009.]
<http://crystalmethodologies.blogspot.com/>.
26. **Fernández Enrich, Margarita.** [En línea] 4 de 2 de 2003. [Citado el: 4 de 3 de 2009.]
<http://www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/282002.ppt>.
27. **utopia.** utopia. [En línea] [Citado el: 5 de 3 de 2009.]
http://utopia.csis.pace.edu/dps/2007/amannette-wright/dps/DCS801_821/Crystal_Methodologies_summary.doc.
28. **Palacio, Juan.** navegapolis. [En línea] 2006. [Citado el: 7 de 3 de 2009.]
http://www.navegapolis.net/files/s/NST-003_01.pdf.
29. **Microsoft.** [En línea] [Citado el: 8 de 3 de 2009.]
<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/planificacion/msf.mspix>.
30. **Scalzone, Lic. Patricia.** [En línea] [Citado el: 8 de 3 de 2009.]
<http://download.microsoft.com/download/4/4/E/44E1B331-E509-4D10-A9E3-B60640A3A403/20051206-ARC-BA.ppt>.
31. **Gattaca S.A.** e-gattaca. [En línea] [Citado el: 4 de 3 de 2009.] <http://www.e-gattaca.com/eContent/library/documents/DocNewsNo50DocumentNo6.PDF>.

32. **GPI Consultores.** gpicr. [En línea] [Citado el: 7 de 3 de 2009.] <http://www.gpicr.com/msf.aspx>.
33. **OMG.** www.uml.org. [En línea] 8 de 1 de 2009. [Citado el: 30 de 3 de 2009.] <http://www.uml.org/>.
34. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *UML. Manual de Referencia.* s.l. : Addison Wesley.
35. **Marquez Marcos, Sergio.** [En línea] 4 de 12 de 2008. [Citado el: 29 de 3 de 2009.] http://petra.euitio.uniovi.es/~i1942861/HD/Trabajos/Informe_Herramientas_Disenio_y_Analisis_OO_v1_0.pdf.
36. **Visible Enterprise.** [En línea] 2008. [Citado el: 29 de 3 de 2009.] <http://www.visible.com/Products/Analyst/index.htm>.
37. **sparxsystems.** [En línea] 2009. [Citado el: 29 de 3 de 2009.] <http://www.sparxsystems.com.ar/products/index.html>.
38. **Visual Paradigm.** [En línea] 2009. [Citado el: 29 de 3 de 2009.] <http://www.visual-paradigm.com>.
39. *teleformación.uci.cu.* [En línea] 2007. [Citado el: 29 de 1 de 2009.] http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_5/Conferencia_Requerimientos_2007-2008.pdf.
40. **Gunnar, Övergaard y Karin, Palmkvist.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004.
41. **aidanamx.** [En línea] 28 de 5 de 2008. [Citado el: 7 de 3 de 2009.] <http://aidanamx.blogspot.com/>.
42. **Díaz, Néstor L.** *Ventajas de PostgreSQL.* [En línea] 31 de 5 de 2003. [Citado el: 10 de 2 de 2009.] http://soperte.tiendalinux.com/porta/Portfolio/postgresql_ventajas_html.
43. **Grupo de Soluciones Innova.** [En línea] GSI, 2008. [Citado el: 30 de 3 de 2009.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
44. **Larman, Craig.** *UML y Patrones.* La Habana : Feliz Varela, 2004.
45. **Agile Spain.** Agile Spain. [En línea] 2009. [Citado el: 14 de 2 de 2009.] <http://www.agile-spain.com/tema/metodologias>.
46. **Ambyssoft.** Ambyssoft. [En línea] 2009. [Citado el: 25 de 4 de 2009.] <http://www.ambyssoft.com/>.
47. **agilemodeling.** [En línea] 2009. [Citado el: 10 de 3 de 2009.] <http://www.agilemodeling.com/>.
48. **enterpriseunifiedprocess.** enterpriseunifiedprocess. [En línea] 2009. [Citado el:] <http://www.enterpriseunifiedprocess.com/>.
49. **kybeleconsulting.** [En línea] [Citado el: 10 de 6 de 2009.] <http://kybeleconsulting.blogspot.com/2009/06/puntos-caso-de-uso.html>.

50. codeproject. [En línea] 2009. [Citado el: 10 de 6 de 2009.]
<http://www.codeproject.com/KB/architecture/usecasepoints.aspx>.

51. **evigo**. [En línea] [Citado el: 8 de 6 de 2009.]
http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/cocomo2k.pdf.

52. **Potencier, Fabien y Zaninotto, Francis**. *Symfony Guía Definitiva* . 2008.

Bibliografía

1. **Free Download Manager.** Free Download Manager. [En línea] [Citado el: 20 de 8 de 2008.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p/.
2. **WellSight.** Wellsight.com. [En línea] 3 de 3 de 2009. [Citado el: 6 de 11 de 2008.] <http://www.wellsight.com>.
3. **AddLink.** AddLink. [En línea] [Citado el: 22 de 8 de 2008.] <http://www.addlink.es/productos.asp?pid=430>.
4. **Peloton.** Drilling & Well Data Software Solutions. [En línea] [Citado el: 3 de 3 de 2009.] <http://www.peloton.com/es/>.
5. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico.* La Habana : Felix Varela, 2005.
6. **Jacobson, Ivar, Booch, Grady y Rumbauch, James.** *El Proceso Unificado de Desarrollo de Software.* La Habana : Félix Varela, 2004.
7. **Universidad de los Andes.** UNIANDES. [En línea] [Citado el: 23 de 1 de 2009.] http://agamenon.uniandes.edu.co/~pfiguero/soo/Magister_Patrones/intropatrones.html.
8. **Universidad Autonama Nacional de Mexico.** UNAM. [En línea] [Citado el: 23 de 1 de 2009.] <http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html>.
9. UML y Patrones. [aut. libro] Craig Larman. *UML y Patrones. Introducción a al análisis y diseño orientado a objetos.* La Habana : Felix Varela, 2004.
10. [En línea] [Citado el: 22 de 1 de 2009.] <http://migueljaque.com/index.php/patrones/43-patronesintroduccion/88-tipospatrones>.
11. **De Frutos, Evaristo y Guerra, Ignacio.** eduforge. [En línea] 12 de 12 de 2007. [Citado el: 4 de 3 de 2009.] https://eduforge.org/docman/view.php/230/3180/SUMA_Metodologia_v0.1.pdf.
12. **Martinez, Enrique A.** crisol. [En línea] Julio de 2005. [Citado el: 4 de 3 de 2009.] <http://www.crisol.cc/eamf-tec/metodologias/Metodologias.doc>.
13. **Figueroa Medina, Yenny.** [En línea] 2006. [Citado el: 5 de 3 de 2009.] http://is.umb.edu.co/aulaumb/file.php/33/METODOLOGIAS_DE_DESARROLLO_DE_SOFTWARE_V01.ppt.
14. **Auladirectiva.** Aula Directiva Online. [En línea] 2006. [Citado el: 5 de 3 de 2009.] <http://www.auladirectiva.com/curso/metrica-v3/presentacion.html>.
15. **UNAP.** [En línea] 22 de 03 de 2004. [Citado el: 5 de 3 de 2009.] <http://www.unap.cl/~setcheve/Metrica/m/index.html>.

16. **US. us.** [En línea] [Citado el: 4 de 3 de 2009.]
<http://www1.us.es/pautadatos/publico/asignaturas/39813/17161/04MetricaV3.pdf>.
17. **madritel.** [En línea] 15 de Julio de 2003. [Citado el: 5 de 3 de 2009.]
<http://web.madritel.es/personales3/edcollado/ingsw/tema2/tema2.htm>.
18. **Universidad de Glamorgan.** glam. [En línea] [Citado el: 5 de 3 de 2009.]
<http://www.comp.glam.ac.uk/pages/staff/tdhutchings/chapter4/CHAPTER4.PPT>.
19. **Canós, Carlos H., Letelier, Patricio y Penadés, Maria Carmen.** willydev. [En línea] [Citado el: 20 de 2 de 2009.] <http://www.willydev.net/descargas/prev/TodoAgil.pdf>.
20. **baufest.** baufest. [En línea] [Citado el: 4 de 3 de 2009.]
http://www.baufest.com/spanish/scrum/scrumconference2006/Que_es_scrum.pdf.
21. **proyectosagiles.** [En línea] [Citado el: 4 de 3 de 2009.] <http://www.proyectosagiles.org/como-funciona-scrum>.
22. **proyectosagiles.** [En línea] [Citado el: 4 de 3 de 2009.]
<http://www.proyectosagiles.org/beneficios-de-scrum>.
23. **Pérez Vásquez, Luis Miguel.** emagister. [En línea] 21 de Febrero de 2008. [Citado el: 4 de 3 de 2009.] <http://grupos.emagister.com/ficheros/vcruzada?idGrupo=1055&idFichero=16300>.
24. **Morató Moscardó, Juan.** UPV. [En línea] 4 de 2 de 2003. [Citado el: 5 de 3 de 2009.]
<http://users.dsic.upv.es/asignaturas/facultad/lsi/trabajos/292002.ppt>.
25. **crystalmethodologies.** [En línea] [Citado el: 5 de 3 de 2009.]
<http://crystalmethodologies.blogspot.com/>.
26. **Fernández Enrich, Margarita.** [En línea] 4 de 2 de 2003. [Citado el: 4 de 3 de 2009.]
<http://www.dsic.upv.es/asignaturas/facultad/lsi/trabajos/282002.ppt>.
27. **utopia.** utopia. [En línea] [Citado el: 5 de 3 de 2009.]
http://utopia.csis.pace.edu/dps/2007/amannette-wright/dps/DCS801_821/Crystal_Methodologies_summary.doc.
28. **Palacio, Juan.** navegapolis. [En línea] 2006. [Citado el: 7 de 3 de 2009.]
http://www.navegapolis.net/files/s/NST-003_01.pdf.
29. **Microsoft.** [En línea] [Citado el: 8 de 3 de 2009.]
<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/planificacion/msf.mspx>.
30. **Scalzone, Lic. Patricia.** [En línea] [Citado el: 8 de 3 de 2009.]
<http://download.microsoft.com/download/4/4/E/44E1B331-E509-4D10-A9E3-B60640A3A403/20051206-ARC-BA.ppt>.
31. **Gattaca S.A.** e-gattaca. [En línea] [Citado el: 4 de 3 de 2009.] <http://www.e-gattaca.com/eContent/library/documents/DocNewsNo50DocumentNo6.PDF>.

32. **GPI Consultores.** gpicr. [En línea] [Citado el: 7 de 3 de 2009.] <http://www.gpicr.com/msf.aspx>.
33. **OMG.** www.uml.org. [En línea] 8 de 1 de 2009. [Citado el: 30 de 3 de 2009.] <http://www.uml.org/>.
34. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *UML. Manual de Referencia.* s.l. : Addison Wesley.
35. **Marquez Marcos, Sergio.** [En línea] 4 de 12 de 2008. [Citado el: 29 de 3 de 2009.] http://petra.euitio.uniovi.es/~i1942861/HD/Trabajos/Informe_Herramientas_Disenio_y_Analisis_OO_v1_0.pdf.
36. **Visible Enterprise.** [En línea] 2008. [Citado el: 29 de 3 de 2009.] <http://www.visible.com/Products/Analyst/index.htm>.
37. **sparxsystems.** [En línea] 2009. [Citado el: 29 de 3 de 2009.] <http://www.sparxsystems.com.ar/products/index.html>.
38. **Visual Paradigm.** [En línea] 2009. [Citado el: 29 de 3 de 2009.] <http://www.visual-paradigm.com>.
39. **teleformación.uci.cu.** [En línea] 2007. [Citado el: 29 de 1 de 2009.] http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_5/Conferencia_Requerimientos_2007-2008.pdf.
40. **Gunnar, Övergaard y Karin, Palmkvist.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004.
41. **aidanamx.** [En línea] 28 de 5 de 2008. [Citado el: 7 de 3 de 2009.] <http://aidanamx.blogspot.com/>.
42. **Díaz, Néstor L.** *Ventajas de PostgreSQL.* [En línea] 31 de 5 de 2003. [Citado el: 10 de 2 de 2009.] http://soperte.tiendalinux.com/porta/Portfolio/postgresql_ventajas_html.
43. **Grupo de Soluciones Innova.** [En línea] GSI, 2008. [Citado el: 30 de 3 de 2009.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.
44. **Larman, Craig.** *UML y Patrones.* La Habana : Feliz Varela, 2004.
45. **Agile Spain.** Agile Spain. [En línea] 2009. [Citado el: 14 de 2 de 2009.] <http://www.agile-spain.com/tema/metodologias>.
46. **Ambyssoft.** Ambyssoft. [En línea] 2009. [Citado el: 25 de 4 de 2009.] <http://www.ambyssoft.com/>.
47. **agilemodeling.** [En línea] 2009. [Citado el: 10 de 3 de 2009.] <http://www.agilemodeling.com/>.
48. **enterpriseunifiedprocess.** enterpriseunifiedprocess. [En línea] 2009. [Citado el:] <http://www.enterpriseunifiedprocess.com/>.
49. **kybeleconsulting.** [En línea] [Citado el: 10 de 6 de 2009.] <http://kybeleconsulting.blogspot.com/2009/06/puntos-caso-de-uso.html>.

50. codeproject. [En línea] 2009. [Citado el: 10 de 6 de 2009.]
<http://www.codeproject.com/KB/architecture/usecasepoints.aspx>.

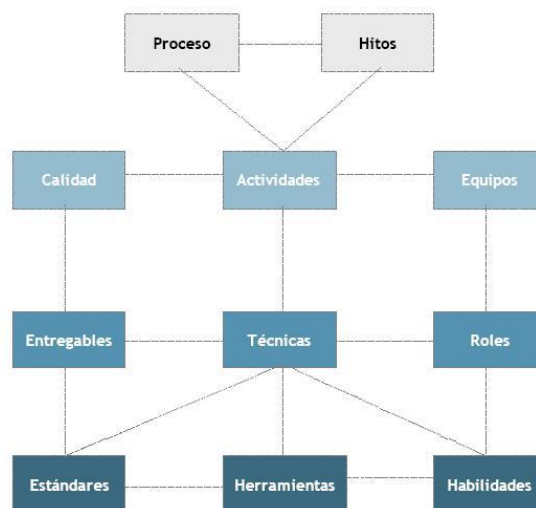
51. **evigo**. [En línea] [Citado el: 8 de 6 de 2009.]
http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/cocomo2k.pdf.

52. **Potencier, Fabien y Zaninotto, Francis**. *Symfony Guía Definitiva* . 2008.

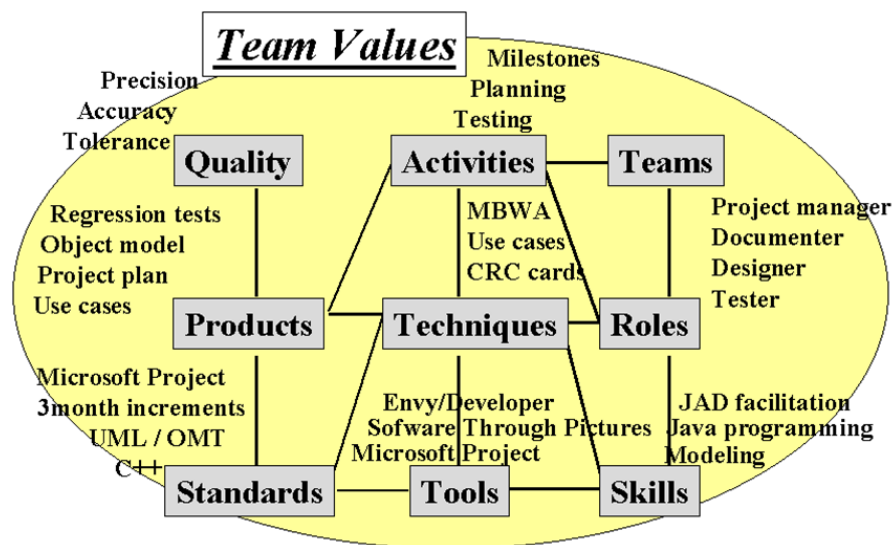
Anexo 1. Well Logger

From	To	Description	Classification	Line Type
0	2.5	SILTY SAND, very fine, medium brown, well graded, slightly moist.	SM	1 - Dash
2.5	9	SANDY GRAVEL, very fine sand to fine pebble gravel, dark gray to brown, well graded, wet, FAINT HYDROCARBON ODOR.	GW	0 - Solid
9	14	SILTY GRAVEL, fine pebble gravel and silt with some sand, dark gray to brown, poorly graded.	GM	0 - Solid
14	16	SAND, very fine-grained, grading to silt, mottled: greenish gray to brown, subangular, poorly graded, note color change, FAIR HYDROCARBON ODOR.	SM	0 - Solid
16	19	SILT, light brown, sandy, low plasticity, argillaceous.	ML	1 - Dash
19	22	SILT, gray, clayey, some coarse sand, slightly plastic, stiff, moist, FAINT HYDROCARBON ODOR.	SW	1 - Dash
22	28	SAND, brown, fine-grained, well graded, wet, FAINT HYDROCARBON ODOR.	SW	1 - Dash
28	33	CLAY, dark gray to black, micaceous, hard, damp, NO ODOR.	CL	1 - Dash
33	36.5	CLAY, dark gray, micaceous, hard, damp, NO ODOR.	CL	0 - Solid

Anexo 2. Definición de Metodología de Software según Alistair Cockburn



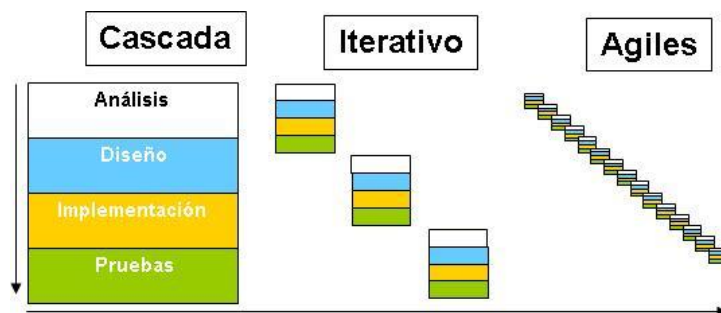
Anexo 2.1. Estructura Estandar de una Metodología



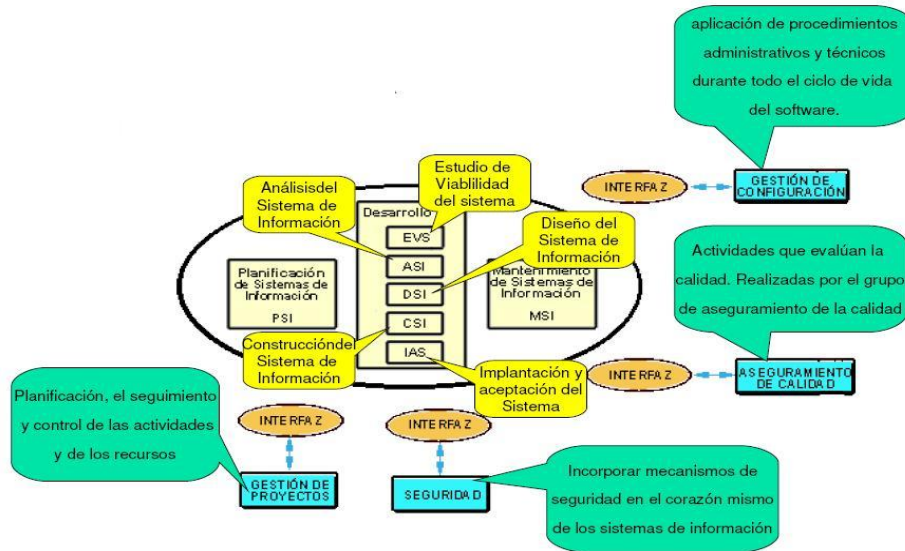
Anexo 3. Nivel de Burocracia



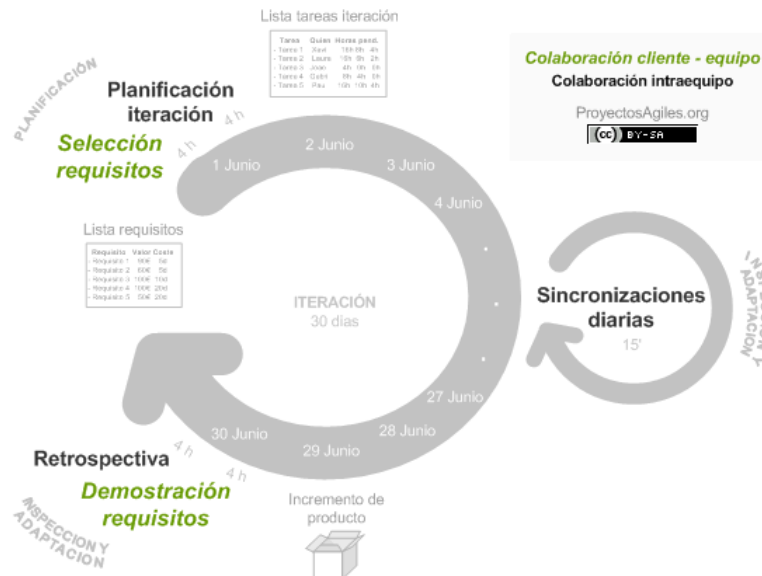
Anexo 4. Ciclo de Vida



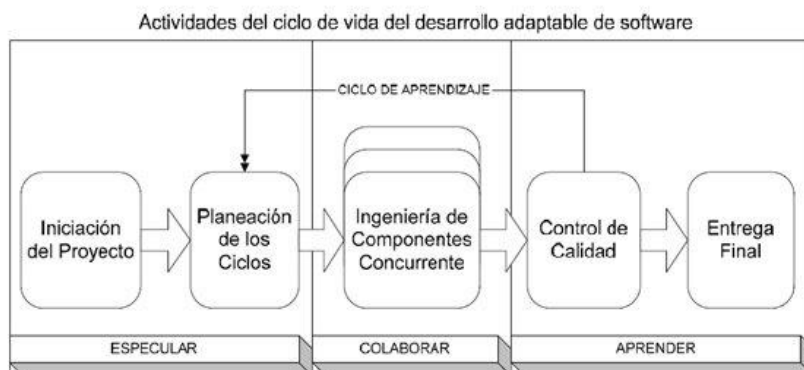
Anexo 5. Metrica V3 Estructura



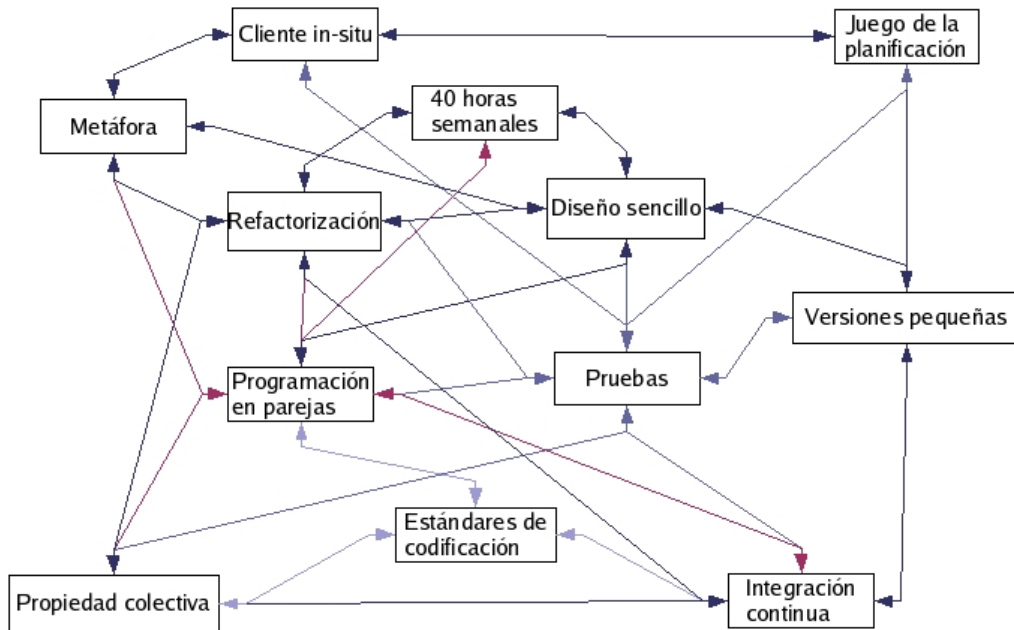
Anexo 6. Scrum



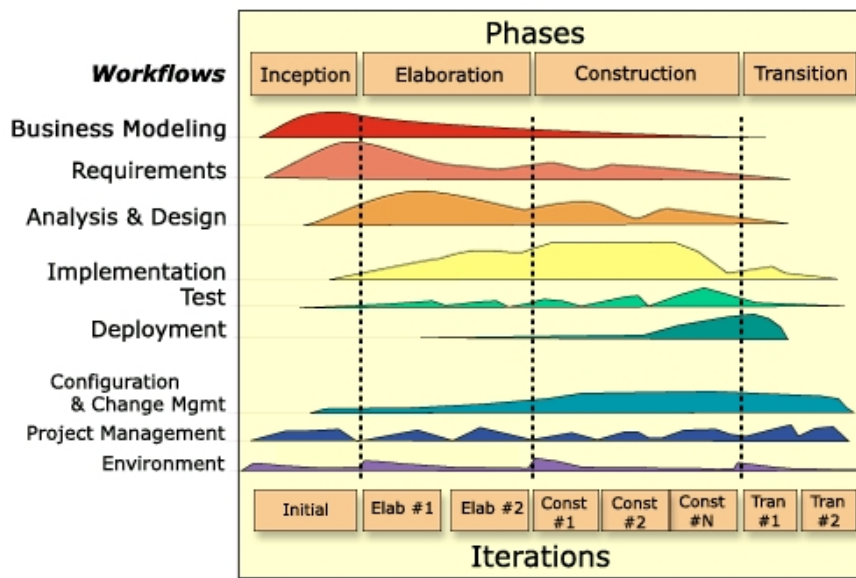
Anexo 7. ASD (41)



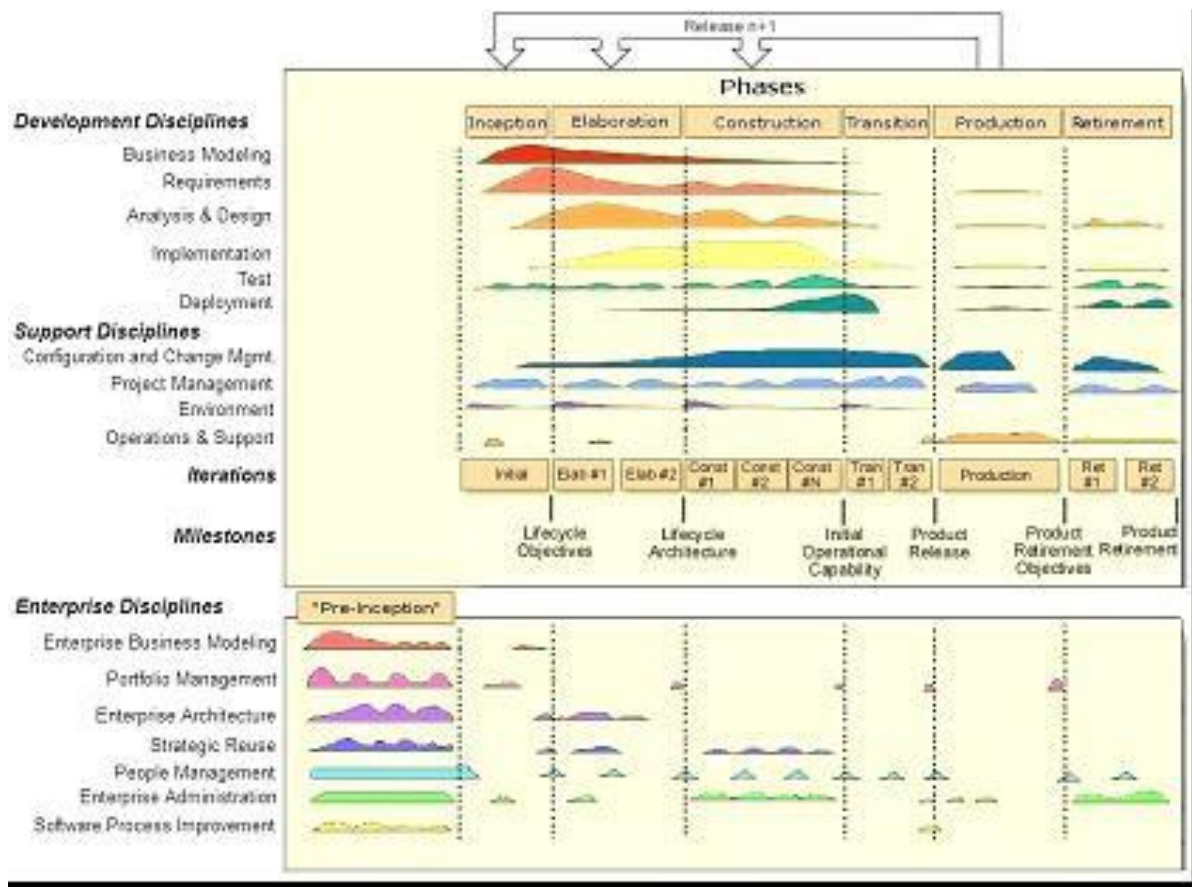
Anexo 8. Las Prácticas XP se refuerzan entre si



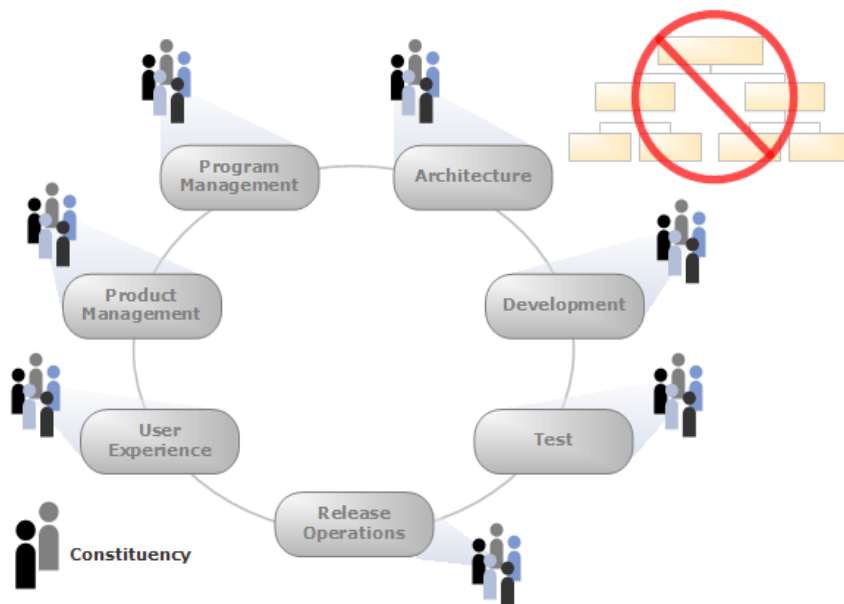
Anexo 9. RUP. Ciclo de Vida



Anexo10. EUP. Ciclo de Vida



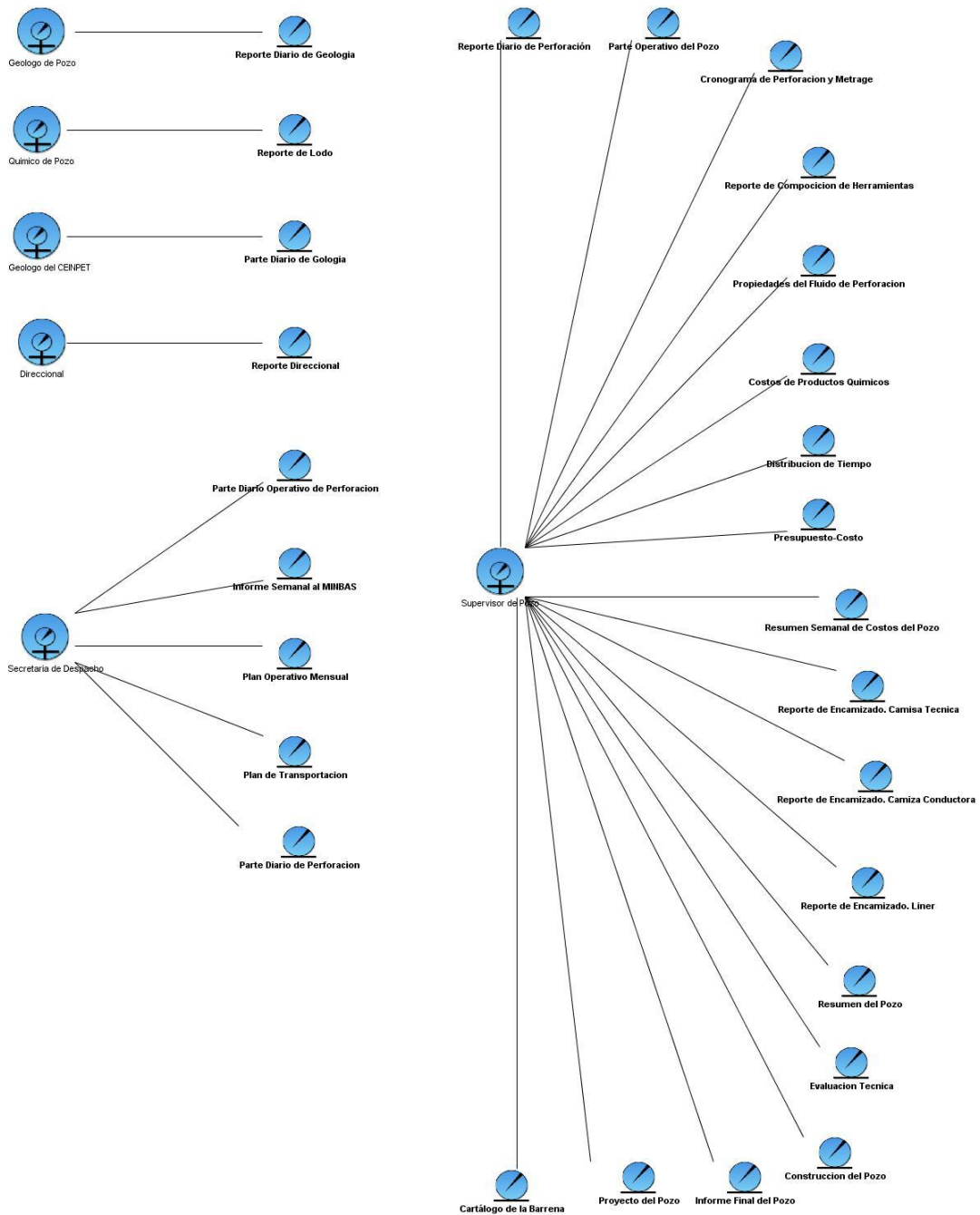
Anexo 11. MSF. Modelo de Equipo



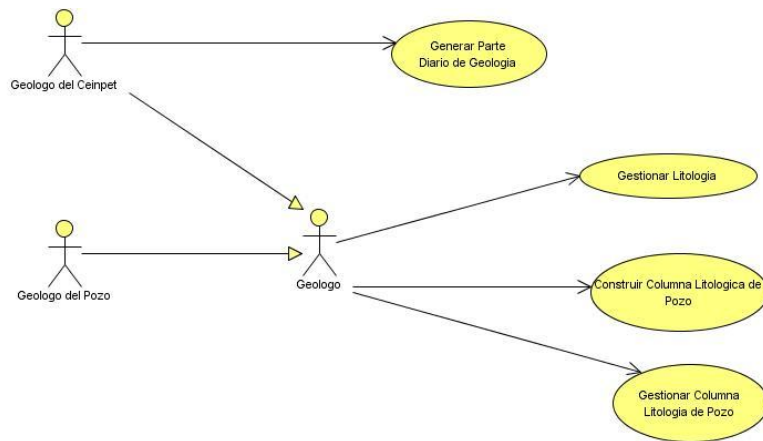
Anexo 12. MSF. Faces



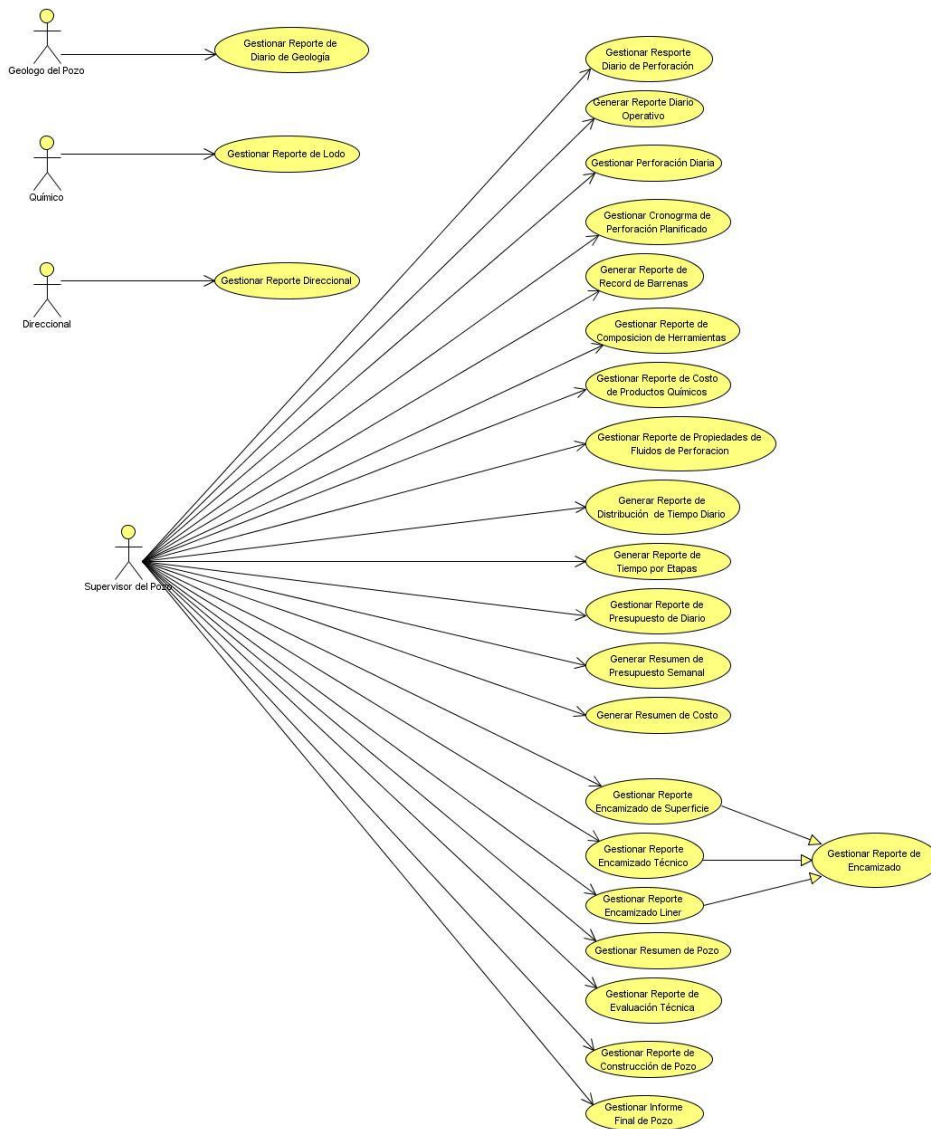
Anexo 13. Modelo de Objetos



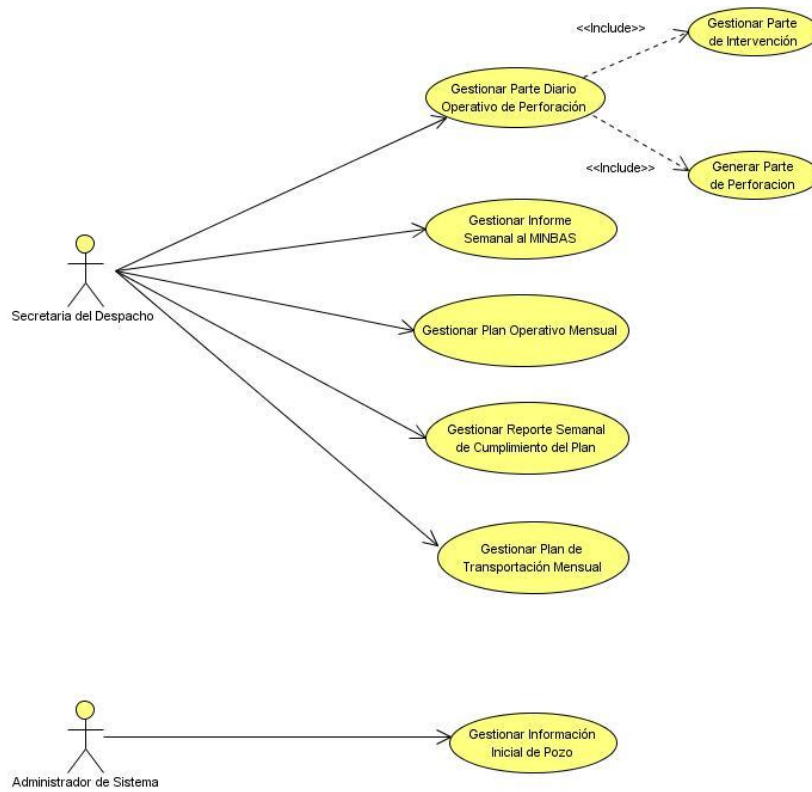
Anexo 14. Subsistema CEINPET



Anexo 15. Subsistema Pozo



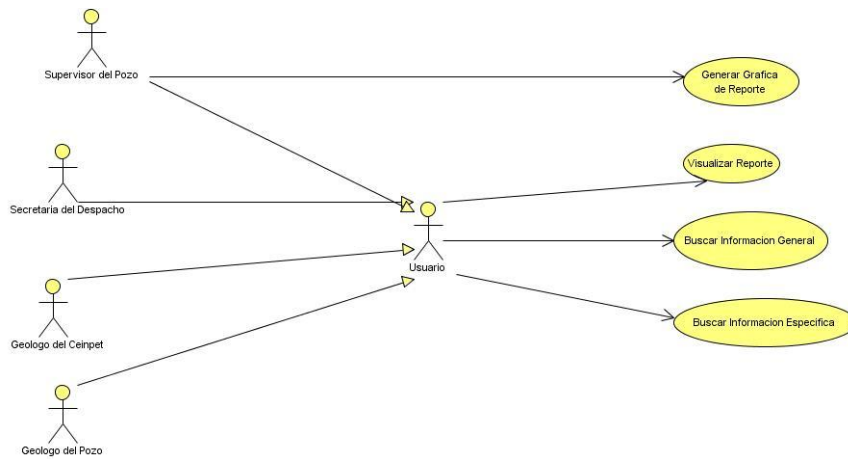
Anexo 16. Subsistema DIPP



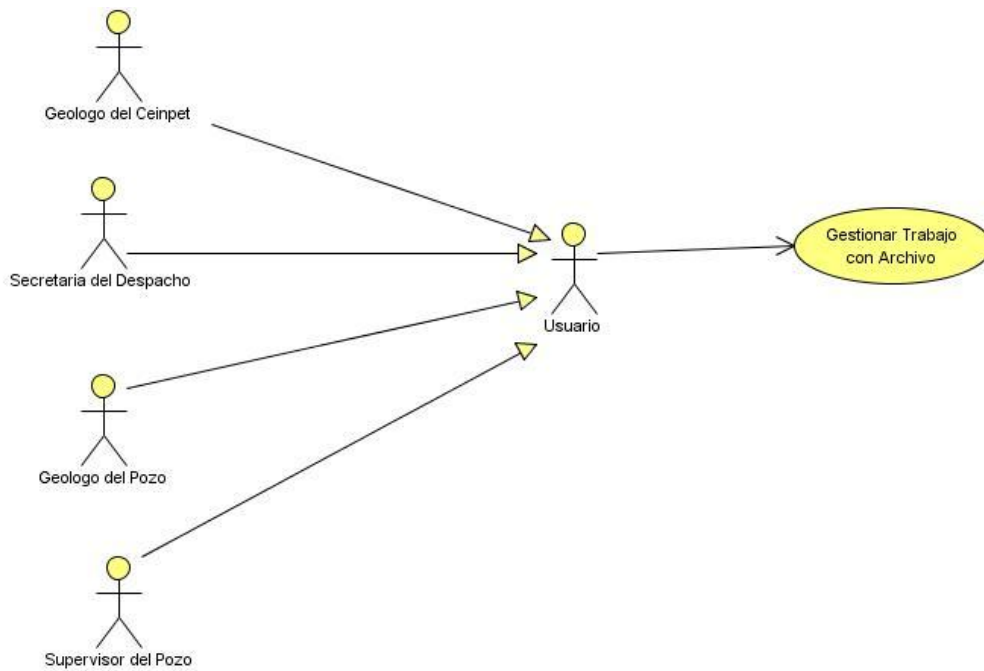
Anexo 17. Subsistema Seguridad



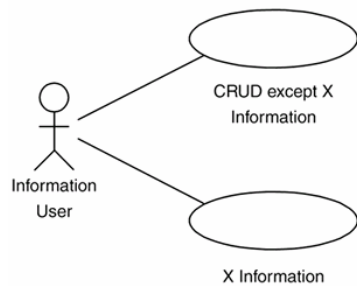
Anexo 18. Subsistema Visualización de Información



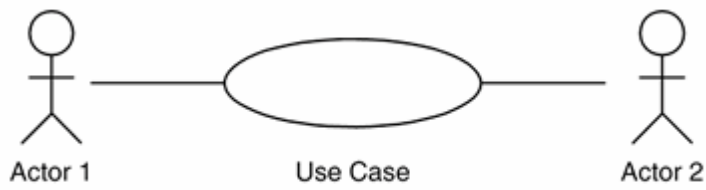
Anexo 19. Subsistema Manejo con Archivos



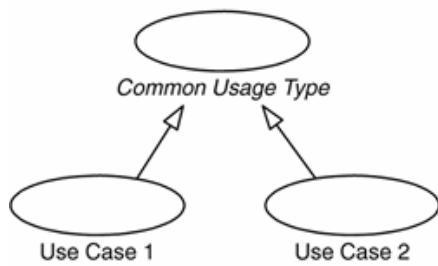
Anexo 20. CRUD Parcial



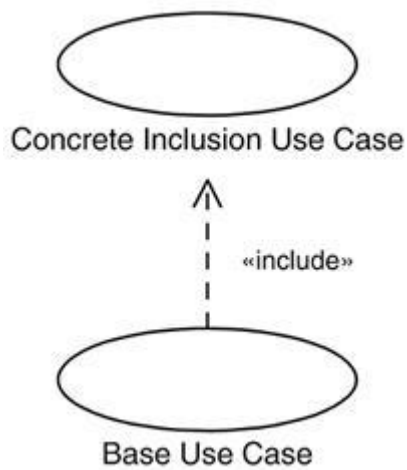
Anexo 21. Múltiples Actores



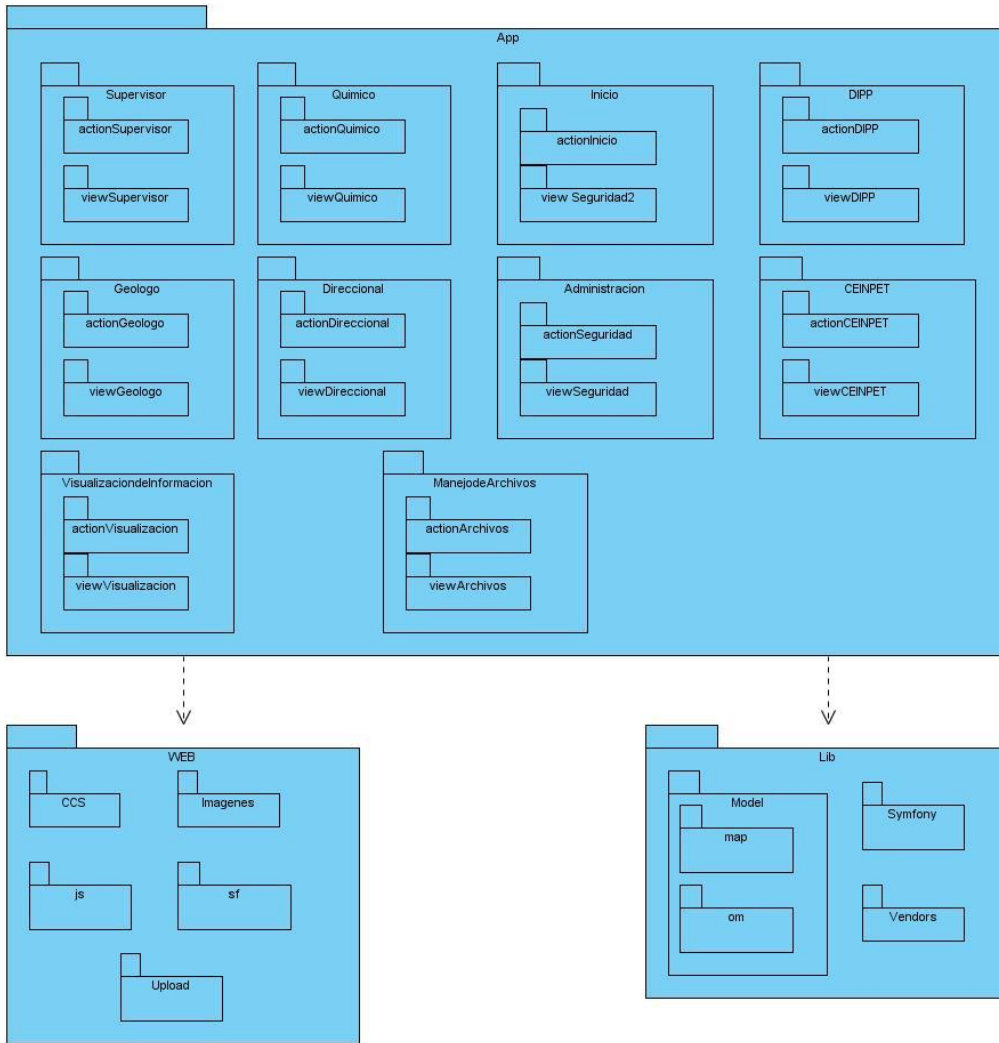
Anexo 22. Concordancia (Commonality): Especialización



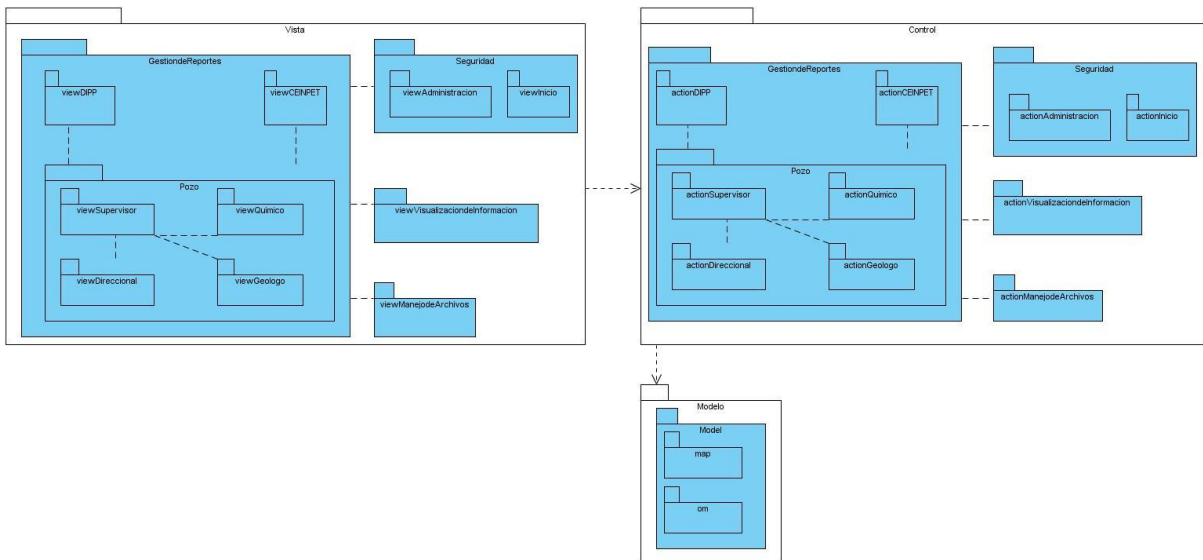
Anexo 23. Concrete Extension or Inclusion: Inclusion



Anexo 24. Vista Lógica



Anexo 25. Vista Arquitectónica por Subsistemas



Glosario de Términos

A

B

C

CUPET: Unión Cuba-Petróleo.

CEINPET: Centro de Investigaciones de Petrleo.

CAD: Estilo para el tratamiento de datos gráficos.

D

DIPP: Dirección de Intervención y Perforación de Pozos.

DISWIN: Proporciona capacidades de envío completas a todas las empresas de transporte por camión clasificadas. DISWIN es el software para maximizar la productividad y hacer ahorrar tiempo y el dinero.

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

Eclipse: Entorno Integrado de Desarrollo, desarrollado por Sun Microsystems.

F

G

H

I

IDE: Integrated Development Enviroment (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

Ñ

O

OMG: Es una organización sin fines de lucro, industria especificaciones consorcio; nuestros miembros definir y mantener la especificación de UML

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.

RQD: *Rock Quality Designation* se define como el porcentaje de recuperación de testigos de más de 10 cm de longitud (en su eje) sin tener en cuenta las roturas frescas del proceso de perforación respecto de la longitud total del sondeo. Además saber que para determinar el RQD en el campo o zona de estudio de una operación minera, existen tres procedimientos de cálculo.

Rational ClearCase: Software de gestión del ciclo de vida y control de los activos de desarrollo de software, así como control integrado de versiones

S

SIPP: Sistema de Manejo Integral de Perforación de Pozos

T

U

V

Visual Studio: Entorno de Desarrollo Integrado desarrollado por Microsoft.

W

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

X

Y

Z