UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE

INGENIERO EN CIENCIAS INFORMÁTICAS

TÍTULO: ***Análisis del Tercer Ciclo de Desarrollo del Proyecto Sistema de Manejo Integral de Perforación de Pozos.***

AUTOR: Reynel Gómez Martínez

TUTOR: Ing. Liliam Celia Beyris Soulary

Co-TUTOR: Ing. David Tavares Cuevas

La Habana, Junio 2011

"Año 53 de la Revolución"

Declaración de Autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 5 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Firma del autor	**Firma del tutor**
Reynel Gómez Martínez	**Ing. Liliam Celia Beyris Soulary**

Resumen

Cada día la necesidad de producir combustible es mayor, dirigiéndose todos los esfuerzos en disminuir los costos de producción. La industria petrolera es la principal y más importante productora de este material, encontrándose inmersa en todo este fenómeno.

Nuestro país se encuentra en una etapa de explotación de sus yacimientos, y no contaba con un software para la gestión de los procesos de la Perforación de Pozos de Petróleo, iniciativa que tomó la Universidad de las Ciencias Informáticas al desarrollar un software para la gestión de dichos procesos. El Proyecto Sistema de Manejo Integral de Perforaciones de Pozos de Petróleo (SIPP), del Centro de Informática Industrial (CEDIN) de la Facultada 5 desarrolla el producto *MaximusDrillPro*, el cual se encuentra en la fase de despliegue y está a punto de liberársele al cliente en su primera versión. En esta etapa fueron identificadas y propuestas por el cliente otras funcionalidades para el producto, las cuales constituyen el punto de partida de este trabajo.

La presente Investigación consiste en el Análisis de elementos relacionados con la Comunicación, Configuración, Internacionalización, Historial de Usuario y la Visualización Dinámica; funcionalidades que ayudarían a proporcionar comodidad del trabajo con el software así como le incorporarían valores agregados al mismo, llegando al estado de que una vez que se diseñe e implemente el modelado propuesto, se podría comercializar el producto a nivel internacional. Para ello se realiza un estudio exhaustivo del Estado del Arte, se escogen las Herramientas y Tecnologías a utilizar, se seleccionan y validan los Requisitos de Software, se construye el Modelo de Dominio, el cual constituye la base para la elaboración de Modelo de Casos de Uso del Sistema, se describen y prototipan los Casos de Usos críticos y secundarios del ciclo de desarrollo, finalizando con el Modelo de Análisis y los Diagramas de Colaboración para estos casos de uso.

Palabras clave: Análisis, caso de uso, comunicación, configuración, historial de usuario, internacionalización, requisitos de software, visualización dinámica.

Índice

Introducción

Al siglo XXl se le ha denominado el siglo de la tecnología de la información. Los sistemas informáticos han alcanzado niveles muy altos de complejidad. Los software de gestión con el transcurso del tiempo han logrado lugares importantes en la economía de muchos países, ya que la información se ha convertido en un componente valioso para el desarrollo de las empresas.

En el desarrollo de la humanidad, las Tecnologías de la Información y la Comunicación (TIC) se han insertado en todas las esferas de la sociedad y en las industrias más importantes para el desarrollo de los países, como es la industria del petróleo.

Se han realizado estudios donde se estima el tiempo de explotación de este recurso natural no renovable al nivel de consumo mundial actual, dando como resultado que las reservas actuales se agotarán en 45 años aproximadamente, por tal razón la eficiencia en cada uno de los procesos dentro de la industria es primordial para la explotación máxima de los reservorios.

Alrededor del mundo existen empresas especializadas en la producción de software empleadas únicamente a la producción de sistemas para la industria del petróleo, los cuales automatizan los procesos fundamentales que se realizan, entres ellas podemos mencionar:

- ***TECNIA:*** Comienza en 1974 ofreciendo inicialmente servicios de Ingeniería para la Industria del Petróleo y del Gas. Hoy, es una empresa global de ingeniería y construcciones, de alta capacidad tecnológica (1).
- ***RCLAB S.R.L (Laboratorios de Estudios Petrofísicos):*** Creado en 1994, se convirtió en empresa líder en proveer servicios a la industria del petróleo como son las ventas de productos principalmente de medición de indicadores, entre muchos otros. (2).
- ***Schlumberger:*** Fundada en 1926 actualmente es la principal y mayor de las empresas proveedoras de productos y servicios en el área del petróleo. Presenta una vasta experiencia en el campo y está avalada por sus altos resultados en compañías de gas y petroleras alrededor del mundo. (3)
- ***Software SAP:*** Implementa un software de gestión empresarial adecuado que ayuda a gestionar las dificultades operativas, a integrar la información, los procesos y proporcionar potentes herramientas para la toma de decisiones que mejoran el rendimiento y reducen (4).

En Cuba la unión Cuba-Petróleo (CUPET), es la empresa que se encarga de todo el proceso relacionado con este recurso, desde la exploración hasta la comercialización. A partir del incremento de la extracción de crudos nacionales, CUPET ha comenzado una fase de automatización de sus procesos en cada una de sus empresas y unidades, imprescindible para mejorar los resultados cualitativos y cuantitativos en cada uno de ellos.

En la Universidad de las Ciencias Informáticas (UCI), se desarrollan sistemas para la automatización de procesos de la Industria del Petróleo. En la Facultad 5, se encuentra el Centro de Desarrollo de Informática Industrial (CEDIN). En este Centro se encuentra el proyecto Sistema de Manejo Integral de Perforación de Pozos (SIPP). Este proyecto desarrolla la solución “MaximusDrillPro”, la cual responde a las necesidades del Centro de Investigaciones de Petróleo (CEINPET) y la Dirección de Intervención y Perforación de Pozos (DIPP), entidad responsable de todos los pozos en perforación en tierra, del territorio cubano.

Este proyecto comenzará su segundo ciclo de vida y ya tiene definido los elementos de análisis y diseño para el desarrollo de la Versión 2.0 de SIPP, versión que aun no cuenta con aspectos relacionados con la configuración de la aplicación; no se ha analizado la posibilidad de incluir elementos que favorezcan la comunicación en tiempo real entre los operadores de la aplicación, así como el uso del mismo por empresas extranjeras que hablen otros idiomas. Por otro lado no se le brinda la posibilidad al usuario de hacer recomendaciones sobre la aplicación, contribuyendo a mejorar la misma. Un último elemento a tener en cuenta es la visualización de reportes, el modelado hecho en las versiones anteriores fue concebido para que el sistema lo hiciera de forma estática, proceso que es lento e ineficiente. Dado que la solución tiene otras características funcionales que no han sido analizadas, se hace necesario su análisis con vistas al desarrollo del tercer ciclo de vida, identificando como **situación problemática** la necesidad que existe de modelar otras características funcionales que no han sido analizadas y que no fueron modeladas en versiones anteriores como los son la Configuración, Comunicación, Historial de Usuario, Internacionalización y la Visualización Dinámica.

Actualmente existe una primera y segunda versión del análisis y diseño de este sistema y es el punto de partida para lograr la modelación del tercer ciclo de vida del proyecto.

Partiendo de la situación expuesta anteriormente se ha definido como **problema a resolver**: ¿cómo propiciar la correcta identificación de las necesidades del cliente en los módulos de Configuración, Comunicación, Historial de Usuario, Internacionalización y la Visualización Dinámica, para el adecuado

análisis del tercer ciclo de desarrollo de SIPP?, definiendo como **objetivo específico** de la investigación: modelar los artefactos correspondientes al Análisis del tercer ciclo de desarrollo del proyecto productivo SIPP.

La investigación estará centrada en los procesos de negocio de la gestión de información en los pozos de petróleo en perforación, lo cual constituye el **objeto de estudio**. Enmarcándose en la automatización de los flujos de información en pozos de petróleo en perforación y en la Dirección de Intervención de Pozos en Perforación de la empresa CUPET, definiéndose como **campo de acción**.

Esta investigación tiene su base **defendiendo la idea** de que mediante la identificación correcta de los procesos y funcionalidades, se podrá obtener un apropiado Modelo de Análisis para el tercer ciclo de desarrollo del Sistema de Manejo Integral de Perforación de Pozos.

Para dar cumplimiento al objetivo específico se realizaran las siguientes **tareas de la investigación**:

1. Elaboración de un marco teórico.
2. Selección de la metodología de desarrollo de software a utilizar, patrones y lenguaje para el modelado efectivo del Análisis del Sistema.
3. Identificación y descripción de los procesos de negocio y las funcionalidades en cada uno de los pozos de petróleo en perforación.
4. Análisis de los casos de uso definidos para el Ciclo de Desarrollo.

Para dar realización a las tareas anteriormente mencionadas se hace necesario aplicar métodos de investigación científica que faciliten el proceso de recogida de información, entre los usados se encuentran los **métodos teóricos** y **métodos empíricos**. Dentro de **los métodos teóricos** se encuentra el **método analítico sintético** y el **histórico lógico**.

El método **analítico sintético** será usado para modelar todo el funcionamiento de los pozos de petróleo en perforación, permite extraer las características más importantes para lograr un entendimiento completo de todos los procesos que se realizan en los mismos, este conocimiento se usará a la hora de elaborar el modelo de negocio o de dominio en dependencia de las características del negocio y el modelo de análisis del sistema.

El método **histórico lógico**, se usará para confeccionar un marco teórico sobre las soluciones que den respuesta al problema en cuestión, su principal objetivo es la recopilación de información referente a las principales empresas en esta área, así como las herramientas utilizadas por estas. Se registran las

investigaciones que precedieron a esta para proveer una mayor base científica que apoye todo el conocimiento obtenido.

Los **métodos empíricos** no dejan de ser importantes y dentro de ellos se encuentran **las entrevistas** y **la observación**. La realización de **entrevistas** es uno de los más importantes pasos en el desarrollo de una investigación, ya que permite conocer interioridades de cada proceso que haciendo un análisis general no podemos deducir, el objetivo principal de este método es extraer el conocimiento sobre diferentes procesos utilizando el "individuo" como forma de información, ya que es el que está en contacto diario con los procesos, también ayuda a esclarecer detalles específicos del funcionamiento de los mismos, que son necesarios para su correcta modelación e implementación.

La **observación** tiene como meta obtener un nivel de realidad sobre los procesos que conforman el objeto de estudio, sin este método se corre el riesgo de no entender totalmente los procesos desarrollados en el mismo.

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

- **Capítulo I:** Se describe un estado del arte y al mismo tiempo se buscan propuestas a nivel internacional y nacional que den solución al negocio en cuestión, estableciendo aspectos comparativos entre las soluciones encontradas. Además se describe el proceso actual y se abordan los aspectos teóricos referentes a los Requerimientos de Software y el Análisis.

- **Capítulo II:** Se identifican las metodologías de desarrollo de software, caracterizando cada una de las metodologías identificadas y se elabora una comparación para fundamentar la selección de la metodología a utilizar. Además de seleccionar la herramienta CASE, el lenguaje en el que será modelado el sistema y las técnicas de captura y validación de requisitos.

- **Capítulo III:** Se representa el Dominio, se describen las funcionalidades, se especifican, prototipan y analizan los Casos de Uso de cada uno de los Módulos que se deben incorporar en el tercer ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción

En este capítulo se abordarán los aspectos generales y conceptos fundamentales para el desarrollo de un marco teórico que sirva de soporte a un mejor entendimiento de los capítulos posteriores, reflejando el desarrollo a nivel mundial de los sistemas informáticos para la gestión de información en los procesos de perforación, así como otros que realicen actividades semejantes para la automatización de procesos comerciales dentro de empresas. Además se dedicará un espacio también a abordar aspectos teóricos referentes a los requisitos de software y el análisis.

1.2 Conceptos asociados al dominio del problema

- **Perforación:** Es la actividad de campo que consiste en mudar un taladro, vestir el equipo y conectar una mecha o una sarta de tubería de acero para penetrar las formaciones hasta construir un hoyo; este se acondiciona y termina convirtiéndolo en un pozo petrolífero (5). La perforación constituye el proceso más importante dentro de la Industria del Petróleo ya que proporciona la certeza de la existencia del hidrocarburo.
- **CUPET:** Empresa que se dedica a la extracción de depósitos de petróleo, refinación y distribución de productos derivados del petróleo, para satisfacer las necesidades del mercado nacional de hidrocarburos y eliminar las importaciones, a partir del incremento de la extracción de crudos nacionales.
- **Configuración de la Aplicación:** Conjunto de datos que determina el valor de algunas variables de una Aplicación, con el fin de obtener un sistema informático personalizado para el usuario.
- **Historial de Usuario:** Almacenamiento en algún lugar de los sucesos o acontecimientos que sirven de indicio o prueba a los procedimientos realizados por los usuarios en una aplicación determinada.
- **Internacionalización:** La práctica de Modelar, Diseñar e Implementar programas que puedan configurarse fácilmente, para interactuar con el usuario en más de un idioma.
- **Comunicación:** Constituye el intercambio de información que realizan los usuarios entre sí a través de la aplicación.

- **Visualización Dinámica:** La información mostrada no permanece estática, va evolucionando a medida que los elementos visualizados lo requieran.

1.3 Objeto de Estudio

El objeto de estudio como consecuencia del planteamiento del problema, delimita la parte de la realidad que es necesario estudiar para solucionar el mismo y lo constituyen los procesos de negocio en la gestión de información en los pozos de petróleo en perforación y en la Dirección de Intervención y Perforación de Pozos, así como el análisis del primer y segundo ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos.
En los pozos de petróleo se generan una serie de reportes e información necesaria para el control y supervisión de los procesos de la perforación, los cuales son enviados a la dirección de intervención y perforación de pozos.

Los diferentes operadores necesitan comunicarse en tiempo real para consultar propuestas y valorar situaciones dadas mediante las vías del correo electrónico y el teléfono. Por otro lado muchas de las empresas contratadas por CUPET son extranjeras y en ocasiones algunos operadores no dominan el idioma español.

El análisis del primer y segundo ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos, que constituyen el punto de partida de esta investigación, permite que se puedan agregar, mejorar y optimizar procesos descritos en estos ciclos, añadiendo nuevas funcionalidades que no son obligatorias para los procesos de perforación, pero que sí le aportan una mayor calidad a los procesos ya descritos e implementados.

1.4 Descripción del Proceso Actual

En Cuba se maneja toda la información proveniente de los pozos en perforación, viajando de un lado al otro por vía telefónica y correos electrónicos. Esta información es fundamental para poder tomar decisiones importantes sobre el futuro de un pozo. Con los dos ciclos de análisis y diseño desarrollados anteriormente se ha llegado a establecer una especie de estándar en los reportes, así como la definición de los diferentes módulos para el producto “MaximusDrillPro”, nombre con el que será liberada la primera versión del proyecto. Muchas veces se necesita establecer una conversación de consultoría, muy breves, entre los supervisores de diferentes pozos y se hace engorroso establecer la conversación vía correo

electrónico; además nuestro país tiene contratos con muchas empresas extranjeras de perforación de pozos, y en ocasiones el supervisor tiene que traducir algunos elementos de reportes y documentos generados, para la comprensión por parte de estas compañías. Otro aspecto importante es que el usuario se sienta cómodo y pueda personalizar un poco su sesión de trabajo en la aplicación, entre otros aspectos configurables que se pueden tener en cuenta.

1.5 Análisis de algunas soluciones existentes

Actualmente existen compañías que brindan diferentes gamas de producciones diferenciadas de software para el sector del Petróleo, que aportan soluciones parciales a algunos de los procesos automatizables. En su mayoría estos software están prediseñados para resolver algunas necesidades fundamentalmente en los procesos de extracción y refinación del petróleo. A continuación se hace referencia a algunas de ellas:

- **Petrokem Loggin Services (PLS):** es una compañía ecuatoriana con más de trece años de experiencia en la industria petrolera de ese país. Ha trabajado para varias operadoras entre las cuales están: Petrosud-Petroriva, Petroproducción, Pacifpetrol, Andes Petroleum Ecuador y el Consorcio Petrolero B14. A cada compañía PLS ha proporcionado un servicio de Control Litológico de una buena calidad con un sistema confiable de adquisición de datos y con un versátil programa capaz de ejercer las tareas de control, cálculo y graficación de todos los requerimientos del cliente en materia de Control Litológico, utilizando el software del Pen Lab. (6).
- **NeuraLog Inc:** ofrece una gran variedad de aplicaciones fácil de usar para la integración, análisis, y captura de datos primordialmente para la industria petrolera. Usando la tecnología de manipulación de imágenes raster de NeuraLog para la captura de la información de pozos, mapas o interpretación rápida de correlaciones y la conectividad directa a estaciones de trabajo o bases de datos de Landmark, Schlumberger/GeoQuest o usando formatos propios y comunes de aplicaciones más usadas en la Industria petrolera en más 70 países en el mundo (7).
- **Schlumberger GeoQuest:** unidad operativa que proporciona software para la gestión de la información, infraestructura y servicios. A través de sus tecnologías y servicios las compañías de petróleo y gas pueden mejorar el rendimiento empresarial, aprovechar el potencial del campo petrolero digital y reducir riesgos de exploración y desarrollo (3).

Por otro lado, existen también otras empresas que se dedican a comercializar software privativo que darían una solución factible a diferentes problemas en la industria petrolífera, solo que se tendría que pagar una licencia para la instalación y uso del producto. A continuación se abordará en algunas de ellas:

- **Well Wizard:** sistema para el muestreo de todos los factores fundamentales en la perforación de pozos como son el aire comprimido y la temperatura que posee el lodo; los cuales son controlados por un software instalado actualmente en todos los pozos en perforación en nuestro país. Por el uso de este software, cada pozo de petróleo en perforación debe pagar aproximadamente 12 000 dólares diariamente (8). Como ventajas de este sistema se pueden mencionar: tiene más de 75 000 bombas de Well Wizard asistentes de la cámara de aire y están en uso más que todas las otras marcas y tipos de muestras de agua y tierra combinados, posee modelos específicos para cada pozo, bajo rendimiento, corta columna de agua, a profundidades de 1000 pies (304.8m), para envolver ID hasta 31,75 mm y una alta confiabilidad probada desde el año 1982, con el primer estándar de la industria de 10 años de garantía.
- **Well Logger:** permite crear informes de perforación de suelo y diagramas de construcción de pozos, ofreciendo una sencilla, aunque robusta, interfaz de usuario que brinda presentaciones personalizables, patrones definidos por el usuario, escala ajustable y vista previa de la impresión. También posee una interfaz de hoja de cálculo fácil de usar, con cajas de entrada que simplifican la entrada de datos para cada perforación incluyendo la litología de la perforación, muestras tomadas, construcción del pozo o detalles anexos de la perforación (9),
- **Software WELLSIGHT:** conjunto de software utilizado para la captación, informes, monitoreo, seguimiento, reconciliación y exportación de datos. Inicialmente 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 países del mundo. 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). (10)
- **Peloton:** es una empresa líder en el desarrollo de sistemas de información de Construcción y Operaciones de pozos. 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. Hasta la fecha, esta empresa se ha auto-financiado y ha administrado su crecimiento cuidadosamente

mientras mantiene los estándares de calidad que son fundamentales para nuestro futuro. Más de 180 empresas petroleras en el mundo usan esta aplicación de Peloton (11).

- **Sistema Manejo Integral de Perforación de Pozos (SIPP):** la industria del petróleo en nuestro país, como en casi todo el mundo, se dividen en tres grandes procesos: Exploración-Perforación-Producción, Refinación y Comercialización. Las entidades especializadas en el proceso de Exploración-Perforación-Producción son las encargadas de la exploración y desarrollo de los campos, la perforación de nuevos pozos, así como la reparación a los que ya se encuentran en producción (Intervención de Pozos).

  El SIPP involucra al Centro de Investigaciones del Petróleo (CEIPET), los pozos en perforación, la Dirección de Intervención y Perforación de Pozos (DIPP), entidad única en el país que dirige y controla las actividades de la perforación e intervención de pozos de petróleo.

  En el primer ciclo de desarrollo del proyecto SIPP se modelaron 24 casos de usos del sistema, alcanzando ya con el segundo ciclo los 70; casos de usos que ya han sido implementados y están próximos a pasar a la etapa de liberación del Producto en su primera versión. Este sistema actualmente se encuentra incompleto ya que existen algunas funcionalidades no operativas que no han sido identificadas y son necesarias, no tanto para un correcto funcionamiento del producto en general, pero si le brindan valor agregado al mismo y la posibilidad de que se pueda desplegar no solo en las empresas de CUPET, sino también en otros países que deseen comprar esta solución.

Después de analizar algunas de las soluciones existentes, se llega a la conclusión de que la mayoría de los sistemas antes expuestos han alcanzado un elevado prestigio en la rama de la informática, vinculada a la industria petrolera, pero tienen algunos inconvenientes que hacen que el uso del producto SIPP que se pretende ofrecer, sea fundamental para Cuba y competitivo a nivel internacional; algunos de estos inconvenientes son:

En el ámbito Mundial.

- Todos son software sobre la plataforma de escritorio.
- Estos sistemas pertenecen a empresas canadienses y estadounidenses por lo que se encuentran bajo licencias propietarias a un elevado costo. Además de que algunos serían imposibles de comprar debido a la política restrictiva de EE.UU implantada a nuestro país.

En el marco de Cuba.

Nuestro país hasta el momento no cuenta con un software diseñado para la gestión de la información en los procesos de perforación, de ahí la iniciativa de la Universidad de las Ciencias Informáticas en desarrollar una aplicación web que facilite estos proceso, evitando redundancias e incoherencias que pueden ser generadas con el trabajo manual y favoreciendo al país en cuanto a:

- El producto sería completamente gratuito para CUPET, así como todas sus actualizaciones.
- Debido a que todos los procesos de negocio han sido tomados en un pozo de petróleo localizado en la provincia de Matanzas, constituirá el software que más se adapta a las particularidades de la industria petrolera nacional cumpliendo con todas sus especificaciones.

1.6 Requisitos de Software

La Standard Glossary of Software Engineering Terminology (IEEE) define a un requisito como una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
Los requisitos de un software, pueden clasificarse en requisitos funcionales y requisitos no funcionales. Los ***requisitos funcionales*** constituyen las capacidades o condiciones que el sistema debe cumplir; y es en la realización de los casos de uso del negocio, donde se obtienen las actividades objeto a automatizar, las cuales constituyen el punto de partida para identificar qué debe hacer el sistema. Una característica a destacar de los requisitos funcionales es que se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

Por otro lado los ***requisitos no funcionales*** son las 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, en la mayoría de los casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se podrá determinar cómo ha de comportarse. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. Existen múltiples categorías para clasificar a los requisitos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, las cuales son: Requisitos de Software, Requisitos de

Hardware, Restricciones en el diseño y la implementación, Requisitos de apariencia o interfaz externa, Requisitos de Soporte, aunque estas no limitan a la definición de otros (12).

1.6.1 Ingeniería de Requisitos

La Ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en el cual trabaja, incluyendo el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es verdaderamente lo que el cliente quiere y cómo interactuarán los usuarios finales con el software. Su importancia está enmarcada en que el diseño y la construcción de elegantes programas que resuelvan el problema incorrecto, no satisface las necesidades de ningún cliente. Por lo tanto es importante atender lo que el cliente quiere antes de comenzar a diseñar a construir cualquier software.

El objetivo del proceso de la ingeniería de requisitos es darle a todas las partes una explicación escrita del problema, esto pudiera lograrse por medio de varios productos de trabajo: escenarios de uso, lista de funciones y características, modelos de análisis o algunas especificaciones (13).

1.6.1.1 Funciones de la ingeniería de requisitos

El proceso de ingeniería de requisitos se lleva a cabo a través de siete distintas funciones: *inicio, obtención, elaboración, negociación, especificación, validación y gestión*. Es importante destacar que algunas de estas funciones ocurren paralelamente y que todas deben adaptarse a las necesidades del proyecto.

Al ***Inicio*** del proyecto los ingenieros de software deben hacer una serie de preguntas libres de contexto, con el objetivo de establecer una comprensión básica del problema. La ***Obtención*** parece muy simple a la hora de preguntarles a los usuarios u otros interesados cuáles son los objetivos para el sistema o producto, qué es lo que debe lograr, de qué forma el producto satisface las necesidades del negocio y como se utilizará el sistema o producto día a día. En la ***Elaboración*** se expande y refina la información conseguida con el cliente durante las dos funciones expuestas anteriormente, enfocándose en el desarrollo de un modelo técnico refinando las funciones, características y restricciones del software. En la ***Negociación*** se les pide a los clientes/usuarios y otros interesados que ordenen los requisitos y seguidamente se discutan los conflictos relacionados con la prioridad. Se identifican y analizan los riesgos asociados a cada requisito y se estima preliminarmente el esfuerzo

requerido para su desarrollo. La ***Especificación*** hace que los requisitos sean más entendibles y por ende más consistentes. En sistemas grandes se recomienda un documento escrito que combine descripciones en el lenguaje natural y modelos gráfico, por otro lado en cuanto a productos o sistemas más pequeños podría ser que no se necesite más que escenarios de uso. En la ***Validación*** se evalúa la calidad de los productos de trabajo y examina la especificación para asegurar que todos los requisitos de software se han establecido de manera precisa; que se han detectado las inconsistencias, omisiones y errores y que estos han sido corregidos y que los productos de trabajo cumplen con los estándares establecidos para el proceso, proyecto o producto. Los requisitos cambian y el deseo de cambiarlos persiste durante la vida del sistema, jugando aquí el papel fundamental la ***Gestión de Requisitos***, la cual no es más que un conjunto de actividades que ayudan al equipo de proyecto a identificar, controlar y rastrear los requisitos y los cambios a éstos, en cualquier momento mientras se desarrolla el proyecto.

1.7 Análisis

1.7.1 Análisis Orientado a Objetos

En el desarrollo de software orientado a objetos no es suficiente usar un lenguaje orientado a objetos, también se necesita realizar un Análisis Orientado a Objetos (AOO), teniendo como habilidad más importante la de asignar eficientemente las responsabilidades a los componentes de software.

El Análisis Orientado a Objetos se basa en los conceptos: objetos, atributos, clases y miembros, todos y partes, y 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, de definir todas las clases que son relevantes al problema a resolver, las operaciones, atributos, relaciones, y comportamientos asociados al mismo, consiguiendo una abstracción mayor que el análisis estructurado, el cual modela los sistemas desde un punto de vista más próximo a su implementación en un ordenador (entrada/proceso/salida). De esta forma, podemos obtener rápidamente un prototipo del sistema, que pueda ser evaluado por el cliente (13).

El modelado visual es la clave para realizar el AOO. Desde los inicios del desarrollo de software Orientado a Objetos (OO) han existido diferentes metodologías para hacer el modelado, actualmente tenemos al Lenguaje Unificado de Modelado (UML) como un estándar para el modelado de sistemas OO, creado por la compañía Rational la cuál desarrolló especificaciones del estándar, y lo abrió al mercado. La misma empresa creó uno de los programas más conocidos hoy en día para este fin, el Racional Rose, pero también existen otros programas como el Poseidón que trae licencias del tipo *community edition* que permiten su uso libremente (14).

1.7.2 Análisis según RUP

La Metodología RUP (Rational Unified Process por sus siglas en inglés, ver Epígrafe 2.2.1.2) genera casi la totalidad de los artefactos correspondientes al flujo de trabajo Análisis y Diseño, en la fase de Elaboración.

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. El Análisis consiste en obtener una visión del sistema que se preocupe de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis (15).

El *Modelo de Análisis* es una abstracción del sistema y permite aspectos de la implementación, por lo que se considera un modelo conceptual, es genérico con respecto al diseño ya que es aplicable a varios diseños, utiliza tres estereotipos conceptuales sobre las clases (Interfaz, Control y Entidad), es menos formal y más dinámico, constituye un bosquejo del diseño del sistema incluyendo la arquitectura, puede no estar mantenido en todo el ciclo de vida de software y define una estructura que es una entrada esencial para modelar el sistema, incluyendo la creación del modelo de diseño. (16)

1.8 Patrones de Casos de Uso

Al igual que cualquier patrón, los patrones de Casos de Uso no son más que relaciones que pueden ser reutilizadas al modelar muchas aplicaciones, a continuación se argumentará en algunos de los más usados en el desarrollo de software.

- **CRUD** (Creating, Reading, Updating, Deleting, por sus siglas en inglés): Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual y puede ser Total o Parcial, en el caso de ser 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.
- **Múltiples Actores Roles Comunes:** Puede suceder que los dos actores jueguen el mismo rol sobre el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol, aplicándolo 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.
- **Concordancia** (Commonality): Consiste en extrae una sub-secuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso expresándolo por separado.

  La **Generalización/Especialización** constituye otro patrón de concordancia que contiene casos de uso del mismo tipo. Implica que el caso de uso “Hijo” hereda todos los atributos, secuencias de comportamiento, puntos de extensión y relaciones definidas en el caso de uso “Padre”, así como también puede definir nuevas operaciones y redefinir o enriquecer con nuevas secuencias de acciones y operaciones ya existentes en el “Padre”
- **Concrete Extension o Inclusion**

  Include: una relación include entre dos Casos de Uso indica que el comportamiento definido en el Caso de Uso a adicionar, es incluido en un lugar dentro de la secuencia del comportamiento realizado por una instancia del Caso de Uso base.

  Extend: establece que un Caso de Uso puede ser extendido con algún comportamiento adicional definido en otro Caso de Uso. La relación contiene una condición y referencia una secuencia de puntos de extensión en el Caso de Uso base. Por lo general es usada para mostrar situaciones de comportamiento opcional, comportamiento que es ejecutado solamente

bajo ciertas condiciones y flujos distintos que pueden ejecutarse en base a la selección del actor.

De alguna forma, estos patrones nos permiten identificar y completar los casos de uso básicos expuestos por el cliente, para un futuro modelado de los diagramas de casos de usos, los cuales, determinan los requisitos funcionales del sistema, representando las funciones que el mismo puede ejecutar.

1.9 Conclusiones Parciales

Con el desarrollo de las Tecnologías de las Informática y las Comunicaciones (TIC) el proceso de informatización de la Sociedad se está convirtiendo en un hecho cada vez más necesario, permitiendo ofrecer soluciones factibles para casi todas las ramas en el desarrollo industrial.

Existen poderosas herramientas que sirven de apoyo en la construcción de software, pero para usarlas correctamente se debe tener el conocimiento necesario sobre el problema que se enfrenta. De ahí que surge la necesidad de conocer los conceptos asociados al dominio del problema, así como algunas de las empresas que brindan soluciones parciales al problema u otras que se dedican a la construcción de software de gestión para pozos de petróleo, todas ellas utilizando software propietario; sin dejar de conocer las ventajas que traerá consigo el Análisis del Tercer Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos.

Capítulo 2: Tendencias y Tecnologías Actuales a Desarrollar.

2.1 Introducción

En este capítulo se realiza una panorámica sobre las tendencias y tecnologías actuales utilizadas para el desarrollo de software. Con el auge de la TIC se hace necesario automatizar los procesos de negocio en cada empresa, para así, optimizar el flujo de información, haciéndolo de manera más segura y rápida. Para ello se abordarán las metodologías de desarrollo de software más utilizadas a nivel mundial, se argumentará el por qué de la utilización de RUP para el análisis de este ciclo de desarrollo; además del Lenguaje y Herramienta CASE utilizada para el modelado.

2.2 Metodologías de Desarrollo de Software

Una metodología de Desarrollo de software no es más que un proceso que define *quién* está haciendo *qué*, *cuándo* y *cómo* para alcanzar un determinado objetivo (15).
Para el desarrollo efectivo de un software y para que queden todas las partes complacidas con el mismo, se depende de un sin número de requisitos y actividades, además de que hay que pasar por muchas etapas. La elección de la metodología de desarrollo se convierte en un paso importante y trascendental en el éxito del producto final. Estas tienen como principal objetivo el servir de guía y organizar las actividades para el cumplimiento y realización de todas las metas trazadas.

Las metodologías de desarrollo de software se pueden ubicar en dos grupos fundamentales:

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

Dentro de las metodologías más usadas se encuentran el Proceso Unificado de Rational (RUP), la programación extrema (XP), Microsoft Solution Framework MSF, Scrum, Iconix, Cristal Methodologies y

AUP. Existen disímiles metodologías de desarrollo en la actualidad, así como categorías para diferenciar las mismas según sus características. En la siguiente tabla se puede hacer una comparación entre las metodologías ágiles y tradicionales para escoger la más adecuada según las características del proyecto a desarrollar:

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

Tabla 1: Diferencias entre las Metodologías Ágiles y Tradicionales

2.2.1 Metodologías Robustas o Pesadas

2.2.1.1 Métrica V3

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, permitiendo alcanzar los objetivos siguientes: (17)

- Definir o proporcionar 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 involucrados 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 de software obtenidos.

Para la conformación 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 detectadas. 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 (17).

2.2.1.2 Rational Unified Process RUP

La metodología Rational Unified Process (por sus siglas en inglés RUP), se divide en 4 fases de desarrollo Inicio, Elaboración, Construcción y Transición, con sus respectivos hitos.
Cada una de estas etapas es desarrollada mediante un ciclo de iteraciones que consisten en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes. RUP muestra como modelar software visualmente para capturar la estructura y comportamiento de arquitecturas y componentes. Las abstracciones visuales ayudan a comunicar diferentes aspectos del software, comprender los requisitos, ver como los elementos del sistema se relacionan entre sí, mantener la consistencia entre diseño e implementación y promover una comunicación precisa. El estándar UML (Lenguaje de Modelado Unificado), creado por Rational Software, es el cimiento para una modelado visual exitoso y constituye el lenguaje de modelado que preferiblemente se debe utilizar con esta metodología (18).

- La característica más peculiar de esta metodología es la de ser ***Iterativo e Incremental*** proponiendo que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros ya que

es práctico dividir el trabajo en pequeñas porciones. Cada una de estas partes es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo y los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, es decir, deben ser seleccionadas y llevadas a cabo de una manera planeada (19).

La iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto y trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior. La iteración proporciona un resultado completo, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. Para ello, cada requisito se debe completar en una única iteración: el equipo debe realizar todas las tareas necesarias para completarlo y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos (19). En cada iteración el equipo evoluciona el proyecto o producto haciendo una entrega incremental a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos requisitos o mejorando los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la priorización de los requisitos en función del valor que aportan al cliente.

- Es una metodología ***Dirigida por Casos de Uso***, ya que un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos, constituyen el modelo de casos de uso, el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas (dirigen el proceso de desarrollo), los casos de uso son desarrollados a la par con la arquitectura del sistema (la arquitectura del sistema) y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el ciclo de vida del proyecto de desarrollo. Los casos de uso reflejan lo que los usuarios finales necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimiento, y es a partir de este momento donde los

casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- RUP está ***Centrado en la Arquitectura***. Para comprender lo antes mencionado hay que tener en cuenta que el concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros stakeholders[1], y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados y los requisitos no funcionales. La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. La arquitectura debe proveer espacio para la realización de todos los casos de uso, ambos deben evolucionar en paralelo, mostrando la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML. (18).

2.2.2 Metodologías Ágiles

2.2.2.1 AUP

Agile Unified Process (por sus siglas en inglés AUP), es una versión simplificada de la metodología Rational Unified Process(RUP), describiendo los procesos de una manera muy simple, fácil de comprender y enfocándose al desarrollo de software para aplicaciones empresariales, usando técnicas y conceptos ágiles que se mantienen fieles a las de RUP.

[1] Cualquier persona o entidad afectada por las actividades de una organización. Estas partes interesadas pueden ser internas, dentro de la propia compañía o externas como clientes, proveedores, competencia, asociaciones.

Al observar el ciclo de vida de AUP. El primer aspecto que da a resaltar es que los flujos de trabajo ha cambiado, primeramente la disciplina de Modelo incluye toda la Modelación del Negocio, Sistema, Análisis y Diseño de RUP, el flujo de Modelado es una de las partes más importantes de AUP, pero no domina todo el proceso. Un segundo aspecto lo constituyen los flujos de trabajo de Configuración y Cambio de Gestión, los cuales son fusionados en el flujo de Gestión de la Configuración. Las metodologías ágiles por lo general a las actividades de Gestión de Cambio las incorporan en el flujo de Modelo. Esta metodología posee las siguientes fases: **Incepción, Elaboración, Construcción y Transición,** además de los flujos de trabajo: **Modelado, Implementación, Testeo, Despliegue, Gestión de la Configuración, Gestión de Proyectos y Ambiente** (20).

2.2.2.2 Extreme Programming (XP)

Esta es la metodología más destacada de todos los procesos de desarrollos ágiles, fue formulada por Kent Beck. La programación extrema se diferencia de las otras metodologías tradicionales en que pone más énfasis en la adaptabilidad que en la previsibilidad. Dentro de las características fundamentales del método se encuentran desarrollo Iterativo e incremental que permite pequeñas mejoras; pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión; programación por parejas, permitiendo que el código sea revisado y discutido mientras se escribe. Propone que la mayor calidad del código escrito de esta manera sea más importante que la posible pérdida de productividad inmediata; frecuente interacción del equipo de programación con el cliente o usuario y la corrección de todos los errores antes de añadir nueva funcionalidad. Otras de sus características más importantes incluyen la Refactorización del código, que no es más que, reescribir ciertas partes del código para aumentar su legibilidad y mantenimiento pero sin modificar su comportamiento. Propiedad del código compartida, este método promueve que todo el personal pueda corregir y entender cualquier parte del proyecto; simplicidad en el código. La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos se tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores (19).

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial en cuanto al costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione, apostando por un crecimiento lento del costo del cambio.

2.2.2.3 Crystal Methodologies

Desarrolladas por Alistair Cockburn y constituye un conjunto de metodologías para el desarrollo de software, caracterizándose por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. 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, como se muestra en la siguiente tabla (22), por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros).

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

Se 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 costo de la comunicación y la mejora individual y global del equipo. Dando 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 y distintas políticas a seguir (23).

2.2.2.4 Dynamic Systems Development Method (DSDM)

Nace en el año 1994 con el objetivo de crear una metodología de Desarrollo de Aplicaciones Rápidas. Define el marco para desarrollar un proceso de producción de software y su principal característica está determinada por ser un proceso iterativo e incremental donde el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases (22).

Es integrable con otras metodologías como RUP o XP y se emplea para el desarrollo de sistemas de información, considerando al cliente o usuario final como un miembro más del equipo de desarrollo; el

analista o diseñador del sistema se encarga de presentar una propuesta de cómo puede quedar el sistema y el cliente dará su punto de vista.

Etapas de desarrollo:

- **Estudio de factibilidad:** donde se realiza un análisis de requerimientos humanos, materiales y financieros necesarios para llevar a cabo el proyecto y se identifican los problemas más importantes de la empresa o negocio del cliente.
- **Estudio del negocio:** cómo planificar las actividades de la empresa.
- **Interacción del modelo funcional:** plantear un modelo o esquema que brinde una solución previa y aceptable a los problemas, es la etapa de diseño del sistema.
- **Interacción de diseño y construcción:** se comienzan a programar los módulos que integrarán el sistema de información y software, conforme a un enfoque de calidad, desarrollo de manuales del sistema y del usuario.
- **Implementación:** entrega del sistema al cliente o usuario final (24).

Después de haber investigado sobre el conjunto de metodologías antes expuestas, las cuales pueden ser utilizadas en el desarrollo de un proyecto de software, se llega a la conclusión de que no existe una metodología que por sí sola facilite el desarrollo exitoso de un proyecto de software. La metodología debe ser adaptada al ambiente en que se desenvuelve el proyecto. Una opción posible es el uso de las metodologías tradicionales, que intentan abordar la mayor cantidad de documentación y artefactos de un proyecto, exigiendo considerables esfuerzos para ser adaptadas, sobre todo, en proyectos más pequeños y por otro lado las metodologías ágiles que ofrecen una solución casi a medida para una gran cantidad de proyectos de media o alta envergadura.

2.3 Selección de la metodología adecuada

Atendiendo a las características de cada una de las metodologías abordadas en el epígrafe anterior (Epígrafe 2.2), se puede establecer una comparación entre las mismas y definir de esta forma cuál es la más adecuada para desarrollar una solución de acuerdo a las características específicas observadas en el desarrollo de este trabajo.

Analizando a simple vista se resaltan rasgos importantes a tener en cuenta para decantar una de las metodologías abordadas: las escasas oportunidades de intercambio efectivo con el cliente y el tener los requerimientos bien definidos, con pocas probabilidades de cambiar en el transcurso del desarrollo del software, si se analizan las características de Scrum también se ven minimizadas sus potencialidades por la constancia en los requisitos y los escasos pronósticos de cambios, por tanto se descarta su utilización.
Por otra parte valorando las características de las metodologías pesadas, las cuales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, mientras que las ágiles están basadas en heurísticas provenientes de prácticas de producción de código, siendo ineficiente aplicarlas en un sistema donde se pueden identificar y definir claramente sus requerimientos tanto funcionales como no funcionales, así como que permiten que se produzcan cambios durante el proyecto; y teniendo en cuenta que se cuenta con un equipo de trabajo organizado por roles y un proyecto que no es pequeño; es posible la utilización de RUP, Métrica V3 o SSADM v4 eficientemente.

RUP es una metodología flexible y lo principal es que determina los límites y el alcance del proyecto en desarrollo, y constituye la metodología que fue utilizada para el desarrollo de las dos versiones anteriores del proyecto.

En la siguiente tabla se establece una comparación entre las metodologías más utilizadas mediante varios criterio (mientras mayor sea el número más elegible será la metodología). La información ofrecida en esta tabla y lo anteriormente expuesto constituye la fundamentación sobre el por qué continuamos con RUP como metodología de desarrollo de software para esta versión del desarrollo del proyecto.

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

2.4 Selección del Lenguaje de Modelación

Siguiendo la Metodología de Desarrollo de Software seleccionada (RUP) y para el modelado del tercer ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos, se decide continuar con el uso del Lenguaje Unificado de Modelado (por sus siglas en inglés UML), el mismo constituye el lenguaje más usado y conocido a nivel internacional. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Utiliza 3 categorías dentro de las cuales se incluyen 13 tipos de diagramas para la modelación como se estructura a continuación (28).

Diagramas de Estructura Estática.	**Diagramas de Comportamiento.**	**Diagramas de Interacción** (Constituyen un subtipo de Diagramas de Comportamiento).
• Diagrama de Clases. • Diagrama de Objetos. • Diagrama de Componentes. • Diagrama de Paquetes. • Diagrama de Despliegue. • Diagrama de Estructura Compuesta.	• Diagrama de Casos de Uso. • Diagrama de Estado. • Diagrama de Actividades	• Diagrama de Secuencia. • Diagrama de Comunicación. • Diagrama de Tiempos. • Diagrama de Vista de Interacción.

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

2.5 Herramientas CASE y procedimientos más utilizados

Las herramientas Computer Aided Systems Engineering (CASE), se pueden definen como el conjunto de software (programas y ayudas) que dan asistencia a los analistas, ingenieros de software y desarrolladores reduciendo el costo en cuanto a términos de tiempo y dinero. También se puede definir como el conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Se puede ver a las CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales. No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil incluirlas en una clase determinada. Algunas de las clasificaciones podría ser atendiendo a: las plataformas que soportan, las fases del ciclo de vida del desarrollo de sistemas que cubren, la arquitectura de las aplicaciones que producen y su funcionalidad.

La realización de un software requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las Herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software (30).

2.5.1 Rational Rose Enterprise Edition

Rational Rose Enterprise Edition es una de las herramientas CASE más poderosas y populares que existen. A continuación se abordan alguna de las características principales que posee esta herramienta por lo cual es catalogada de tal forma:

- Permite la generación automática del código a partir de los modelos, en diferentes lenguajes de programación.
- Permite realizar el diseño y el análisis del sistema antes de la codificación del mismo.
- Puede ser integrada a varios de los más importantes entornos de desarrollo.
- En la Ingeniería inversa, permite generar los diagramas una vez conocido el código.
- Para el Análisis de la calidad del código (29).

2.5.2 Enterprise Architect 7.5 (EA)

Enterprise Architect 7.5 es una herramienta de construcción y modelado de software de alto rendimiento, propiedad de la empresa SPARX SYSTEMS. Es flexible, completa y potente al modelado en UML. Provee un alto desarrollo en cuanto a sistemas, administración de proyectos y análisis de negocio. Abarca integralmente el ciclo de vida, cubriendo el desarrollo de software desde el relevamiento de los requerimientos, a través de las etapas de análisis, modelos de diseño, testing y finalmente el mantenimiento y re-uso. Fue construido en base al excepcional éxito de las versiones previas con un completo soporte para el estándar UML 2.2, donde los modeladores tienen todo el poder y la expresividad de los 13 diagramas de UML 2.2 (Epígrafe 2.4).

Permite la integración con los IDE[2] Visual Studio y Eclipse, para el trabajo en equipo; se integra con la herramienta de integración Teamcenter Systems Engineering, soportando la generación e ingeniería inversa de código fuente de los lenguajes C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP , 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 (31).

2.5.3 Visual Paradigm for UML 6.1 Enterprise Edition

Visual Paradigm for UML (VP-UML) es un herramienta CASE que posibilita el diseño y está creada para ayudar al desarrollo de software, soportando estándares clave de la industria de software, tales como el Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI, ofreciendo un completo conjunto de herramientas de los equipos de desarrollo de software necesarias para la captura de requisitos , la planificación de programas , la planificación de controles , la clase de modelado y modelado de datos, apoyando el ciclo de vida completo del software, análisis y diseño orientado a objeto, construcción, prueba y despliegue (32). Integra, en un paquete completo, una gran cantidad de funcionalidades (herramientas) en torno al proceso de desarrollo de software, las cuales se especifican a continuación:

[2] Un entorno de desarrollo integrado (en inglés integrated development environment) es un programa informático compuesto por un conjunto de herramientas de programación.

- 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 se puede utilizar VP Teamwork Server, CVS and Subversion.
- Herramienta generación e ingeniería inversa en Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python. Además, apoya la generación de código C #, VB. NET, Lenguaje de Definición de Objetos (ODL), Flash ActionScript, Delphi, Perl, Objective-C, y Ruby. Ingeniería inversa también apoya clase Java class .NET dll y exe, JDBC.
- Herramienta para generar documentación automáticamente a partir de los diagramas en diferentes formatos PDF, HTML y Microsoft Word.
- Herramienta de Integración con varios IDE (Eclipse, NetBeans y 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.

La interoperabilidad constituye otra de las características más importantes del VP-EE, soporta importación y exportación de XMI de versiones 1.0, 1.2 y 2.1. Rational Rose archivos de proyecto (.MDL / .CAT) también pueden ser importados a través de la Rose Importer. Para aprovechar al máximo la interoperabilidad de productos de VP EE con otras aplicaciones, se han introducido a la importación-exportación de modelado de proyectos desde y 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 (16).

2.5.4 Selección de la herramienta CASE para el modelado

En la siguiente tabla se establece una comparación entre las diferentes herramientas CASE abordadas anteriormente.

Aspectos	VP-EE	Rational Rose	EA
Modelado UML	Si.UML 2.1	Si. UML 1.0	Si. UML 2.1
Metodología de Desarrollo	Independiente de la Metodología	Orientado a RUP	Independiente de la Metodología
Modelado de Base de Datos	Si	Si	Si
Integración con el IDE	Alto	Bajo	Alto
Trabajo en Equipo	Si	Si	Si
Generación de Código	Alto	Bajo	Alto
Generación de Documentación	Si	Si	Si
Licencia	Comercial (Privada). Versión Community (Gratis).	Comercial (Privada)	Comercial (Privada). Versión de Prueba por 30 días.
Soporte	Multiplataforma	Multiplataforma	Multiplataforma
Conocimiento de la Herramienta	Alto	Medio	Ninguno

Después de haber analizado las herramientas CASE más usadas a nivel internacional y que brindan un sin número de funcionalidades y utilizando la tabla anterior como base comparativa entre las mismas, se decide la utilización de Visual Paradigm Enterprise Edition para la modelación de los procesos previstos

para el análisis del tercer ciclo de desarrollo del Proyecto SIPP. Fundamentando un poco más el por qué de su selección, podemos mencionar las siguientes características:

1. Es totalmente Independiente de la metodología que se utiliza, lo que ayudaría en tiempo y recursos al Proyecto en caso de la necesidad de hacer una migración de metodología.
2. Soportar UML 2.0, con su amplia gama de Diagramas.
3. Genera código directo e inverso en PHP 5.0, lenguaje de programación utilizado actualmente para la implementación del Proyecto.
4. Permite el Modelado de Base de Datos, generando automáticamente el modelo entidad relación y el modelo objeto relacional, a partir de un modelo de clases persistentes.

2.6 Técnicas para la captura de requisitos

Cuando se desarrolla un Proyecto de Software los analistas utilizan ciertos métodos a fin de recopilar los datos sobre una situación existente, como son las entrevistas, los cuestionarios, la inspección de registros y la observación. Cada uno de ellos tiene sus ventajas y desventajas en dependencia del negocio en que se desarrolle el proyecto. Generalmente, se utilizan dos o tres para complementar el trabajo de cada una y ayudar a asegurar una investigación completa. A continuación se verán describirán cada una de ellas.

2.6.1 Tormenta o lluvia de ideas

Es una técnica muy útil que sirve de guía cuando se ha elegido un tema o una idea, pero no se está seguro de qué se va a escribir sobre el mismo o el conocimiento sobre ese tema es pobre. El objetivo que se persigue al realizar una tormenta de ideas es ***"hacer un listado"*** de todo lo que se le pueda ocurrir al equipo de trabajo y que esté relacionado con el tema elegido. Como técnica de grupo, se hace imprescindible la participación espontánea de todos, generando muchas ideas y también soluciones a algún problema determinado.

2.6.2 Entrevista

Las entrevistas son usadas para recabar información en forma verbal, a través de preguntas que propone el analista. Quienes responden a las mismas pueden ser gerentes o empleados, los cuales son usuarios

actuales del sistema; existiendo también usuarios potenciales del sistema propuesto, que son aquellos que proporcionaran datos o serán afectados por la aplicación. Pudiéndose entrevistar al personal en forma individual o en grupos.

Las entrevistas se pueden clasificar en **estructuradas** las cuales utilizan preguntas estandarizadas. El formato de respuestas para las preguntas puede ser abierto o cerrado y las **entrevistas no estructuradas** que requieren menos tiempo de preparación, porque no se necesita tener por anticipado las palabras precisas de las preguntas. Sin embargo, analizar las respuestas después de las entrevistas lleva más tiempo que con las entrevistas estructuradas.

2.6.3 Cuestionario

Los cuestionarios proporcionan una alternativa muy útil para las entrevistas; sin embargo, existen ciertas características que pueden ser apropiadas en algunas situaciones e inapropiadas en otras. Las preguntas estandarizadas pueden proporcionar datos más confiables. Por otra parte, la característica anterior también es desventaja de los cuestionarios. Aunque su aplicación puede realizarse con un mayor número de individuos, es muy rara una respuesta total. Necesitándose algún seguimiento de los cuestionarios para motivar al personal a responder.

El desarrollo y distribución de los cuestionarios es caro; por lo tanto, el tiempo invertido en los mismos debe utilizarse de una forma inteligente. También es importante el formato y contenido de las preguntas en la recopilación de hechos significativos.

2.6.4 Observación

Observar las operaciones le proporciona al analista hechos que no podría obtener de otra forma. La observación proporciona información de primera mano en relación con la forma en que se llevan a cabo las actividades. Las preguntas sobre el uso de documentos, la manera en la que se realizan las tareas, pueden contestarse rápidamente si se observan las operaciones. La observación es muy útil cuando el analista necesita ver de primera mano cómo se manejan los documentos, como se llevan a cabo los procesos y si ocurren los pasos especificados.

2.6.5 Inspección de registros

Los registros[3] permiten que los analistas se familiaricen con algunas operaciones, oficinas de la compañía y relaciones formales a las que debe darse apoyo. No obstante, no muestran como se producen los hecho y las actividades, donde se ubica el poder verdadero para las decisiones, o como se realizan las tareas en la actualidad. Los otros métodos con objeto de encontrar datos estudiados en esta sección son más eficaces para proporcionar al analista este tipo de información.

2.6.6 Encuesta

La encuesta es una de las técnicas de investigación social más difundidas y se basa en las declaraciones orales o escritas de una muestra[4] de la población con el objeto de recabar información. Se puede basar en aspectos objetivos (hechos, hábitos de conducta, características personales) o subjetivos (opiniones o actitudes).

Las encuestas se pueden clasificar en:

- Encuesta **personal**: donde se entrevista por separado a cada uno de los individuos que constituyen la muestra.
- Encuesta **telefónica**: el contacto entrevistador-entrevistado se realiza a través del teléfono.
- Encuesta **auto-administrada**: el propio encuestado lee el cuestionario y anota las respuestas. Puede no estar acompañado del entrevistador (encuesta por correo u otra vía)

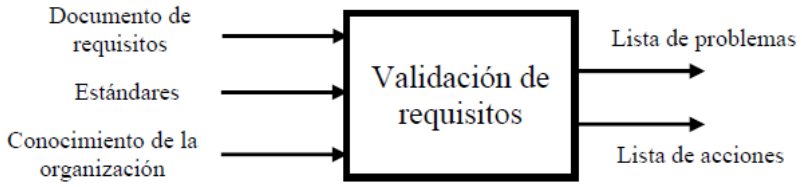
2.7 Técnicas para la validación de requisitos

La validación de los requisitos consiste en demostrar que las características funcionales realmente definen el sistema que el cliente desea, y constituye una tarea fundamental en el desarrollo de un software ya que los errores en el documento de requisitos puede conducir a importantes costos, cuando son descubiertos durante el desarrollo o después de que el sistema esté es uso. El costo implicado en arreglar un problema

[3] Manuales escritos sobre políticas, regulaciones y procedimientos de operaciones estándar que la mayoría de las empresas mantienen como guía para gerentes y empleados.

[4] Subconjunto de casos o individuos de una población estadística.

en los requisitos, haciendo un cambio en el sistema, por lo general es mucho mayor que reparar los errores del diseño o de codificación. La razón de esto es que un cambio en los requisitos normalmente significa un cambio en el diseño y la implementación y por ende se deben probar nuevamente el sistema. La actividad de validación tiene como entrada el documento de requisitos, los estándares relacionados y el conocimiento de la organización, y como salida se obtiene una lista de problemas y una lista de acciones recomendadas, tal y como muestra en la siguiente figura.



La validación se realiza a través de diversos métodos, los más utilizados son:

- ***Revisión de requisitos:*** consisten en reuniones donde el equipo de analistas intenta localizar errores en el documento de especificación.
- ***Prototipado:*** consiste en construir una maqueta del fututo sistema software a partir de los requisitos recogidos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud.
- ***Casos de uso:*** es una técnica para documentar posibles requisitos, graficando la relación del sistema con los usuarios u otros sistemas.
- ***Generación de casos de prueba (test de requisitos):*** este método tiene como objetivo comprobar la verificabilidad de los requisitos. Consiste en la definición de casos de prueba que permitan verificar el cumplimiento de los requisitos funcionales.

A continuación se abordará en los tres primeros métodos, los cuales serán aplicados a los requisitos propuestos para este ciclo de desarrollo.

2.7.1 Revisión de requisitos

La revisión de requisitos es uno de los mejores métodos de validación de requisitos. Consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de

calidad deseados. Estas reuniones son llevadas a cabo por el analista encargado del proyecto y un conjunto de colegas[5] que, preferiblemente, no están relacionados con el proyecto y, además, son competentes en la actividad de requisitos. El resultado final de las reuniones de revisión es un documento que contiene la lista de defectos localizados y una lista de acciones recomendadas. Generalmente, las revisiones de requisitos permiten: descubrir una gran cantidad de defectos en los requisitos, reducir los costos de desarrollo entre un 25 y 30% y reducir el tiempo de pruebas entre un 50 y 90%. Y se realizan habitualmente siguiendo un procedimiento compuesto por los siguientes pasos: **Preparar el plan de la revisión, Distribuir los documentos a revisar, Preparar la reunión, Realizar la reunión de revisión, Identificar los defectos y acciones a realizar, Realizar las correcciones que sean precisas a los documentos revisados, e Informar de las modificaciones realizadas a los participantes en la reunión.**

2.7.2 Prototipos

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas, y en todos los casos, los principios subyacentes son los mismos: el prototipado consiste en la creación de una maqueta o versión del producto final. Los objetivos de los prototipos varían en función de la disciplina. En el caso de la actividad de requisitos, los prototipos se utilizan, fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos.

Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. En ingeniería de requisitos, los prototipos más comunes son los siguientes:

- **Mock-ups**. Se trata de pantallas, típicamente dibujadas a mano en papel, que representan un aspecto concreto del sistema. El soporte que proporcionan a la validación es muy limitado, con la excepción, quizás, de aclarar el interfaz gráfico deseado en casos complejos.
- **Storyboards.** Son una evolución de los mock-ups, ya que además de la interfaz, se muestra la secuencia de acciones, o escenarios, que se deben realizar con el programa.

[5] Los cuales pueden ser voluntarios, seleccionados por el analista o asignados a la revisión por cualesquiera otros medios.

- **Maquetas.** Una maqueta es una versión simplificada del sistema software deseado. Típicamente, una maqueta representa únicamente la interfaz del sistema y, opcionalmente, las conexiones entre pantallas mediante la utilización de elementos activos como los botones. Si fuera necesaria mayor fidelidad, podrían codificarse partes del sistema, de tal modo que además, del interfaz, el software pudiera ofrecer algunos resultados reales. Ello es lo que se conoce como “prototipo funcional”.

2.7.3 Casos de Uso

Los casos de uso son una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de palabras técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. Esta técnica permite describir la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor, además es una descripción de un conjunto de escenarios donde cada uno de ellos comienza con un evento inicial desde un actor hacia el sistema. La mayoría de los requerimientos funcionales se pueden expresar con casos de uso, basándose en escenarios para la obtención de requisito.

2.8 Conclusiones Parciales

Después de haber visto las características fundamentales de algunas de las metodologías, las herramientas CASE, el lenguaje de modelado UML, así como las técnicas para la validación y captura de requisitos abordadas en este capítulo y realizando un amplio estudio de las mismas, se podrá comprender el por qué de la selección de las tecnologías utilizada para el Análisis de la tercera iteración del proyecto SIPP, brindándose también para otros desarrolladores o interesados en el tema, información acerca de la selección de las herramientas más propicias para el modelado a la hora de construir un determinado sistema, teniendo en cuenta las características del mismo.

Capítulo 3: Presentación de la Solución Propuesta.

3.1 Introducción

En este capítulo se hace una descripción del dominio de cada uno de los módulos en cuestión para llegar al modelo general del mismo. Se describe cada una de las clases del dominio, se plantean los requisitos funcionales y no funcionales del sistema, describiendo y prototipando los casos de uso que responden a cada requisito funcional, para finalmente conformar los diagramas de análisis y colaboración. Constituyendo todo este conjunto de artefactos, la solución propuesta para esta investigación.

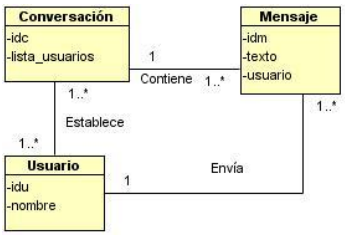
3.2 Modelos de Dominios

El Modelo de Dominio es una representación visual estática del entorno real objeto del proyecto. Es decir, es un diagrama con los objetos reales que existen relacionados con el proyecto a desarrollar y las relaciones que hay entre ellos, bajo ningún concepto son clases de software, aunque algunos objetos del Modelo pueden terminar siéndolo. Ayuda a comprender los conceptos con los que trabajan y que utilizan los usuarios y con los cuales deberá trabajar la aplicación resultante del análisis de estos modelos.

El Modelo de Negocio de RUP incluye toda la organización, sus relaciones y sus procesos. Sin embargo, el Modelo de Dominio se centra en una parte del negocio, la relacionada con el ámbito del proyecto, de ahí que en este contexto el término "Dominio" representa una parte del "Negocio". El modelo de dominio es considerado estático porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "parada" de las clases y sus interacciones.

3.2.1 Diagrama de Clases del Dominio por Módulos

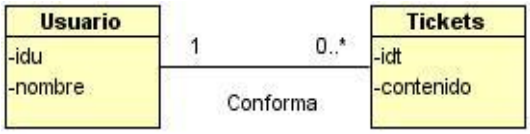
3.2.1.1 Módulo de Comunicación



Descripción del Modelo

El usuario necesita establecer una conversación para consultar cualquier información en tiempo real, relacionada con el proceso de perforación, logrando la misma mediante el envío de sus mensajes y la recepción de los de los otros usuarios vinculados a la conversación.

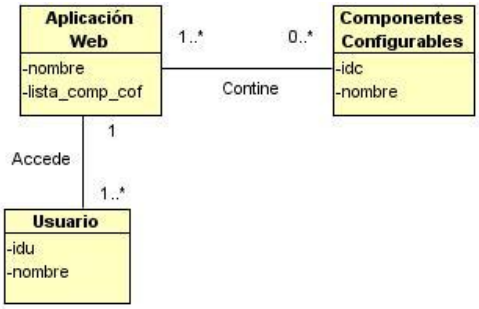
3.2.1.2 Módulo de Historial de Usuario



Descripción del Modelo

El usuario realiza diversas operaciones que les son permisibles acorde al rol que desempeña en el mismo. A medida que se relaciona con la aplicación le surgen ideas para un mejor funcionamiento del sistema, estas ideas (tickets) son enviadas por el usuario y aprobadas o rechazadas por el administrador del sistema, tratando siempre de no inferir en el desempeño satisfactorio del Negocio.

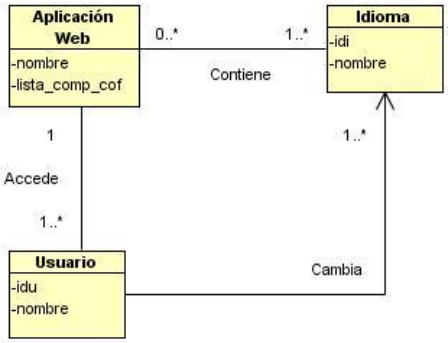
3.2.1.3 Módulo de Configuración de la Aplicación



Descripción del Modelo

La aplicación web posee diferentes componentes configurables (paquetes del menú principal), estos paquetes se le muestran al usuario en dependencia del "rol" que le sea asignado, el mismo puede configurarlos quedando su sesión de trabajo, personalizada.

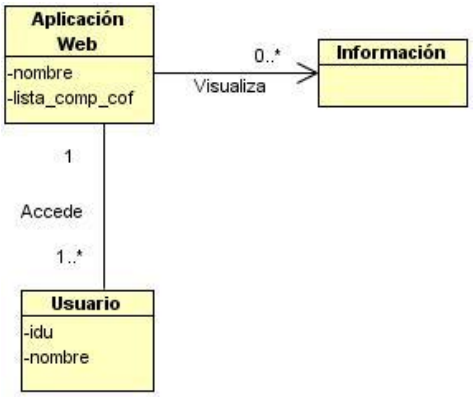
3.2.1.4 Módulo de Internacionalización



Descripción del Modelo

Los usuarios que interactúan con la aplicación web pueden dominar diferentes idiomas, sintiéndose más cómodo en uno específico, por lo que la misma le debe brinda la posibilidad de cambiar el idioma al usuario.

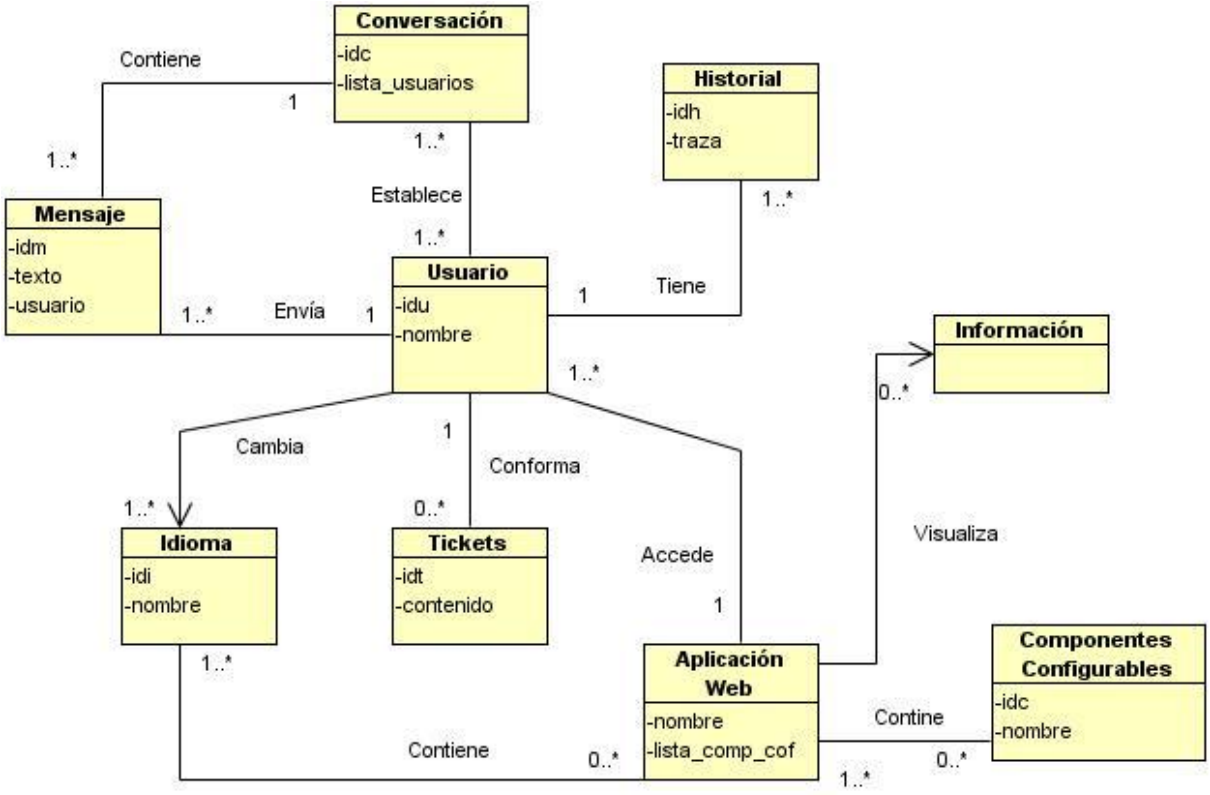
3.2.1.5 Módulo de Visualización Dinámica



Descripción del Modelo

La aplicación web de gestión, visualiza constantemente información para el usuario, ésta información es cargada de la base de datos cada vez que se realice una petición. El usuario actualiza algunos valores y debe presionar algún “botón” que es el que envía la solicitud; cuando la aplicación lo puede hacer automáticamente.

3.2.2 Diagrama de Clases del Dominio General



Un **Usuario** es aquella persona que esté registrada con al menos uno de los roles permitidos por la aplicación.

El **Mensaje** es la cadena de caracteres que constituye la información, que el emisor envía al receptor a través de un canal determinado o medio de comunicación.

Una **Conversación** en la web es el diálogo entre dos o más personas estableciéndose una comunicación a través de una sala de chat.

Una **Ticket** se define como boleto o entrada. Es una especie de entrada (texto) que realiza el usuario en una aplicación, recomendando al administrador de la misma, posibles cambios o sugerencias.

Una **Aplicación Web** es aquella aplicación informática que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador.

Los **Componentes Configurables** de una aplicación web son aquellos que le brindan la posibilidad al usuario de transformarla, de una forma sencilla y agradable, para él.

El **Idioma** no es más que el sistema de comunicación (escrito), que utiliza un software para presentarles la información a los usuarios.

3.3 Requisitos del software

Para el modelado del sistema, se identifican sus requisitos funcionales y no funcionales, modelándose los funcionales en términos de casos de uso del sistema y especificando las condiciones o capacidades que el sistema debe cumplir, teniendo en cuenta que todas las ideas que los stakeholders, clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidato a requisito de software.

3.3.1 Requisitos Funcionales

RF 1 El sistema debe permitir establecer una conversación.

RF 1.1 El sistema debe permitir enviar un mensaje público.

RF1.2 El sistema debe permitir enviar un mensaje privado.

RF 2 El sistema debe permitir limpiar el chat.

RF 3 El sistema debe permitir personalizar el chat.

RF 4 El sistema debe permitir añadir emoticonos.

RF 5 El sistema debe permitir enviar reporte por correo.

RF 6 Gestionar tickets.

RF 6.1 El sistema debe permitir mostrar los tickets de los usuarios.

RF 6.2 El sistema debe permitir eliminar los tickets de los usuarios.

RF 7 El sistema debe permitir exportar tickets.

RF 8 El sistema debe permitir conformar un ticket.

RF 9 Gestionar permisos de configuración.

RF 9.1 El sistema debe permitir asignarle nuevos permisos de configuración al usuario.

FR 9.2 El sistema debe permitir modificar los permisos de configuración del usuario.

RF 9.3 El sistema debe permitir mostrar los permisos de configuración del usuario.

RF 9.4 El sistema debe permitir eliminar los permisos de configuración de un usuario.

RF 10 El sistema debe permitir configurar el espacio de trabajo.

RF 11 Gestionar vistas de reportes.

RF 11.1 El sistema debe permitir crear una vista.

RF 11.2 El sistema debe permitir editar una vista.

RF 11.3 El sistema debe permitir guardar una vista.

RF 11.4 El sistema debe permitir eliminar una vista.

RF 12 El sistema debe permitir cambiar el idioma

RF 13 Gestionar Fórum

RF 13.1 El sistema debe permitir crear un mensaje en fórum.

RF 13.2 El sistema debe permitir eliminar un mensaje del fórum.

3.3.2 Requisitos No funcionales

Teniendo en cuenta que los requisitos no funcionales son las propiedades o cualidades que el sistema debe poseer, dígase estas como las propiedades o cualidades que hacen al producto atractivo, usable, rápido, confiable, entre otras… y contando con el análisis del Primer y Segundo Ciclo de Desarrollo del Proyecto, en los cuales se recogen las cualidades que el sistema debe poseer. Se decide incorporar otras, así como refinar las características no funcionales que lo requieran especificadas en estos ciclos anteriores. A continuación se reflejan las mismas:

Apariencia o interfaz externa

RNF 1 El sistema tendrá un menú para el idioma y la configuración, visibles todo el tiempo para acceder de forma directa y rápida por el usuario.

RNF 2 El sistema tendrá en su ambiente una combinación de colores agradable y contará con una estructura sencilla y amigable, permitiendo que el usuario se adapte con facilidad.

Usabilidad

RNF 3 La aplicación debe ser de fácil comprensión, navegación, configuración y utilización para usuarios de nivel alto, medio o bajo de experiencia en el campo de la informática.

RNF 4 La información debe ser mostrada de forma lógica y organizada para el usuario.

Ayuda

RNF 5 El sistema debe contar con un manual de usuario donde se le incorporen las funcionalidades de este ciclo de desarrollo.

RNF 6 El sistema debe contar con una opción de Ayuda, garantizando que esté visible todo el tiempo para el usuario.

RNF 7 El sistema incorporará a los mensajes de “alerta” y “error”, recomendaciones que le sirvan de ayuda al usuario durante las operaciones que lo requieran, permitiendo la orientación del usuario en cada procedimiento.

Hardware

RNF 8 Se debe disponer de dos servidores, uno para el Correo Electrónico y otro para el Chat, aunque en un principio se pueden montar ambos en la misma PC.

3.4 Actores del sistema

Los actores del sistema se pueden definir como terceros que interactúan con el software (sistema) sin ser parte del mismo, pudiendo ser, según UML, un rol desempeñado por un usuario o cualquier otro sistema que interactúa con el sujeto. Los actores del sistema no forman parte del mismo, pueden intercambiar información con éste, pueden representar un rol que juegan una o varias personas, un equipo o un sistema automatizado y pueden ser un recipiente pasivo de información.

Teniendo en cuenta las funcionalidades identificadas para este ciclo de desarrollo, las cuales son el resultado de la profundización del negocio del primer y segundo ciclo de desarrollo del proyecto SIPP no se identificaron nuevos “Actores del Sistema”, reflejándose a continuación los ya existentes.

Descripción de los actores del sistema	
Actores	**Descripción**
Supervisor del Pozo.	Este actor proviene del trabajador del negocio del mismo nombre. Interactúa principalmente con los casos de Uso del Módulo Supervisor. Este Módulo pertenece al Subsistema Pozo.
Supervisor Jefe del Pozo.	Este actor aglutina las funciones de los actores Supervisor del Pozo y Administrador del Subsistema Pozo.
Secretaria de Despacho.	Este actor proviene del trabajador del negocio del mismo nombre. Interactúa principalmente con los casos de Uso del Módulo Despacho de la DIPP. Este Módulo pertenece al Subsistema DIPP.
Geólogo de Pozo.	Este actor proviene del trabajador del negocio del mismo nombre. Interactúa principalmente con los casos de Uso del Módulo Geología. Este Módulo pertenece al Subsistema Pozo.
Geólogo del CEINPET.	Este actor proviene del trabajador del negocio del mismo nombre. Interactúa principalmente con los casos de Uso del Subsistema CEINPET.
Directivo.	Este actor representa a los directivos de la DIPP, consultando toda la información referente a los pozos así como Iniciar la perforación de un Pozo.
Administrador del Sistema	Este actor es el encargado de administrar el sistema. Interactúa principalmente con los casos de Uso del Módulo Administración. Este Módulo

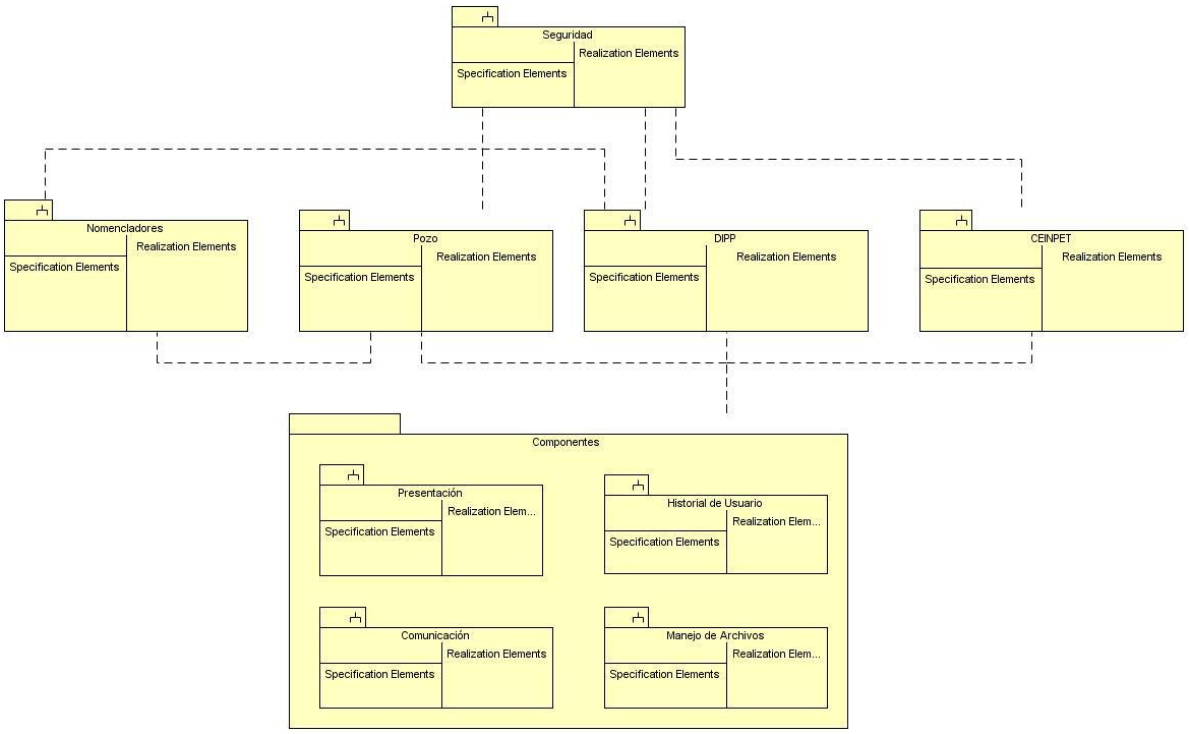
	pertenece al Subsistema Pozo.
Técnico en Perforación	Este actor es el encargado de dar inicio a los pozos (Gestionar Pozo), cambiarle el estado, así como consultar información de los pozos en perforación.
Administrador de Nomencladores	Este actor es el encargado de administrar los nomencladores de perforación que utilizaran en cada uno de los pozos.
Administrador del Subsistema Pozo	Este actor es el encargado de administrar el sistema en el pozo.
Usuario registrado	Este actor representa las interacciones genéricas de los actores que interactúan con los mismos casos de usos.

3.5 Modelo de Sistema

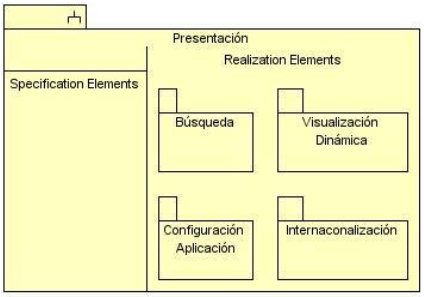
UML proporciona una amplia gama de estereotipos para conformar los modelos de sistema, destacándose los estereotipos de Subsistemas y Paquetes. La agrupación de los casos de uso del sistema utilizando estos estereotipos se considera efectiva ya que contribuye a un mejor entendimiento del funcionamiento del sistema propuesto, además de obtener una mejor trazabilidad de los casos de uso a través de los flujos de trabajo.

En la siguiente imagen se muestra el Modelo de Sistema al cual se le adicionaron los subsistemas de Comunicación e Historial de Usuario, además de incluir los paquetes de Visualización Dinámica, Configuración de la Aplicación e Internacionalización en el subsistema de Presentación.

Modelo de Sistema



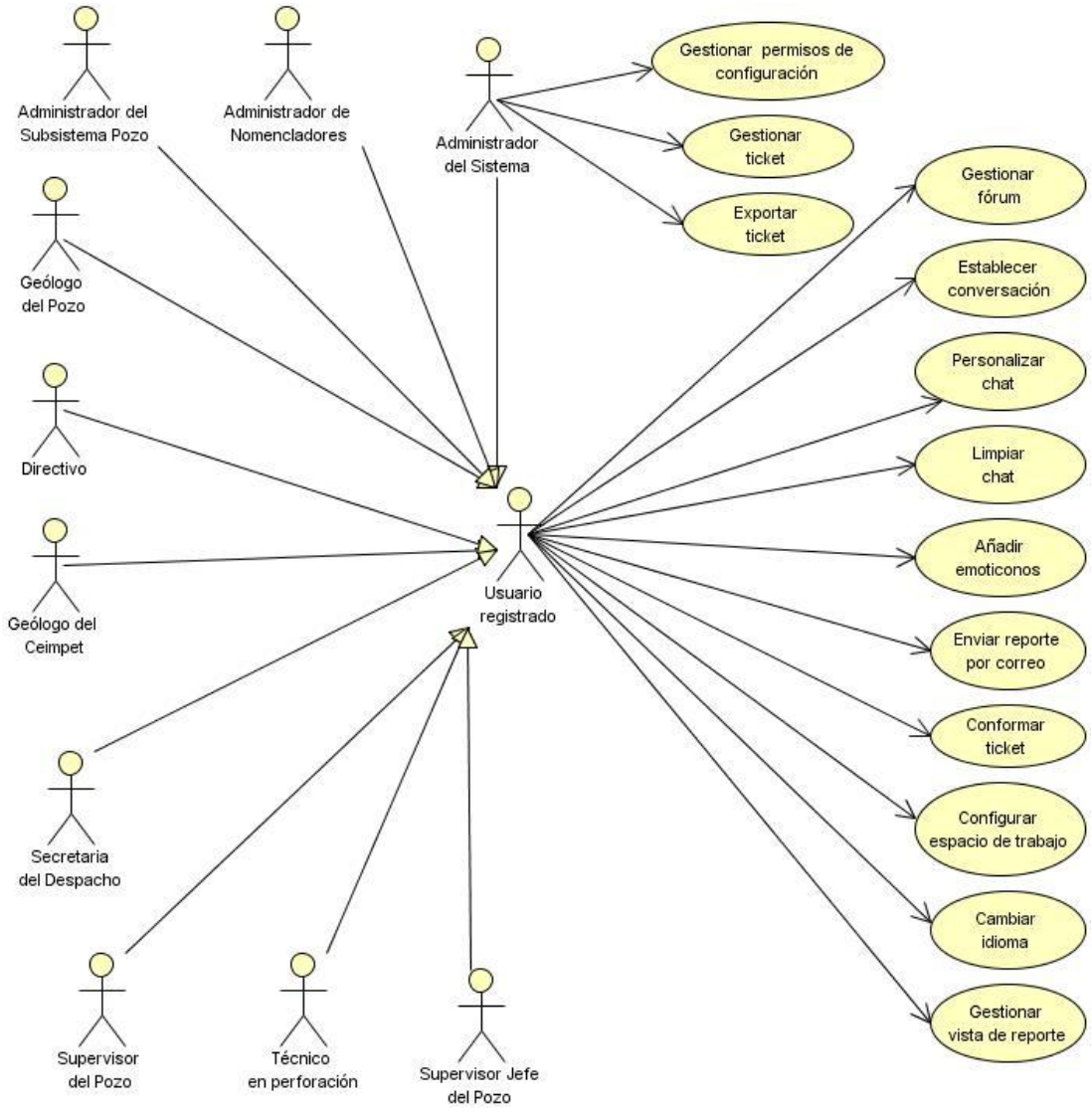
Subsistema de Presentación



3.6 Diagrama de casos de uso del sistema

Los diagramas de Casos de Uso del sistema son ubicados dentro de los diagramas de comportamiento de UML y constituyen una representación gráfica de los procesos y su interacción con los actores, se utilizan para ilustrar los requisitos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

En la siguiente figura se representa el diagrama de casos de uso del sistema propuesto para este ciclo de desarrollo, plasmándose todos los casos de usos necesarios para satisfacer los requisitos funcionales identificados.



3.6.1 Casos de uso del sistema

Un Casos de Uso del sistema es un artefacto que describe las acciones, reacciones y el comportamiento del sistema desde el punto de vista del usuario, estableciendo un acuerdo entre el cliente y los

desarrolladores sobre las condiciones y posibilidades que debe cumplir el sistema, respondiendo siempre a los requisitos funcionales.

Casos de Uso del Sistema			
Críticos	**Secundarios**	**Auxiliares**	**Opcionales**
Gestionar permisos de configuración	Enviar reporte por correo		Personalizar chat
Configurar espacio de trabajo	Gestionar ticket		Añadir emoticonos
Establecer conversación	Conformar ticket		Limpiar chat
Cambiar idioma	Exportar ticket		
Gestionar vistas de reporte	Gestionar Fórum		

3.7 Proceso de mejora

La Universidad de Ciencias Informáticas (UCI) está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI[6] y con la contratación de los servicios de consultoría del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey, estos servicios permiten:

- Ayudar a revisar la estrategia de mejora de procesos de software, para asegurar que su organización está basada en procesos y con un programa de mejora continua alineado con sus objetivos de negocio.
- Ayudar a establecer las bases y fundamentos para seguir mejorando sus procesos y fortalecer su cultura de calidad en el desarrollo de software
- Alinear los procesos de desarrollo de software con los principios y requisitos del modelo CMMI, estableciendo planes de mejora con los que la organización oriente sus procesos hacia la consecución de sus metas.

[6] CMMI (Capability Madurity Model Integration), es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software.

El proceso de mejora tiene definidos los roles y documentos para el nivel 2 de madurez de CMMI, de esta manera, los nuevos proyectos, módulos y componentes se deben documentar por los mismos. En esta investigación se utilizará la documentación establecida por este proceso para la descripción de los Casos de Uso del Sistema del Epígrafe siguiente.

3.8 Especificación de Casos de Uso

En este epígrafe se reflejarán las descripciones de los casos de uso clasificados como críticos para este ciclo de desarrollo, quedando constancia en toda su totalidad en el documento "***Especificación de casos de uso.pdf***".

3.8.1 Caso de Uso: Gestionar permiso de configuración

3.8.1.1 Descripción

Objetivo	Permitir al Actor asignar, modificar, mostrar y eliminar los permisos de configuración.	
Actores	Administrador del Sistema	
Resumen	El caso de uso se inicia cuando el Actor desea asignar, modificar, mostrar o eliminar permisos de configuración de la aplicación, de un usuario determinado.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	Debe estar el Actor registrado en el sistema.	
Poscondiciones	Se gestionan los permisos de configuración del usuario deseado.	
Flujo de eventos		
Flujo básico "Gestionar permiso de configuración"		
	Actor	**Sistema**
1.	Selecciona la opción "Configuración de Permisos".	Despliega un menú para gestionar los permisos de configuración.
2.	Selecciona una de las opciones definidas para gestionar los permisos de configuración.	Ejecuta una de las sesiones definidas para el caso de uso: 1.Asignar permisos

		2. Listar/Eliminar 3. Modificar
Flujos alternos		
Nº Evento Presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón “ ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Aceptar”.	Re-direcciona al menú principal.
3.		Termina el caso de uso.
Flujos alternos		
Nº Evento presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón “ ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
3.		Termina el caso de uso.
Sesión 1: “Asignar permisos”		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en la ruta Configuración de Permisos – Asignar.	Muestra el formulario “Asignar Permisos”, con una lista desplegable para seleccionar el usuario y los permisos que se le pueden asignar.
2.	Selecciona el usuario.	
3.	Marca los permisos que le serán asignados el usuario seleccionado.	
4.	Presiona clic en el botón “Asignar”.	Envía un mensaje de confirmación de la operación “¿Está seguro que desea asignar los permisos al usuario seleccionado?” y los botones “Aceptar” y

		"Cancelar".
5.	Presiona clic en el botón "Aceptar"	Envía un mensaje del éxito de la operación "Los permisos has sido asignados correctamente".
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento Campos vacíos		
	Actor	**Sistema**
4.	Presiona clic en el botón "Asignar" sin seleccionar el usuario o el(los) permisos.	Muestra un mensaje de error "Faltan campos por completar" y el botón "Aceptar".
5.	Presiona clic en el botón "Aceptar".	Permanece en la misma interfaz.
6.		Termina el caso de uso
Flujos alternos		
Nº Evento No confirma la operación		
	Actor	**Sistema**
5.	Presiona clic en el botón "Cancelar".	Permanece en la misma interfaz.
6.		Termina el caso de uso.
Sesión 2: "Listar/Eliminar"		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en la ruta Configuración de Permisos – Listar/Eliminar.	Muestra el formulario "Listar/Eliminar", con una lista desplegable para seleccionar el permiso.
2.	Selecciona el permiso.	Muestra el formulario "Lista de usuarios" con el listado de los usuarios que se les asignó el permiso seleccionado.
3.	Marca el usuario, del cual desea eliminar el permiso seleccionado anteriormente.	
4.	Presiona clic en el botón "".	Envía un mensaje de confirmación de la operación "¿Está seguro que desea eliminarle el permiso de configuración al usuario seleccionado?" y los

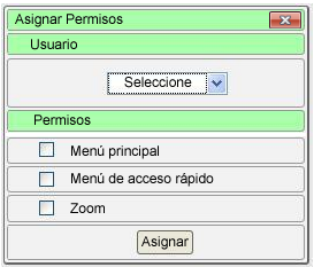
		botones “Aceptar” y “Cancelar”.
5.	Presiona clic en el botón “Aceptar”	Envía un mensaje del éxito de la operación “El permisos has sido eliminado correctamente”.
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento Marca el usuario, del cual desea eliminar el permiso seleccionado anteriormente.		
	Actor	**Sistema**
3.	No marca el usuario, del cual desea eliminar el permiso seleccionado anteriormente.	
4.	Presiona clic en el botón “”.	Muestra un mensaje de error “Primero debe seleccionar al usuario”.
6.		Termina el caso de uso
Flujos alternos		
Nº Evento No confirma la operación		
	Actor	**Sistema**
5.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
6.		Termina el caso de uso.
Sesión 3: “Modificar”		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en la ruta Configuración de Permisos – Modificar.	Muestra el formulario “Seleccionar usuario” con una lista desplegable para seleccionar el usuario.
2.	Selecciona el usuario.	Muestra el formulario “Permisos” con los permisos que tiene asignado el usuario seleccionado.
3.	Actualiza los permisos del usuario seleccionado.	
4.	Presiona clic en el botón “Modificar”.	Envía un mensaje de confirmación de la operación “¿Está seguro que desea actualizar los permiso de configuración del usuario seleccionado?” y los

		botones “Aceptar” y “Cancelar”.
5.	Presiona clic en el botón “Aceptar”	Envía un mensaje del éxito de la operación “Los permisos has sido actualizados correctamente”.
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento No confirma la operación		
	Actor	**Sistema**
5.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
6.		Termina el caso de uso.
Relaciones		
Requisitos funcionales	RF 9	
Requisitos no funcionales	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6, RNF 7	
Asuntos pendientes		

3.8.1.2 Prototipo elemental de interfaz gráfica de usuario

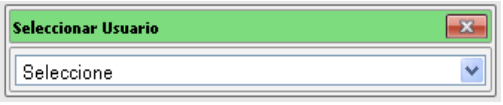
Flujo básico: Gestionar permiso de configuración.

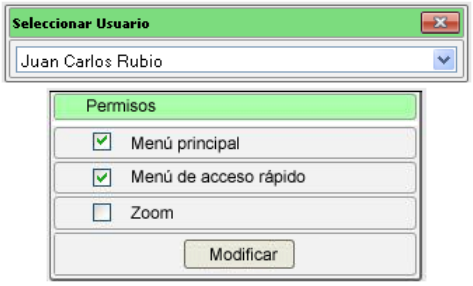
Sesión 1: Asignar



Sesión 2: Listar/Eliminar

Sesión 3: Modificar





3.8.2 Caso de Uso: Configurar espacio de trabajo

3.8.2.1 Descripción

Objetivo	Permitir al Actor configurar su espacio de trabajo en la aplicación, en correspondencia con los permisos de configuración que le han sido asignados
Actores	Usuario registrado
Resumen	El caso de uso se inicia cuando el Actor desea configurar su espacio de trabajo en la aplicación, siempre en correspondencia con los permisos de configuración que le han sido otorgados.
Complejidad	Media
Prioridad	Alta
Precondiciones	Debe estar los Actores registrado en el sistema. Deben tener asignado al menos un permiso de configuración.
Poscondiciones	Se configura la aplicación, de una forma sencilla, en la sesión de trabajo del usuario registrado.
Flujo de eventos	
Flujo básico "Configurar espacio de trabajo"	

	Actor	**Sistema**
1.	Selecciona en el menú de acceso rápido "  " la opción de configuración deseada.	Ejecuta una de las sesiones definidas para el caso de uso: 1. Zoom 2. Menú Principal 3. Menú de acceso rápido

2.		Termina el Caso de Uso
Sesión 1: "Zoom"		
Nº Evento Presiona botón de aumentar zoom		
	Actor	**Sistema**
1.	Presiona clic en el botón "  "	Aumenta el tamaño de fuente de la aplicación en un 10%.
2.		Termina el Caso de Uso
Flujos alternos		
Nº Evento Presiona botón de disminuir zoom		
	Actor	**Sistema**
1.	Presiona clic en el botón "  "	Disminuye el tamaño de fuente de la aplicación en un 10%.
2.		Termina el Caso de Uso
Flujos alternos		
Nº Evento Cantidad de clic en los botones supera la cantidad permitida		
	Actor	**Sistema**
	Presiona clic en el botón "  " o en el "  ".	Oculta el botón de aumentar o disminuir zoom, en correspondencia con el límite máximo o mínimo establecido en la aplicación.
2.		Termina el caso de uso.
Sesión 2: "Menú Principal"		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en el botón "  ".	Muestra el formulario "Configuración del Menú Principal" con los campos: Posición Superior (marcado por defecto). Inferior

		Posiciones de los Sub-Menús (generando tantos campos como a sub-menús tenga acceso el usuario registrado).
2.	Selecciona la Posición y los Sub-Menús en el orden deseado.	
3.	Presiona clic en el botón “Configurar”.	Muestra un mensaje de confirmación de la operación “¿Está seguro que desea configurar su sesión de trabajo?” y los botones “Aceptar” y “Cancelar”.
4.	Presiona clic en el botón “Aceptar”.	Configura la sesión de trabajo.
5.		Envía un mensaje del éxito de la operación “Su sesión ya está configurada”.
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento *Selecciona la Posición y los Sub-Menús en el orden deseado.*		
	Actor	**Sistema**
2.	No selecciona los sub-menús en todos los campos.	
3.	Presiona clic en el botón “Configurar”.	Muestra un mensaje de error “Aun hay campos sin seleccionar”.
4.		Permanece en la misma interfaz.
5.		Termina el caso de uso
Flujos alternos		
Nº Evento *Selecciona la Posición y los Sub-Menús en el orden deseado.*		
	Actor	**Sistema**
2.	Selecciona en dos o más campos de Sub-Menús el mismo sub-menú.	
3.	Presiona clic en el botón “Configurar”.	Muestra un mensaje de error “Hay sub-Menús repetidos”.
4.		Permanece en la misma interfaz.

5.		Termina el Caso de Uso.
Flujos alternos		
Nº Evento *No confirma la operación*		
	Actor	**Sistema**
4.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
5.		Termina el Caso de Uso.
Flujos alternos		
Nº Evento Presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón “  ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
3.		Termina el caso de uso.
Flujos alternos		
Nº Evento presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón “  ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Aceptar”.	Re-direcciona al menú principal.
3.		Termina el caso de uso.
Sesión 3: “Menú de Acceso Rápido”		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en el botón “  ”	Muestra el formulario “Configurar Menú de Acceso Rápido” con el campo “Mostrar” y los elementos que se pueden mostrar o no

		opcionalmente.
2.	Selecciona (marcando) los elementos que desea que estén visibles en el Menú de Acceso Rápido, en su sesión de trabajo.	
3.	Presiona clic en el botón “Configurar”.	Muestra un mensaje de confirmación de la operación “¿Está seguro que desea configurar su menú de acceso rápido?” y los botones “Aceptar” y “Cancelar”.
4.	Presiona clic en el botón “Aceptar”.	Configura el menú.
5.		Envía un mensaje del éxito de la operación “Su el menú de acceso rápido ya está configurado”.
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento No confirma la operación		
	Actor	**Sistema**
4.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
5.		Termina el caso de uso.
Flujos alternos		
Nº Evento Presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón “  ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz.
3.		Termina el caso de uso.
Flujos alternos		
Nº Evento presiona el botón de cancelar operación.		
	Actor	**Sistema**

1.	Presiona clic en el botón “ ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Aceptar”.	Re-direcciona al menú principal.
3.		Termina el caso de uso.
Relaciones		

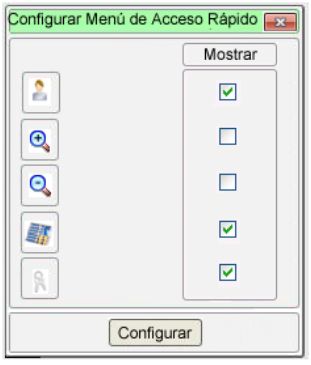
Requisitos funcionales	RF 10
Requisitos no funcionales	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6 RNF 7
Asuntos pendientes	

3.8.2.2 Prototipo elemental de interfaz gráfica de usuario

Flujo básico: Configurar espacio de trabajo.

Sesión 2: Menú Principal

Sesión 3: Menú de Acceso Rápido



3.8.3 Caso de Uso: Cambiar idioma

3.8.3.1 Descripción

Objetivo	Permitirle al Actor cambiar el idioma del menú de la aplicación, mensajes y lenguaje no técnico utilizado en la misma. Estimando para una primera iteración los idiomas (Español e Inglés).
Actores	Usuario registrado
Resumen	El caso de uso se inicia cuando el Actor desea ver el contenido (menús, mensajes, etc.) en otro idioma.
Complejidad	Media
Prioridad	Alta
Precondiciones	Debe estar el Actor registrado en el sistema.
Poscondiciones	La aplicación cambia el idioma.

Flujo de eventos

Flujo básico "Cambiar idioma"

	Actor	Sistema
1.	Realiza la petición al servidor del Sitio Web escribiendo la dirección "*url.*" en el navegador.	Muestra el formulario de registro "Identificación" con las credenciales (usuario y contraseña), el campo de selección "Idioma" con el idioma (Español) por defecto y el botón "Entrar"
2.	Selecciona el idioma deseado.	
3.	Presiona clic en el botón "Entrar".	Cambia el idioma de la aplicación, al idioma seleccionado por el usuario.
4.		Termina el Caso de Uso

Flujos alternos

Nº Evento *Selecciona el idioma deseado.*

	Actor	Sistema
2.	Selecciona el idioma (English [EN]).	Actualiza la interfaz de registro con el idioma Inglés.
3.		Vuelve al Paso 3 del Flujo básico "Cambiar idioma".

Sesión 1: "Cambiar idioma estando registrado"

Nº Evento

	Actor	Sistema
1.	Presiona clic en el idioma deseado de la barra de idiomas “  ” del menú de acceso rápido.	Cambia el idioma de la aplicación, para el idioma seleccionado por el usuario.
2.		Muestra un mensaje del éxito de la operación en el idioma seleccionado. • Español: “Ha cambiado el idioma a *Español*” • Inglés: “Changed the language to *English*”
3.		Termina el Caso de Uso
Relaciones		
Requisitos funcionales	RF 12	
Requisitos no funcionales	RNF 1, RNF 2, RNF 7	
Asuntos pendientes		

3.8.3.2 Prototipo elemental de interfaz gráfica de usuario

Flujo Básico: Cambiar idioma

Flujo Alterno

3.8.4 Caso de Uso: Establecer conversación

3.8.4.1 Descripción

Objetivo	Permitirle al Actor poder establecer una conversación en tiempo real con otros usuarios de la aplicación.

Actores	Usuario registrado
Resumen	El caso de uso se inicia cuando el Actor necesita comunicarse con otro(s) usuario(s) de la aplicación para consultar información en tiempo real.
Complejidad	Media
Prioridad	Alta
Precondiciones	Debe estar el Actor registrado en el sistema.
Poscondiciones	Se establece la conversación.

Flujo de eventos

Flujo básico "Establecer conversación"

	Actor	**Sistema**
1.	Solicita establecer una conversación presionando clic en el botón "  " de la barra de acceso rápido.	Muestra la ventana "Establecer conversación" con un listado de los usuarios conectados, los desconectados, y el botón "Enviar mensaje de difusión".
2.		Ejecuta una de las sesiones definidas para el caso de uso: 1. Mensaje privado. 2. Mensaje público.
3.		Termina el Caso de Uso

Flujos alternos

Nº Evento Presiona el botón de cancelar operación.

	Actor	**Sistema**
1.	Presiona clic en el botón "  ".	Muestra un mensaje de confirmación de la operación "¿Desea cancelar la operación?" y los botones "Aceptar" y "Cancelar".
2.	Presiona clic en el botón "Cancelar".	Permanece en la misma interfaz.
3.		Termina el caso de uso.

Flujos alternos

Nº Evento Presiona el botón de cancelar operación.

	Actor	**Sistema**

1.	Presiona clic en el botón “  ”.	Muestra un mensaje de confirmación de la operación “¿Desea cancelar la operación?” y los botones “Aceptar” y “Cancelar”.
2.	Presiona clic en el botón “Aceptar”.	Re-direcciona al menú principal.
3.		Termina el caso de uso.
Sesión 1: “Mensaje privado”		
Nº Evento		
	Actor	**Sistema**
1.	Pasa con el mouse por encima del nombre del usuario con el cual desea establecer la conversación.	Muestra de color rojo el nombre del usuario.
2.	Presiona clic encima del nombre del usuario.	Muestra la ventana de dialogo “Conversación”.
3.	Introduce el texto deseado.	
4.	Presiona la tecla “Enter”.	Envía información al servidor de Chat.
5.		Obtiene información del servidor chat.
6.		Muestra la información.
7.		Termina el caso de uso.
Flujos alternos		
Nº Evento *Pasa con el mouse por encima del nombre del usuario con el cual desea establecer la conversación*		
	Actor	**Sistema**
1.	Mueve el mouse de encima del nombre del usuario del cual desea establecer la conversación.	Vuelve a mostrar el nombre del usuario del color predefinido”.
2.		Termina el caso de uso.
Sesión 2: “Mensaje público”		
Nº Evento		
	Actor	**Sistema**
1.	Presiona clic en el botón “Enviar mensaje	Muestra un mensaje de confirmación de la

	de difusión".	operación "¿Desea enviar un mensaje a todos los usuarios?" y los botones "Aceptar" y "Cancelar".
2.	Presiona clic en el botón "Aceptar".	Muestra la ventana "Mensaje de difusión" con un campo para introducir el texto del mensaje y el botón "Enviar".
3.	Introduce el texto deseado.	
4.	Presiona la tecla "Enviar".	Envía información al servidor de Chat.
5.		Obtiene información del servidor chat.
6.		Muestra la información.
7.		Termina el caso de uso.

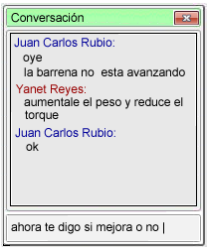
Flujos alternos		
Nº Evento *Presiona clic en el botón "Aceptar".*		
	Actor	**Sistema**
2.	Presiona clic en el botón "Cancelar"	Permanece en la misma interfaz.
3.		Termina el Caso de Uso.

Relaciones	
Requisitos funcionales	RF 1
Requisitos no funcionales	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6 RNF 7, RNF 8
Asuntos pendientes	

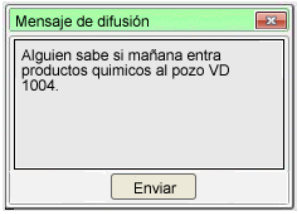
3.8.4.2 Prototipo elemental de interfaz gráfica de usuario

Flujo Básico: Establecer conversación

Sesión 1: "Mensaje privado"



Sesión 2: "Mensaje público"



3.8.5 Caso de Uso: Gestionar vistas de reporte

3.8.5.1 Descripción

Objetivo	Permitirle al Actor poder establecer su propia vista de un reporte determinado para que solo le sea visualizada la información que necesita del mismo.
Actores	Usuario registrado
Resumen	El caso de uso se inicia cuando el Actor necesita crear una nueva vista de un reporte determinado, guardarla, editar una ya creada o eliminarla.
Complejidad	Media
Prioridad	Alta
Precondiciones	Debe estar el Actor registrado en el sistema.
Poscondiciones	Se gestiona la vista de reporte seleccionado.

Flujo de eventos		
Flujo básico "Gestionar vistas de reporte"		
	Actor	**Sistema**
1.	Solicita en el menú principal la opción "Vistas de Reporte."	Despliega un submenú para gestionar las vistas de reportes.
2.		Ejecuta una de las sesiones definidas para el caso de uso: 1. Crear vista 2. Editar vista 3. Eliminar vista
3.		Termina el Caso de Uso
Flujos alternos		
Nº Evento Presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón "  ".	Muestra un mensaje de confirmación de la operación "¿Desea cancelar la operación?" y los botones "Aceptar" y "Cancelar".
2.	Presiona clic en el botón "Cancelar".	Permanece en la misma interfaz.
3.		Termina el caso de uso.
Flujos alternos		
Nº Evento Presiona el botón de cancelar operación.		
	Actor	**Sistema**
1.	Presiona clic en el botón "  ".	Muestra un mensaje de confirmación de la operación "¿Desea cancelar la operación?" y los botones "Aceptar" y "Cancelar".
2.	Presiona clic en el botón "Aceptar".	Re-direcciona al menú principal.
3.		Termina el caso de uso.
Sesión 1: "Crear vista"		
Nº Evento		
	Actor	**Sistema**
1.	Selecciona la opción Vistas de	Muestra el formulario "Crear nueva vista" para

	Reportes – Crear vista.	seleccionar el reporte del cual se desea conformar la misma.
2.	Selecciona el reporte.	Muestra el formulario “Nueva vista” con campos para seleccionar los elementos de reporte que se desean ver en la nueva vista que será creada y el campo “Guardar vista” deshabilitado por defecto.
3.	Marca el campo “Guardar vista”.	Habilita un campo para escribir el nombre de la nueva vista.
4.	Introduce el nombre de la nueva vista.	
5.	Selecciona los elementos que tendrá la nueva vista del reporte.	
6.	Presiona clic en el botón “Crear”	Muestra un mensaje de confirmación de la operación “¿Está seguro que desea crear esta nueva vista del reporte?” y los botones “Aceptar” y “Cancelar”.
7.	Presiona clic en el botón “Aceptar”.	Guarda la vista en la base de datos.
8.		Muestra un mensaje del éxito de la operación “La nueva vista ha sido creada y guardada correctamente”.
9.		Termina el caso de uso.

Flujos alternos

Nº Evento *Marca el campo “Guardar vista”.*

	Actor	**Sistema**
3.	No marca el campo “Guardar vista”.	
4.	Selecciona los elementos que tendrá la nueva vista del reporte.	
5.	Presiona clic en el botón “Crear”	Muestra un mensaje de confirmación de la operación “¿Está seguro que desea crear esta nueva vista del reporte?” y los botones “Aceptar” y “Cancelar”.
6.	Presiona clic en el botón “Aceptar”.	Muestra un mensaje del éxito de la operación “La nueva vista ha sido creada correctamente”.

7.		Termina el caso de uso.
Flujos alternos		
Nº Evento Presiona clic en el botón “Aceptar”.		
	Actor	**Sistema**
6.	Presiona clic en el botón “Cancelar”.	Permanece en la misma interfaz
7.		Termina el caso de uso.
Sesión 2: “Editar vista”		
Nº Evento		
	Actor	**Sistema**
1.	Selecciona la opción Vistas de Reportes – Editar vista.	Muestra el formulario “Editar vista” con los campos de selección “Reporte” y “Nombre de la vista”, donde este último campo es llenado con el nombre de las vista creadas por el usuario.
2.	Selecciona el reporte deseado.	
3.	Selecciona la vista deseada.	Muestra el formulario llamado con el mismo nombre de la vista seleccionada, con el contenido de la misma y el botón “Modificar”.
4.	Actualiza los campos de la vista.	
5.	Presiona clic en el botón “Modificar”.	Muestra un mensaje de confirmación de la operación “¿Está seguro que desea modificar esta vista?” y los botones “Aceptar” y “Cancelar”.
6.	Presiona clic en el botón “Aceptar”	Muestra un mensaje del éxito de la operación “La vista ha sido actualizada correctamente”.
7.		Termina el caso de uso.
Flujos alternos		
Nº Evento *Presiona clic en el botón “Aceptar”.*		
	Actor	**Sistema**
6.	Presiona clic en el botón “Cancelar”	Permanece en la misma interfaz.
7.		Termina el Caso de Uso.

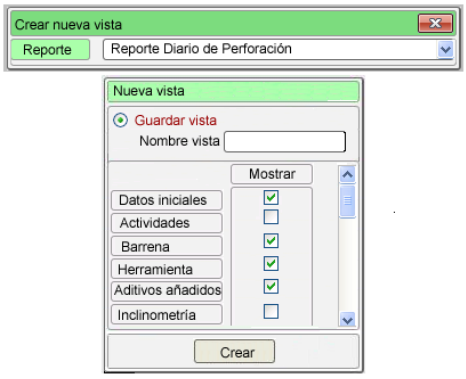
Flujos alternos		
Nº Evento Selecciona el reporte deseado.		
	Actor	**Sistema**
2.	Selecciona el reporte deseado, del cual no se ha elaborado ninguna vista.	Muestra un mensaje de alerta "Del reporte seleccionado aun no se ha creado ninguna vista." Y el botón "Aceptar".
3.	Presiona clic en el botón "Aceptar".	Permanece en la misma interfaz.
4.		Termina el Caso de Uso
Sesión 3: "Eliminar vista"		
Nº Evento		
	Actor	**Sistema**
1.	Selecciona la opción Vistas de Reportes – Eliminar vista.	Muestra el formulario "Listado de vistas" y el campo de selección "Reporte".
2.	Selecciona el reporte deseado.	Muestra el listado de las vistas realizadas por el usuario registrado.
3.	Selecciona la vista deseada.	
4.	Presiona clic en el botón "  ".	Muestra un mensaje de confirmación de la operación "¿Está seguro que desea eliminar la vista seleccionada?" y los botones "Aceptar" y "Cancelar".
5.	Presiona clic en el botón "Aceptar".	Muestra un mensaje del éxito de la operación "La vista se ha eliminado correctamente".
6.		Termina el caso de uso.
Flujos alternos		
Nº Evento *Presiona clic en el botón "Aceptar".*		
	Actor	**Sistema**
5.	Presiona clic en el botón "Cancelar"	Permanece en la misma interfaz.
6.		Termina el Caso de Uso.
Flujos alternos		

Nº Evento Selecciona el reporte deseado.		
	Actor	**Sistema**
2.	Selecciona el reporte deseado, del cual no se ha elaborado ninguna vista.	Muestra un mensaje de alerta "Del reporte seleccionado aun no se ha creado ninguna vista." Y el botón "Aceptar".
3.	Presiona clic en el botón "Aceptar".	Permanece en la misma interfaz.
4.		Termina el Caso de Uso
Flujos alternos		
Nº Evento Selecciona la vista deseada.		
	Actor	**Sistema**
3.	No selecciona la vista deseada.	
4.	Presiona clic en el botón "✖".	Muestra un mensaje de error "Primero debe seleccionar la vista".
5.		Permanece en la misma interfaz.
6.		Termina el Caso de Uso
Relaciones		
Requisitos funcionales	RF 11	
Requisitos no funcionales	RNF 1, RNF 2, RNF 3, RNF 4, RNF 5, RNF 6 RNF 7	
Asuntos pendientes		

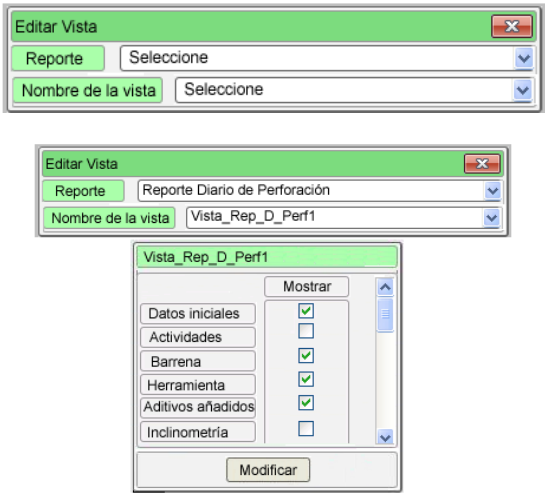
3.8.5.2 Prototipo elemental de interfaz gráfica de usuario

Flujo Básico: Establecer conversación

Sesión 1: Crear vista



Sesión 2: Editar vista



Sesión 3: Eliminar Vista

3.9 Modelo de Análisis

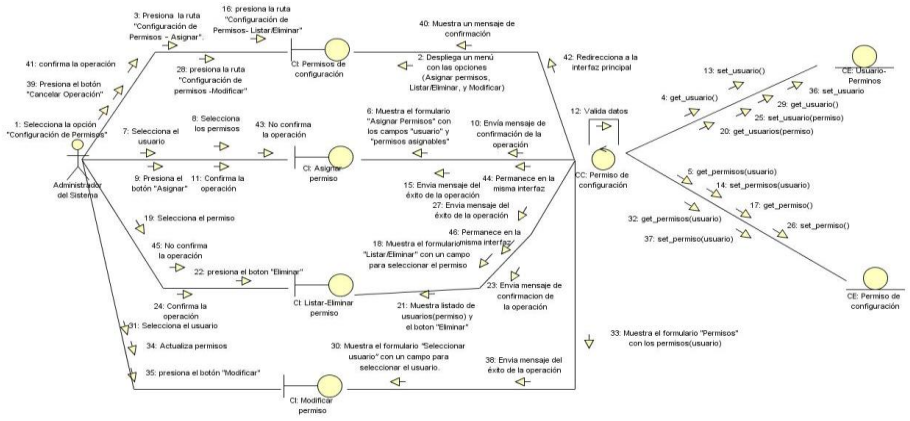
Durante el análisis, se estudian los requisitos que fueron descritos, refinándolos y estructurándolos, construyendo el Modelo de análisis el cual contiene clases del análisis y sus objetos organizados en paquetes que colaboran. Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. RUP propone clasificar estas clases en tres tipos: entidad, interfaz y de control. En una aplicación *cliente/servidor* de tres capas, en la capa de usuario aparecen fundamentalmente clases interfaz ya que allí se ejecutan las aplicaciones del cliente. En la capa intermedia están las clases controladoras ya que en ellas se agrupan los servicios que son compartidos por múltiples aplicaciones. En la capa servidor estarían las clases entidad porque estas contienen los datos que persistirán en la base de datos (35).

3.9.1 Diagramas de Colaboración

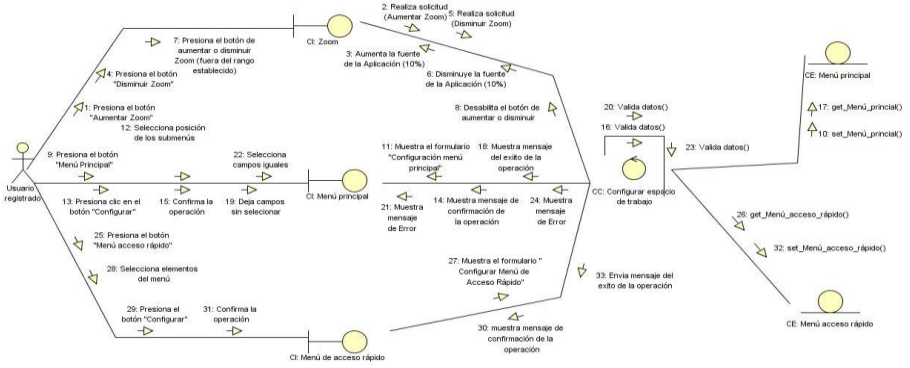
En el Lenguaje Unificado de Modelado (UML) 2.0, un diagrama de comunicación es una versión simplificada del diagrama de colaboración de la versión de UML 1.x. Un diagrama de comunicación modela las interacciones entre objetos o partes en términos de mensajes en secuencia. Los diagramas de comunicación representan una combinación de información tomada desde el diagrama de clases, secuencia, y diagrama de casos de uso describiendo tanto la estructura estática como el comportamiento dinámico de un sistema.

A continuación se muestran los diagramas de colaboración para los CU definidos como críticos en este ciclo de desarrollo.

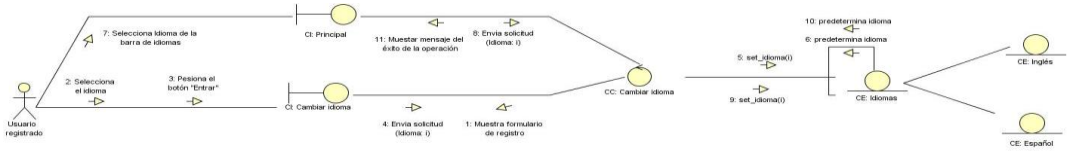
CU: Gestionar permiso de configuración



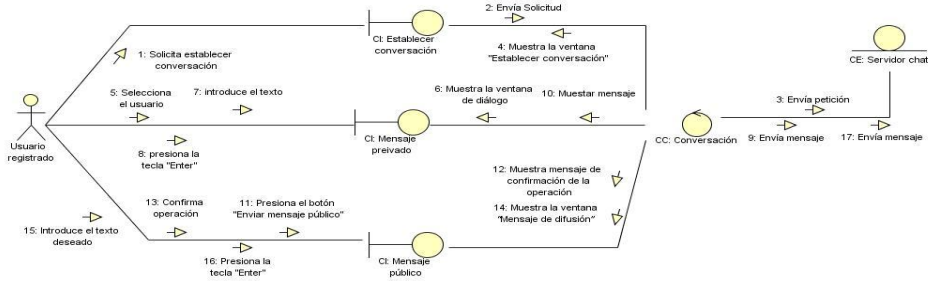
CU: Configurar espacio de trabajo



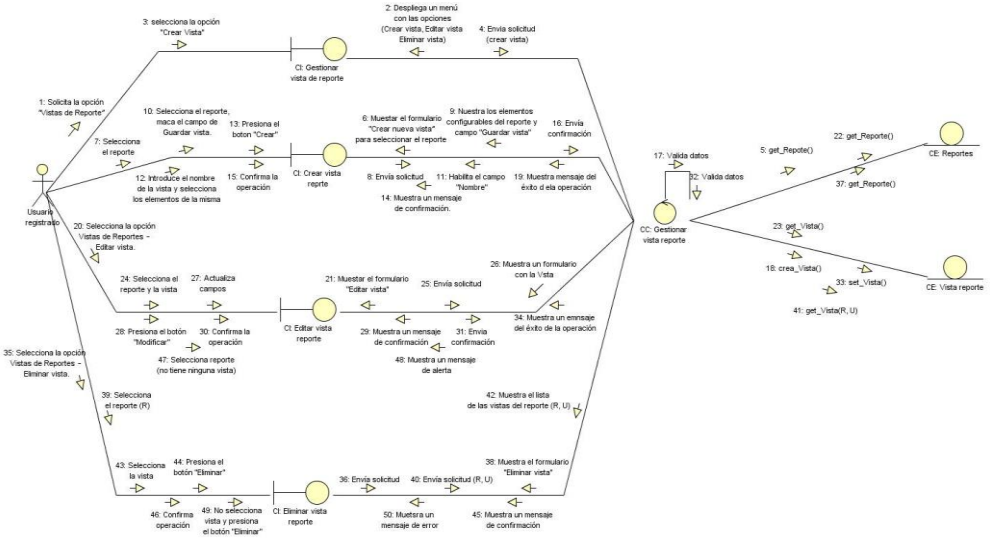
CU: Cambiar idioma



CU: Establecer conversación



CU: Gestionar vistas de reporte



3.10 Conclusiones Parciales

Durante el desarrollo de este capítulo se identificaron y describieron las Clases del Dominio las cuales conforman el Modelo de Dominio, identificándose también los Requisitos Funcionales y No Funcionales. Se conformó el modelo de Casos de Uso del Sistema, describiendo y prototipando cada uno de ellos, para finalmente elaborar los Modelos de Análisis y los Diagramas de Colaboración correspondientes.

Conclusiones

El presente Trabajo Diploma posibilitó no solo desarrollar el Análisis para el tercer ciclo de desarrollo del proyecto Sistema de Manejo Integral de Perforación de Pozos de Petróleo, sino que permitió además arribar a las siguientes conclusiones:

- Con el estudio de los Trabajos Diploma que anteceden a este, se obtuvo el conocimiento necesario para identificar nuevos procesos.
- El estudio del Estado del Arte facilitó la captura de los Requisitos, teniendo en cuenta las necesidades expuestas por el cliente.
- Se profundizó en las Metodologías de Desarrollo de Software más usadas a nivel mundial, seleccionando la adecuada según las características de este Ciclo de Desarrollo.
- La correcta utilización de la herramienta CASE viabilizó la representación de los Artefactos a través de modelos sustentados por UML.
- La utilización de Técnicas de Captura y Validación de Requisitos, permitió su consistencia.
- La consistencia de los Requisitos facilitó la construcción de un apropiado Modelo de Casos de Uso del Sistema.
- Con el Modelo de Casos de Uso del Sistema obtenido, se proporcionó un adecuado Análisis del Sistema.

Recomendaciones

Con el objetivo de incorporar mejoras significativas al Análisis realizado en este Trabajo Diploma se recomienda:

- Diseñar, implementar y probar el Análisis propuesto.

- Profundizar en el estudio del framework para la generación de reportes *PhpReport* desarrollado por el Centro de Tecnologías de Gestión de Datos (DATEC) de la Universidad de Ciencias Informáticas, con el objetivo de una futura utilización del mismo en el proyecto.

- Profundizar en el módulo de internacionalización *i18n* que brinda el framework Symfony, para todo el proceso relacionado con el cambio de idioma de la Aplicación.

- Estudiar la posibilidad de incluir plugins que faciliten el trabajo relacionado con el correo electrónico y el chat.

- Estandarizar los términos, denominaciones y nombres que son usados por en estos Módulos, para regirse por una denominación común y añadirle valor agregado al sistema.

Bibliografía

1. **Tecna**. [En línea] [Citado el: 20 de 10 de 2010.] http://www.tecna.com.
2. Inc, RCLAB S.R.L. [En línea] [Citado el: 20 de 10 de 2010.] http://www.rclabsrl.com.ar.
3. **Shlumberger**. [En línea] [Citado el: 20 de 10 de 2010.] http://www.slb.com.
4. **SAP**. [En línea] [Citado el: 20 de 10 de 2010.] http://www.sap.com/spain/sme/whysap/industries/oil-gas/index.epx.
5. **Petróleo Villa** [En línea] [Citado el: 1 de 11 de 2010.] http://petro-villa.blogspot.com/2009/02/introduccion-la-perforacion-de-pozos-de.html.
6. **(PLS), Petrokem Loggin Services**. [En línea] [Citado el: 1 de 11 de 2010.] http://www.petrokemls.com .
7. **NeuraLog Inc**. [En línea] [Citado el: 1 de 11 de 2010.] http://www.neuralog.com.
8. **cee.** [En línea] [Citado el: 2 de 11 de 2010.] http://cee.com/products/sampling/WellWizard/WellWizard.html.
9. **Free Download Manager**. [En línea] [Citado el: 2 de 11 de 2010.] http://www.freedownloadmanager.org/es/downloads/Bien_Maderero_7024_p.
10. **WellSight**. [En línea] [Citado el: 3 de 11 de 2010.] http://www.wellsight.com.
11. **Peloton. Drilling & Well Data Software Solutions.** [En línea] [Citado el: 3 de 11 de 2010.] http://www.peloton.com/es.
12. **EVA. Ingeniería de Sotfware I. Conferencia 7**. [En línea] [Citado el: 9 de 11 de 2010.]
13. **Pressman, Roger S.** Ingenieria de Software. Un enfoque práctico. La Habana : Felix Varela, 2005.
14. **Ramírez, Alexandra.** [En línea] [Citado el: 18 de 11 de 2010.] http://s3.amazonaws.com/lcp/elida-nieves/myfiles/AS_Clase-231008-1.doc.
15. **Jacaboson, I., Booch, G., Rumbaugh J.** El Proceso Unificado de Desarrollo de Software. s.l. : Addison Wesley, 2000.
16. **Cuevas, David Tavares**. ANÁLISIS Y DISEÑO DEL SISTEMA DE MANEJO INTEGRAL DE PERFORACIÓN DE POZOS. Ciudad de la Habana : s.n., 2009.
17. **UNAP**. [En línea] [Citado el: 27 de 11 de 2010.] http://www.unap.cl/~setcheve/Metrica/m/index.html.
18. **Sanchez, María A. Mendoza**. [En línea] [Citado el: 27 de 11 de 2010.] http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.
19. **León Rodríguez, Milton Benito**. Diseño del Segundo Ciclo de Desarrollo del Sistema de Manejo Integral de Perforación de Pozos. 2010.
20. [En línea] 2005. [Citado el: 1 de 12 de 2010.] http://www.ambysoft.com/unifiedprocess/agileUP.html.

21. **ProyectosAgiles.org.** [En línea] [Citado el: 1 de 12 de 2010.] http://www.proyectosagiles.org/que-es-scrum.
22. **José H. Canós, Patricio Letelier yMª Carmen Penadés**. [En línea] [Citado el: 1 de 12 de 2010.] http://www.willydev.net/descargas/prev/TodoAgil.pdf.
23. [En línea] 23 de 5 de 2008. [Citado el: 1 de 12 de 2010.] http://crystalmethodologies.blogspot.com.
24. **Java Mexico-Comunonad de Desarrolladores Mexicanos**. [En línea] 27 de 02 de 2010. [Citado el: 2 de 12 de 2010.] http://www.javamexico.org/blogs/carraro/que_es_dsdm.
25. **Huarachl, Maritza**. Ingernieria de Softeare. EL ALTO – BOLIVIA : s.n., 2009.
26. **Ensayos de calidad, Tareas, Monografías y Trabajos de Investigación Personalizados** - BuenasTareas.com. [En línea] 2010. [Citado el: 2 de 12 de 2010.] http://www.buenastareas.com/ensayos/Ssdam/51800.html.
27. **Ramírez Cruz, Eduardo**. Análisis y Diseño del Sistema para Graficar Columnas Litológicas de Pozos de Petróleo.2010.
28. **Introduction to OMG's Unified Modeling Language.** [En línea] Julio de 2005. [Citado el: 5 de 12 de 2010.] http://www.omg.org/gettingstarted/what_is_uml.htm.
29. **GSInnova Grupo Soluciones**. [En línea] 2007. [Citado el: 5 de 12 de 2010.] http://www.rational.com.ar/herramientas/roseenterprise.html.
30. **INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA**. Herramientas Case. 1999.
31. **sparxsystems.** [En línea] 2010. [Citado el: 6 de 12 de 2010.] http://www.sparxsystems.com.ar/products/ea/index.html.
32. **Visual Paradigm.** [En línea] 2010. [Citado el: 6 de 3 de 2010.] http://www.visual-paradigm.com.
33. **Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio.** [En Linea] 2011 [Citado el: 10 de 1 de 2011.] http://www.sistedes.es/TJISBD/Vol-3/No-3/articles/pnis-09-berrocal-casos.pdf
34. **Ingeniería De Requerimientos. Ingeniería De Software** [En Linea] 2011 [Citado el: 1 de 3 de 2011.] http://www.monografias.com/trabajos6/resof/resof2.shtml
35. **EVA. Ingeniería de Sotfware I. Conferencia 10.** [En línea] [Citado el: 3 de 5 de 2011.]
36. **Ingeniería de Requisitos en Aplicaciones para la Web** [En Linea] 2011 [Citado el: 2 de 3 de 2011.] http://www.pst.ifi.lmu.de/~kochn/ideas03-escalona-koch.pdf
37. **Arias Chaves, Michael.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software [En Linea] 2011 [Citado el: 2 de 3 de 2011.] http://www.latindex.ucr.ac.cr/intersedes10/10-art_11.pdf
38. **Las mejores prácticas para la internacionalización de las aplicaciones web.** [En Linea] 2011 [Citado el: 14 de 3 de 2011.] http://es.w3support.net/index.php?db=so&id=155719
39. **VALIDACION DE REQUISITOS.** [En Linea] 2011 [Citado el: 21 de 3 de 2011.] http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf
40. **Modelo de Dominio**. [En Linea] 2011 [Citado el: 22 de 3 de 2011.] http://migueljaque.com/index.php