



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 9



OFICINA NACIONAL DE RECURSOS MINERALES

HERRAMIENTA PARA LA CAPTURA DE DATOS DE CAMPO DEL INVENTARIO DE PETRÓLEO Y GAS PERTENECIENTE AL SISTEMA DE GESTIÓN DE DATOS GEOLÓGICOS

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

Autor: Alberto Menéndez Romero

Tutor: Ing. Vladimir Martell Fernández

Ciudad de La Habana, 20 de junio de 2010
"Año del 52 de la Revolución"

NOTA: Este documento constituye un extracto de 80 páginas del documento original.

Mira el electrón: Se comporta según lo mires.

Mírate: Te conviertes en la visión que tienes de ti.

Te ves según lo que crees de ti.

Crees en ti y te vuelves a crear.

Cuando no crees en tu potencial te niegas toda posibilidad de ser.

Crear es crear y la creencia falsa genera cosas vanas.

Sin asiento en tu realidad, esas creaciones son sólo vanidad.

El ser es original.

Como una obra de arte, sólo puede ser de veras valorado por su autenticidad.

El arte de vivir originalmente, el de ser auténticos, es el arte de ser uno mismo.

De: Jorge Carvajal Posada en "El arte de ser uno mismo"

Dedicatoria

A mis padres: Rodalmis y Alberto

A mis abuelos: Mirtha y Celestino

A mis hermanas: Liliet, Lesliet, Elianis y Lianis

A mi novia Yeni

A mis amigos (los de verdad)

Agradecimientos

A mi tutor Vladimir, por todo su apoyo, confianza y ayuda, “él es el mejor de todos”.

A mis padres por toda su entrega, confianza y ayuda aunque las situaciones fueran difíciles, por todo el sacrificio hecho hasta el día de hoy

A mis abuelos (los mejores y más lindos del mundo) por estar al pendiente

A mis hermanas por quererme tanto, portarse bien y ser tan lindas (☺)

A Yeni (mi novia), por vivir los momentos felices y difíciles a mi lado, por su apoyo y comprensión, por su belleza y amor.

A mi familia por ser tan unida y única

A mis amigos de grupo: los de antes y los de ahora. A mi amigo Adrian y Jose (Bolo), a las polillas Aliana y Liset

A los del SGDG: Eddy, Dago, Figura, Joel (Mentica), el Compa, Tony, Reinaldo (Aldo), Héctor, Yamil (Tulipán Rojo), Yamilé (premio Ada), Rocny, en fin, a todos.

A mi gente del Secretariado FEU.

A los que me preguntaron alguna vez “¿Y la tesis cómo va?”

... a todos gracias.

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Oficina Nacional de Recursos Minerales y a la Facultad 9 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 _____.

Alberto Menéndez Romero
Autor

Ing. Vladimir Martell Fernández
Tutor

DATOS DEL TUTOR

Nombre y apellidos: Ing. Vladimir Martell Fernández.

Correo electrónico: vmartell@uci.cu

Categoría docente: Instructor Recién Graduado.

Año de graduación: 2008.

Profesión: Ingeniero en Ciencias Informáticas.

Breve descripción: Graduado en la Universidad de las Ciencias Informáticas. Actualmente profesor del Departamento Técnicas de Programación y 2do Jefe del Departamento Geoinformática del Centro de Desarrollo “Geoinformática y Señales Digitales” de la facultad 9.

RESUMEN

Producto del vertiginoso auge de las Tecnologías de la Información y las comunicaciones, las cuales han impuesto un nuevo orden mundial al proceso de gestión de la información, la mayoría de las empresas de la actualidad optan por alternativas automatizadas con el objetivo de seguir desarrollando sus principales procesos a la altura que supone la dinámica de la toma de decisiones de la actualidad.

Con la presente investigación se pretende desarrollar una aplicación informática de beneficio para la Oficina Nacional de Recursos Minerales que facilite la captura de datos de campo que maneja en Sistema de Gestión de Datos Geológicos implementado en el Centro de Desarrollo de Software Geoinformática y Señales Digitales de la facultad 9 en la Universidad de las Ciencias Informáticas.

De la explotación de la futura herramienta se derivan ventajas importantes como son: mayor eficiencia en el proceso de captura de datos de campo con reducción de posibles errores producto de la actividad humana, estandarización en el uso de formatos de entrada a la aplicación, entre otras.

Palabras claves:

Aplicación informática, implementación, Inventario Nacional de Recursos y Reservas de Petróleo y Gas, ONRM, Sistema de Gestión de Datos Geológicos.

ÍNDICE

Introducción.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	7
1.1 Introducción	7
1.2 Conceptos asociados al dominio del problema	7
1.3 Objeto de estudio.....	10
1.3.1 Descripción General	10
1.3.2 Descripción actual del dominio del problema	11
1.3.3 Situación problemática	11
1.4 Existencia de soluciones semejantes a la presente investigación	12
1.5 Conclusiones	13
CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A UTILIZAR.....	14
2.1. Introducción	14
2.2. Aplicación Informática.....	14
2.2.1. Aplicación de escritorio o Desktop	15
2.3. Arquitectura de software.....	15
2.3.1. Modelo – Vista – Controlador (MVC).....	16
2.4. Lenguajes de programación	17
2.4.1. Java	18
2.4.2. Biblioteca gráfica Swing	19
2.5. Sistemas Gestores de Base de Datos.....	20
2.5.1. HDSQL	21
2.6. XML	22
2.7. Metodologías para el desarrollo del software	23
2.7.1. El proceso Unificado de desarrollo de Software (RUP).....	24
2.8. El Lenguaje Unificado de Modelado (UML)	25
2.9. Visual Paradigm como herramienta CASE	26
2.10. Entornos de Desarrollo Integrado (IDE).....	27
2.10.1. NetBeans	27
2.11. Conclusiones.....	28
CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	29
3.1. Introducción	29
3.2. Modelo del Negocio	29
3.2.1. Actores y trabajadores del negocio	29
3.2.2. Procesos del Negocio.....	30
3.2.3. Diagrama de casos de uso del Negocio.....	31
3.2.4. Diagrama de Actividades.....	32
3.2.5. Descripción textual de los casos de uso del Negocio	32
3.3. Requisitos Funcionales.....	34
3.4. Requisitos No Funcionales	35
3.5. Descripción del Sistema Propuesto.....	36
3.5.1. Descripción de los Actores	36
3.5.2. Diagrama de casos de uso del Sistema	37
3.5.3. Descripción textual de los casos de uso del Sistema.....	38
3.6. Conclusiones	44
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	45
4.1. Introducción	45

4.2. Principios de Diseño	45
4.2.1. Estándares de la Interfaz de la aplicación.....	46
4.2.2. Concepción general de la ayuda	46
4.3. Diagrama de clases del diseño.....	47
4.4. Diseño de la base de datos	48
4.4.1. Diagrama Entidad – Relación.....	48
4.4.2. Diagrama de clases persistentes	49
4.5. Generalidades de la implementación	50
4.5.1. Modelo de componentes	50
4.5.2. Modelo de despliegue	50
4.6. Prueba del sistema propuesto.....	51
4.6.1. Pruebas de caja negra	51
4.6.2. Caso de uso: Exportar Informe XML.	53
4.7. Conclusiones	57
CONCLUSIONES	58
REFERENCIAS BIBLIOGRÁFICAS	60
BIBLIOGRAFÍA	61
ANEXO III DIAGRAMAS DE CLASES DEL DISEÑO	63

Introducción

El surgimiento de las Tecnologías de la Información y la Comunicación (en lo adelante TIC) y su avance acelerado en los últimos años, ha generado grandes cambios en la sociedad, tanto en las estructuras económicas, como en las culturales y educativas. Su utilización constituye un eslabón fundamental para impulsar el desarrollo económico y social de un país, para esto debe lograrse la incorporación de la sociedad como sujeto impulsor de esta estrategia de desarrollo. (Anabel Carreño 2008).

“La tecnología ha impactado en todas las esferas de la vida social” (Rangel 2006), influyendo de forma muy directa en cada uno de sus sectores; determinando nuevos métodos e imponiendo un nuevo ritmo de vida. En la actualidad nadie pone en duda que la investigación científica, el desarrollo y la innovación tecnológica, se potencian con la utilización y aprovechamiento de las TIC. Su incorporación en diversos espacios provoca una revolución tecnológica que se enmarca como un medio alternativo de informar y comunicar; se dice que se vive en la sociedad de la información y en la sociedad del conocimiento, las TIC tienen estrecha relación con estas dos aseveraciones.

La informatización de la sociedad se define en Cuba, como el proceso de utilización ordenada y masiva de las Tecnologías de la Información y la Comunicación para satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad. Este proceso busca lograr más eficacia y eficiencia, que permitan una mayor generación de riquezas y hagan sustentable el aumento sistemático de la calidad de vida de los cubanos.

Formando parte del proceso de informatización de la sociedad cubana se encuentra el sector geológico que mantiene un desarrollo considerable y sostenido desde los últimos años obteniéndose importantes resultados:

- Se logró la digitalización de los datos referentes a los principales recursos minerales como níquel y petróleo, mapas geológicos a diferentes escalas, levantamientos aerofísicos nacionales, más de 100 bases de datos de las principales zonas de interés económico, estructuradas y documentadas.

- El desarrollo de la teoría para la normalización de las Bases de Datos Geológicas (Tecnología de Base de Datos con Atributos y Registros Variables (BDARV)).
- Análisis (versión informática) de los datos relacionados con minerales sólidos (Programa GeoDato).
- El desarrollo de medios de procesamiento de la información geológica como software de minería y sistemas de información geográfica.

Una de las entidades que forma parte del área de la geología en Cuba es la Oficina Nacional de Recursos Minerales (en lo adelante ONRM), creada cuando entra en vigor a partir de 1995, la Ley No 76 o Ley de Minas (publicada en la Gaceta Oficial de la República en ese año).

La ONRM, autoridad minero petrolera de la República de Cuba, independiente pero subordinada al Ministerio de la Industria Básica (MINBAS), es la responsable directa de salvaguardar el conocimiento geológico de la nación que actualmente está constituido por los fondos documentales de proyectos, estudios técnico-económicos, informes, bases de datos, entre otros.

Entre sus misiones fundamentales se encuentra garantizar el aprovechamiento racional de los recursos minerales del país y ejercer con eficiencia, rigor técnico y responsabilidad el control estatal sobre las actividades de la geología, minería y petróleo en todo el territorio nacional y la zona económica de Cuba.

A finales de los años noventa, la ONRM enfrentó la informatización de su gestión mediante el empleo de herramientas de la ofimática, la adquisición y asimilación de Sistemas de Información Geográfica (SIG), los sistemas de minería y de geoestadística, entre otras soluciones; de este modo se informatizó en una primera aproximación el cumplimiento de las funciones más rutinarias de la ONRM.

Sin embargo, a pesar del avance que significó la puesta en práctica de estas soluciones, no suplieron todas las necesidades requeridas, era necesario preservar de manera sustentable y poner a disposición de la sociedad el conocimiento geológico cubano con la agilidad que exige la dinámica de la toma de decisiones en la actualidad.

Es por ello, que se hace necesario la existencia de nuevas soluciones para garantizar que la información no se pierda o se deteriore y que el nivel de acceso a la misma sea permitido solo a las personas con responsabilidades y permisos para acceder, consultar y manipular estas informaciones. De este modo la ONRM junto a otras empresas dedicadas al estudio y la investigación geólogo-minera se suman al Programa Nacional de Informatización del Conocimiento Geológico (en lo adelante PNICG) surgido a mediados del año 2006 por la necesidad de expandir el desarrollo informático para adquirir un aprovechamiento óptimo del conocimiento de la geología.

A partir de septiembre del 2006 la Universidad de las Ciencias Informáticas (en lo adelante UCI) pasa a formar parte del PNICG al aprobar el Consejo de Estado la decisión planteada por parte de la Ministra de la Industria Básica, de autorizar a la UCI a incluir entre sus proyectos productivos la informatización de las áreas de la ONRM. Surge así, con este objetivo, en la Facultad 9 de la UCI, en el Centro de Desarrollo de Software Geoinformática y Señales Digitales, el proyecto Sistema de Gestión de Datos Geológicos (en lo adelante SGDG), producto del cual se obtendría mayor rapidez en la toma de decisiones de la entidad y la preservación de alto grado del conocimiento geológico alcanzado en Cuba al informatizar en gran medida los procesos que tienen lugar en la autoridad minera del país.

Dentro de estos procesos se encuentra el de captura de datos de campo que tiene lugar en la Dirección de Hidrocarburos de la ONRM (ONRM 2008), el cual registra en un sistema informático los datos de campo de los contratistas¹ logrando un mayor control sobre los mismos. Sin embargo, este proceso se ve actualmente afectado por el nivel de diversidad de la entrega, pues existen contratistas que lo realizan en disímiles formatos de soporte digital y otros sencillamente en copia dura, lo que trae como consecuencia:

- Demora en la entrega de los datos de campo por parte de los contratistas con tecnología más pobre, atrasando el proceso de creación del Inventario Nacional de Recursos y Reservas de Petróleo y Gas (en lo adelante INRRPG) en la ONRM y dificultándolo por la diversidad de los formatos en que son entregados.

¹ Persona natural o jurídica debidamente autorizada para el ejercicio de una o varias fases de la actividad minera por el correspondiente título.

- Un mayor esfuerzo por parte del personal de la ONRM para registrar los datos de campo en el sistema, pues necesitan un mayor conocimiento de los diferentes formatos en que son entregados.

Por otra parte, la actividad de inserción manual de los datos desde los informes de los contratistas al sistema aumenta la probabilidad de obtener errores en la información y a su vez, disminuye la velocidad del proceso, por si fuera poco, una vez introducidos los errores, su rectificación es más lenta y costosa.

Después de haber expuesto lo planteado, luego de un análisis profundo que constata la problemática existente, se ha identificado el siguiente **problema científico** el cual da origen a esta investigación:

¿Cómo agilizar los procesos de captura de datos de campo del módulo Inventario de Petróleo y Gas que forma parte del Sistema de Gestión de Datos Geológicos?

Como resultado del análisis del propio problema se definió como **objeto de estudio** los procesos de captura de datos del módulo Inventario de Petróleo y Gas del SGD, y como **objetivo general** del presente trabajo de diploma: Desarrollar una aplicación que permita la captura de los datos de campo del módulo Inventario de Petróleo y Gas que forma parte del SGD, siendo el **campo de acción** la informatización de los procesos de captura de datos de campo del Inventario de Petróleo y Gas del SGD.

Por lo anteriormente expuesto se propone como **idea a defender que**:

El desarrollo de una herramienta informática permitirá agilizar los procesos de captura de datos del módulo Inventario de Petróleo y Gas que forma parte del SGD.

Para lograr el cumplimiento de los objetivos expuestos anteriormente se desarrollarán las diferentes tareas en la presente investigación:

- 1) Caracterizar el proceso captura de datos del módulo Inventario de Petróleo y Gas del SGD en la Oficina Nacional de Recursos Minerales.
- 2) Caracterizar las tendencias y tecnologías actuales a desarrollar.

- 3) Elaborar el Diagrama de casos de uso del Negocio del módulo Inventario de Petróleo y Gas.
- 4) Especificar los requisitos funcionales del software.
- 5) Elaborar el Diagrama de casos de uso del Sistema del módulo Inventario de Petróleo y Gas.
- 6) Elaborar los diagramas de clase del diseño del módulo Inventario de Petróleo y Gas.
- 7) Elaborar el diagrama de implementación.
- 8) Implementar de los CU definidos.
- 9) Desarrollar los Casos de Prueba que certifiquen la veracidad de los algoritmos empleados.

Como resultado de todas las tareas y actividades desarrolladas y además de un largo proceso de investigación se esperan los siguientes resultados:

- 1) Confección de una herramienta informática para la captura datos del Inventario de Petróleo y Gas.
- 2) La documentación técnica del proceso ingenieril asociada al desarrollo de la herramienta anterior.

En el trascurso y desarrollo de esta investigación científica se tendrán en cuenta una serie de métodos científicos para la obtención, procesamiento y llegada a conclusiones, los cuales se exponen a continuación:

Métodos Teóricos

1. Histórico – lógico: Se utilizará para el estudio de los procesos de gestión de información, trabajos e investigaciones anteriores y partir de ellos como base para el desarrollo de la presente investigación.
2. Modelación: Se utilizará en la modelación de los diagramas dentro de la metodología de desarrollo de software seleccionada para llevar a cabo la solución.
3. Analítico - Sintético: Se utilizará para el estudio de la bibliografía y la selección de la más adecuada para el desarrollo del presente trabajo; además, para dividir el problema de investigación en elementos por separado, profundizar en el

estudio de cada elemento y luego sintetizarlos en la confección de la solución propuesta.

Métodos Empíricos

1. Entrevista: Se realizarán entrevistas a los especialistas del Inventario Nacional de Recursos y Reservas de la Oficina Nacional de Recursos Minerales.

El presente trabajo está dividido en 4 capítulos fundamentales:

Capítulo 1: En este capítulo se expone el objeto de estudio, la descripción del entorno donde se encuentra el negocio y su organización, se describe detalladamente la situación problemática y el análisis de otras soluciones que puedan brindar respuesta al problema científico planteado en el presente trabajo.

Capítulo 2: En este capítulo se explican las principales tecnologías, lenguajes de programación y herramientas que se utilizarán para la construcción de la solución propuesta, así como las ventajas de utilizarla.

Capítulo 3: En este capítulo se describe la solución propuesta, se seleccionan los actores, los trabajadores, los casos de uso del negocio, los requisitos funcionales y no funcionales y los casos de uso del sistema, todo esto unido a los correspondientes diagramas que lo modelan.

Capítulo 4: En este capítulo se plantea la construcción propuesta en el Capítulo 3, en función de diagramas de clases y estándares del diseño, generalidades de la implementación, diseño de la base de datos, modelo de despliegue y modelo de implementación, finalizando el capítulo con los diseños de casos de pruebas que validen la implementación realizada.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el capítulo que sigue, se presentan los conceptos relacionados con la fundamentación teórica de la presente investigación, el objeto de estudio y la existencia de soluciones informáticas que respondan a las exigencias actuales de la ONRM. Finalmente se detallan los principales procesos que se llevan a cabo hoy como parte de las actividades de la captura de datos del balance de petróleo y gas.

1.2 Conceptos asociados al dominio del problema

Los conceptos asociados al dominio se toman a partir de las definiciones más actualizadas establecidas internacionalmente por organizaciones mundiales y regionales del petróleo; estos conceptos son útiles para el entendimiento del problema en cuestión, y ayuda a entender las relaciones que existen entre ellos.

Hidrocarburos

Los hidrocarburos son los compuestos químicos resultantes de la combinación del carbono con el hidrógeno. Son los compuestos orgánicos más simples y pueden ser considerados como las sustancias principales de las que se derivan todos los demás compuestos orgánicos. (ONRM)

Gas natural

El gas natural es la mezcla de hidrocarburos ligeros que existe en fase gaseosa (puede ser libre o acompañante) o disuelto en el petróleo en el subsuelo, que generalmente viene acompañado de vapor de agua, compuestos de azufre como el sulfuro de hidrógeno así como de otros gases no hidrocarbonados, tales como el dióxido de carbono, el nitrógeno y el helio. En la mayoría de los casos, el gas natural solo permanece en forma de gas a condiciones atmosféricas de presión y temperatura. (ONRM)

Petróleo

El petróleo es una mezcla natural, líquida, aceitosa e inflamable de hidrocarburos gaseosos, líquidos y sólidos, en los cuales los componentes gaseosos y sólidos son disueltos en hidrocarburos líquidos, trayendo como consecuencia la formación de soluciones o suspensiones coloidales, pudiendo contener además otros compuestos no hidrocarbonados, ejemplo de ello lo constituye el dióxido de carbono, compuestos de azufre, nitrógenos y otros. (ONRM)

Yacimiento: Depósito o acumulación natural de petróleo crudo y gas natural confinado, de interés comercial, local y único en profundidad. El yacimiento puede estar compuesto de una o varias capas productoras. (MINBAS, 2007)

Recursos

Son las cantidades totales de petróleo y gas así como sustancias relacionadas que son estimadas, y que estén contenidas o que han sido producidas desde acumulaciones conocidas, más aquellas cantidades estimadas en acumulaciones que todavía no se han descubierto. (ONRM)

Recursos de Petróleo: Son aquellas concentraciones de petróleo que existen en la corteza terrestre en forma y cantidad tales que pueden ser estudiadas por los métodos geólogo geofísicos y de ingeniería. (ONRM)

Balance

El Balance se elabora anualmente, con el objetivo de tener toda la información sobre el estado de las reservas y recursos minerales; controlando y garantizando el uso racional de las mismas; siendo esta, una de las funciones fundamentales de la Oficina Nacional de Recursos Minerales. Es el estado que refleja la situación del patrimonio de una entidad en un momento determinado. Se refiere al conjunto de actividades que se realizan sobre los datos de recursos y reservas de petróleo y gas que existen en todo el territorio nacional y en la zona económica de Cuba. (ONRM)

Inventario de Recursos y Reservas de Petróleo: El Inventario Nacional de Recursos y Reservas de la República de Cuba es el documento Oficial mediante el cual se

actualiza cada año tanto los recursos así como las reservas con que el estado cubano dispone para elaborar los planes de extracción y proyección de la exploración para las etapas siguientes. La Oficina Nacional de Recursos Minerales como entidad encargada de su elaboración y control, certifica que todos los datos contenidos en este documento reflejan efectivamente y de manera oficial las cifras de cada yacimiento y cada prospecto. (ONRM)

Entidades

Las entidades son las Empresas Nacionales, Contratistas y el Centro de Investigación que trabajan en la extracción y explotación de hidrocarburo en Cuba (Petróleo y Gas Natural). (ONRM)

Empresas Nacionales

Son las empresas cubanas que trabajan en la perforación y extracción de petróleo, conocidas por Empresas de Perforación y Extracción de Petróleo (EPEP), que existen, por ejemplo en Occidente específicamente en Boca de Jaruco, en el Centro concretamente en Varadero y UEPEP (Unidad Empresarial de Perforación y Extracción de Petróleo) en Majagua. (ONRM)

CEINPET

CEINPET es el Centro de Investigación de Petróleo en Cuba que está avalado por una alta calificación y experiencia de trabajo, que se encarga de dar respuesta de forma integral a toda la actividad petrolera, desde la exploración hasta la refinación, en el proceso de investigación – desarrollo – producción. Hasta estos momentos hay un solo centro de investigación en Cuba, que tiene operaciones en el campo donde extrae, procesa e investiga. (ONRM)

Contratistas

Persona jurídica que establece relaciones petroleras contractuales con el Estado cubano u otra entidad, ya sea como añadida o sustituta de partes anteriores de un contrato. (ONRM)

SGDG

Se refiere el Sistema de Gestión de Datos Geológicos. Aplicación informática desarrollada en Centro de Desarrollo de Software GEySED, perteneciente a la facultad 9 de la Universidad de las Ciencias Informáticas; con el objetivo de informatizar los procesos que tienen lugar en las distintas áreas de la ONRM.

1.3 Objeto de estudio

1.3.1 Descripción General

El proceso de captura de datos parte de la información que entregan los contratistas y entidades que colaboran con la ONRM al INRRPG, relacionadas con el movimiento de los Recursos y Reservas ocurridos durante el año calendario en operaciones; se contemplan en dos modelos de tablas:

- ❖ Tabla de los Recursos.
- ❖ Tabla de las Reservas.

Tabla de los Recursos: Refleja los datos relacionados con el movimiento de los recursos. Contempla el estado inicial, todos los cambios ocurridos durante el año (re-estimación, re-clasificación, recursos a reservas y viceversa, etc.) y el estado al cierre del año. (Ver Anexo I – Tabla 1)

Tablas de las Reservas: Refleja los datos relacionados con el movimiento de los reservas. Contempla el estado inicial, todos los cambios ocurridos durante el año (extracción, re-estimación, re-clasificación, pérdidas y dilución, abandonos, etc.) y el estado al cierre del año. (Ver Anexo I – Tabla 2,3 y 4)

Al INRRPG se incorporan los recursos y reservas de los informes de exploración y/o explotación aprobados por la ONRM a través de los certificados elaborados por los especialistas de conformidad con el grupo de balance. De éste se emiten las certificaciones de recursos y reservas que avalan de forma oficial la existencia de las cantidades por categorías, con sus calidades correspondientes en los yacimientos y áreas petroleras.

1.3.2 Descripción actual del dominio del problema

En la actualidad la ONRM utiliza la aplicación informática SGD G, la cual se creó con el objetivo de informatizar los procesos que tienen lugar en las distintas áreas de la entidad. Entre estos procesos se encuentra el de captura de los datos que tienen lugar en la Dirección de Hidrocarburos, encargada de la realización de las actividades y servicios que conciernen al Petróleo y Gas explorado, perforado o extraído dentro del territorio económico de Cuba.

El SGD G, como consecuencia de lo anterior, le permite a la ONRM realizar el proceso de captura de datos de campos introduciendo al sistema los datos primarios que entregan las diferentes compañías extranjeras, empresas nacionales, CEINPET, contratistas de forma general, que posteriormente se incorporarán al INRRPG.

Dentro del Grupo de Balance se designa un especialista encargado de validar la integridad de los datos y certificar su correcta elaboración. Una vez completado esta actividad, el mencionado especialista introduce de forma manual absolutamente todos los datos desde los informes adquiridos hacia el SGD G.

1.3.3 Situación problemática

Actualmente, este proceso de captura de datos de campo se ve afectado de forma negativa por los distintos formatos de entrega por parte de los contratistas; pues en la actualidad estos datos llegan a la oficina en varios formatos como: copia dura, documentos de Microsoft Word, Excel, Access, presentaciones de Microsoft PowerPoint, archivos digitales de Adobe Acrobat Reader (PDF), entre otras variedades. Toda esta diversidad de formatos supone mayor esfuerzo y conocimiento informático por parte del personal encargado en introducir los datos al SGD G.

Otro factor importante que frena la creación efectiva del INRRPG lo constituyen la imposibilidad de los contratistas menos adelantados tecnológicamente de entregar la información solicitada a la ONRM pues al no contar con apoyo automático necesitan de mayor cantidad de tiempo para elaborar la documentación.

Finalmente, la agilidad del proceso se compromete de manera considerable en el hecho de tener que reescribir los datos desde los informes obtenidos del campo hacia la aplicación que los gestiona, introduciéndose en buena medida la posibilidad de errores imposibles de detectar en esta etapa del procesamiento de la información, que generarían resultados de análisis sobre bases incorrectas. Unido a ello, el evidente retraso debido el traspaso innecesario de documentos con un mismo objetivo.

Por otra parte, la actividad de inserción manual de los datos desde los informes de los contratistas al sistema aumenta la probabilidad de obtener errores en la información y a su vez, disminuye la velocidad del proceso, por si fuera poco, una vez introducidos los errores, su rectificación es más lenta y costosa.

Es por ello que se hace necesaria la existencia de un mecanismo que permita la entrada de la información al SGD G de forma automatizada y ordenada, así como la validación de estos datos de forma tal que el documento tenga la calidad necesaria.

1.4 Existencia de soluciones semejantes a la presente investigación

Durante el desarrollo de la presente investigación, a través de entrevistas con los especialistas encargados de elaborar el INRRPG y extensas búsquedas realizadas en Internet, no se tiene conocimiento sobre la existencia de aplicaciones que estandaricen los informes que se incorporan al INRRPG o que se asemejen a los objetivos de la presente investigación.

Las compañías petroleras de otros países que operan en Cuba y las empresas cubanas, a excepción de la ONRM, no cuentan con el diseño de una aplicación informática que les facilite el almacenamiento de los datos primarios que se recogen de la exploración, extracción y explotación del petróleo.

En la actualidad, el SGD G es la única solución informática que existe con el objetivo de incorporar los datos de campos de los contratistas al INRRPG y su utilización se enmarca únicamente en la ONRM, además de suponer la existencia de las deficiencias que se plantearon anteriormente.

1.5 Conclusiones

Durante el desarrollo del presente capítulo se expusieron los conceptos asociados al dominio del problema para una mejor comprensión de la investigación, así como las relaciones que existen entre ellos. Se realizó un estudio entre los procesos que se desarrollan en la extracción y explotación de petróleo y los factores determinantes que influyen en la elaboración del INRRPG en Cuba.

Luego se determinó que la aplicación que existe actualmente no brinda las posibilidades necesarias para llevar a cabo con excelencia los procesos concernientes a la gestión de los datos del INRRPG que se desarrolla en la ONRM.

CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS ACTUALES A UTILIZAR

2.1. Introducción

El presente capítulo describe las tecnológicas a utilizar para la construcción de la herramienta propuesta, que incluye: tipos de aplicaciones, metodologías de desarrollo de software, herramientas de modelado, lenguajes de programación y sistema gestor de base de datos; las principales ventajas y la opinión del autor de la actual investigación en cada caso.

Como producto de las actividades del arquitecto de software, las herramientas, lenguajes y tecnologías que a continuación se describen fueron comparadas y evaluadas de tal forma que se consideran factibles para la construcción de la aplicación propuesta durante el ciclo de desarrollo de software del sistema SGD G.

2.2. Aplicación Informática

En un ordenador, computadora o sistema de cómputo, el software representa las instrucciones responsables para que el hardware (la máquina) realice su tarea. Como concepto general, el software puede dividirse en varias categorías basadas en el tipo de trabajo realizado: por una parte, el sistema operativo que controla los trabajos del sistema, y por otra, el software de aplicación que dirige las distintas tareas que se requiera.

Un programa informático, sinónimo de software, es el conjunto de instrucciones que ejecuta un ordenador o computadora donde una vez ejecutadas realizarán una o varias tareas (Stair, Ralph M, 2003); también suele referirse al código fuente original o a la versión ejecutable.

Una aplicación informática se puede definir como un programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Generalmente resultan ser una solución informática para la automatización de ciertas tareas complicadas como pueden ser la contabilidad, la redacción de documentos, la gestión de un almacén, la comunicación de datos, entre otros.

2.2.1. Aplicación de escritorio o Desktop

Cuando se refiere al término aplicación de escritorio o desktop se está en presencia de una aplicación informática que se ejecuta en una computadora de escritorio o en una portátil (laptop) de manera local, en contraste a las aplicaciones web.

Este tipo de aplicación informática se ejecuta en el ordenador en su propia dirección de memoria física o virtual y no requiere cargarse en otras aplicaciones como navegadores para poder funcionar. No obstante, una aplicación de escritorio puede encontrarse distribuida en varios ordenadores y comunicarse a través de diferentes mecanismos de la red, tales como TCP/IP, UDP, RPC, entre otros; además de poder hacer uso de servidores de almacenamiento de datos.

2.3. Arquitectura de software

La arquitectura de una aplicación es la vista conceptual de la estructura de ella misma. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código. (Cornejo 2001)

Una arquitectura adecuada es la pieza clave para lograr el cumplimiento de los requerimientos funcionales como no funcionales de un sistema; su representación y diseño se han convertido en temas dominantes de la ingeniería de software; donde encontrar una definición aceptada universalmente no es tarea fácil.

Se puede decir que la arquitectura de software de un programa o un sistema computacional es la estructura del sistema, la cual comprende elementos del software, las propiedades externamente visibles de esos elementos, y las relaciones entre ellos.

La definición de la IEEE² Std 1471-2000 sobre el término arquitectura de software plantea que "...es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución".

² Institute of Electrical and Electronics Engineers

Booch, Jacobson y Rumbaugh plantean que la “Arquitectura es la estructura de los componentes más significativos de un sistema interactuando a través de interfaces con otros componentes conformados por componentes sucesivamente pequeños e interfaces”. (Jacobson. I, Booch. G, Rumbaugh. J, 2000)

En los sistemas grandes y complejos la arquitectura de software brinda al equipo de desarrollo una visión común, ayuda a comprender la estructura del sistema, organizar el desarrollo del mismo, así como fomentar la reutilización y tomar en cuenta las posibles evoluciones del sistema.

2.3.1. Modelo – Vista – Controlador (MVC)

Reconocido como estilo arquitectónico por Taylor y Medvidovic (RICHARD TAYLOR 1995), el MVC ha sido propio de las aplicaciones en Smalltalk³ por lo menos desde 1992, antes que se generalizaran las arquitecturas en capas múltiples.

En ocasiones se le define más bien como un patrón de diseño o como práctica recurrente, y en estos términos es referido en el marco de la estrategia arquitectónica de Microsoft. En otras bibliografías es tratado a veces en términos de un estilo decididamente abstracto o como patrón de aplicación ligado a una implementación específica. Buschmann y otros lo consideran un patrón correspondiente al estilo de los sistemas interactivos o de llamada y retorno (Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad y Michael Stal 1996).

MVC sugiere la separación del software en tres componentes: Modelo, Vista y Controlador, los cuales serán explicados brevemente:

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí lo constituyen los datos puros y la lógica de los propios datos que puestos en el contexto del sistema proveen de información al usuario y en algunos casos a la propia aplicación.

³ Sistema informático que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación de escritorio, la “Vista” sería un formulario o componente visual con contenido sobre la cual el usuario puede realizar sus operaciones.

Controlador: Es la parte encargada de manejar y responder las solicitudes del usuario, procesando toda la información necesaria y modificando el Modelo en caso de ser necesario.

El ciclo de vida del patrón básicamente empieza cuando el usuario hace una solicitud al controlador con información sobre lo que desea realizar. El controlador decide a quien debe delegar la tarea y es aquí donde el modelo empieza su trabajo.

En esta etapa el modelo se encarga de realizar operaciones sobre la información que maneja para cumplir con lo que le solicita el controlador. Una vez que termina su labor, le regresa al controlador la información resultante de sus operaciones, el cual a su vez dirige a la vista. La vista se encarga de transformar los datos en información visualmente entendible al usuario.

Este modelo de arquitectura presenta varias ventajas:

- ❖ La separación del modelo de la vista, es decir, separar los datos de la representación visual de los mismos.
- ❖ Agregar o quitar una funcionalidad no afecta a la aplicación.
- ❖ La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.
- ❖ Es posible crear más de una vista para el mismo modelo.
- ❖ Facilita el mantenimiento en caso de errores.
- ❖ Permite el escalamiento de la aplicación en caso de ser requerido. (LÓPEZ 2008)

2.4. Lenguajes de programación

La Real Academia Española define el término lenguaje como el “conjunto de sonidos articulados con que el hombre manifiesta lo que piensa o siente” o al “conjunto de

señales que dan a entender algo” y se define para el ámbito informático como el “conjunto de signos y reglas que permite la comunicación con un ordenador.” (RAE).

Un lenguaje de programación se puede definir como un idioma artificial diseñado para expresar tareas a realizar por máquinas como las computadoras, que puede usarse para crear programas que controlen el comportamiento físico y lógico de la máquina, para expresar algoritmos con precisión, entre otros.

Los lenguajes de programación son las herramientas necesarias que permiten la creación de los programas informáticos; están formados por un conjunto de símbolos y reglas que definen su estructura y el significado de sus elementos y expresiones.

Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos necesarios para la creación de los programas informáticos de una manera adecuada a la capacidad cognitiva humana en lugar de a la capacidad ejecutora de las máquinas, es decir, facilita la comunicación con un computador mediante signos convencionales cercanos a los de un lenguaje natural.

2.4.1. Java

Java es un lenguaje de programación de alto nivel y orientado a objetos desarrollado por Sun Microsystems⁴ a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++⁵, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir muchos errores.

Entre noviembre del 2006 y mayo del 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL⁶, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar las aplicaciones Java todavía no es software libre).

⁴ Empresa informática de Silicon Valley, fabricante de semiconductores y software. Las siglas SUN se derivan de Stanford University Network.

⁵ Tanto C como C++ constituyen lenguajes de programación.

⁶ GNU General Public License. En idioma español, Licencia Pública General de GNU.

Dentro de las principales características del lenguaje se encuentran: (Rumbaugh, J. Jacobson, I. 2000, 2000)

- ❖ Java es un lenguaje de programación orientado a objetos, por lo que se enfoca en los datos (objetos) y en la manera de llegar a ellos (interfaces), no en las herramientas que se utilizan para manejarlos.
- ❖ Como las aplicaciones de Java se compilan en un formato de bytecode⁷ de arquitectura neutral, se pueden ejecutar en cualquier sistema, siempre y cuando este sistema instrumente la máquina virtual de Java.
- ❖ Java es un lenguaje intérprete. El compilador de Java genera bytecode para la máquina virtual de Java (JVM - Java Virtual Machine), en vez de código nativo de máquina.
- ❖ Java es robusto, puesto que no permite el manejo directo de memoria.
- ❖ Los programas escritos en Java no pueden ser atacados por virus, pues para que estos tengan efecto deben utilizar rutinas de acceso directo a memoria, que Java no tiene.
- ❖ Los diseñadores de Java intentaron crear un lenguaje que el programador pudiera aprender con rapidez.

2.4.2. Biblioteca gráfica Swing

Desde sus inicios el entorno Java cuenta con una biblioteca de componentes gráficos conocida como AWT⁸, la cual está concebida como una API⁹ estandarizada que permite utilizar los componentes nativos de cada sistema operativo; por lo que se puede decir que AWT es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.

Para que los programadores de AWT puedan mantener la consistencia gráfica entre los distintos sistemas operativos, estos se encuentran confinados a un mínimo denominador común entre ellos, es decir que sólo se disponen en AWT de las funcionalidades comunes en todos los sistemas operativos.

⁷ Código intermedio entre el código fuente y el código máquina.

⁸ Kit de Herramientas de Ventana Abstracta (del inglés Abstract Window Toolkit)

⁹ Interfaz de Programación de Aplicaciones (del inglés Application Programming Interface)

Para solventar este problema, en 1997, Sun Microsystems y Netscape Communications Corporation anunciaron la intención de combinar distintas tecnologías Java, cuyo resultado fue la biblioteca gráfica Swing.

Swing es una biblioteca gráfica para Java que incluye componentes de interfaz gráfica de usuario tales como cajas de texto, botones, componentes desplegados y tablas. Implementa una arquitectura Modelo – Vista – Controlador. Entre sus características fundamentales se encuentra:

- ❖ Permite que el aspecto de cada componente de una aplicación pueda cambiar sin introducir cambios sustanciales en el código de la aplicación.
- ❖ Independencia de plataforma.
- ❖ Altamente particionada debido a que los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto.

Se hace necesario destacar que la librería Swing no reemplaza completamente a AWT, sino que utiliza las características, componentes y ventajas de este último para incorporar nuevos componentes que facilitan la creación de aplicaciones gráficas en Java.

2.5. Sistemas Gestores de Base de Datos

En la actualidad un porcentaje elevado de las tareas que se realizan en un ordenador, computadora o sistema informático requiere de la preservación de cierta cantidad de datos con el objetivo de que perduren en el tiempo, lo cual se ha convertido en una necesidad de vital importancia para la mayoría de las organizaciones e instituciones. Es por esto que en la actualidad han surgido técnicas y tecnologías que aportan los mecanismos necesarios a las aplicaciones informáticas para que cumplan con este objetivo.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Este término "...no incluye la aplicación informática que consiste en los formularios y reportes con que el usuario interactúa" (Riordan, Rebecca, 1999). Las aplicaciones más usuales de una

base de datos son realizadas para la gestión de empresas e instituciones públicas; también son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Un Sistema Gestor de Base de Datos (en lo adelante SGBD) es un conjunto de programas informáticos que permiten crear y mantener una base de datos, asegurando la integridad, confidencialidad y seguridad de los datos que posteriormente se convertirán en información relevante para una organización. Entre sus principales funciones se encuentra el de centralizar los accesos a los datos y actuar como intermediario entre los datos físicos y el usuario.

Un SGBD tiene los siguientes objetivos específicos: (C. G. Hernández 2006)

- ❖ Independencia de los datos y los programas de aplicación.
- ❖ Minimización de la redundancia.
- ❖ Integración y sincronización de las bases de datos.
- ❖ Integridad de los datos.
- ❖ Seguridad y protección de los datos.
- ❖ Facilidad de manipulación de los datos.
- ❖ Control centralizado.

Entre los SGBD comúnmente utilizados en el mundo se tienen Oracle, MySQL, Microsoft SQL Server, PostgreSQL, entre otros.

2.5.1. HSQL

HSQLDB por sus siglas en inglés, Hyperthreaded Structured Query Language Database, es un sistema gestor de bases de datos libre. Sus cimientos están basados en HypersonicSQL, un proyecto de Thomas Mueller, y actualmente cuenta con 4 años de desarrollo, completamente escrito en el lenguaje de programación Java.

HSQLDB posee un pequeño pero rápido motor de base de datos que trabaja con tablas en memoria y tablas en disco. Actualmente ofrece dos modos o versiones, el modo embebido o incrustado y el modo servidor.

La utilización de HSQLDB se puede encontrar principalmente en proyectos de software libre tales como OpenOffice Base, productos comerciales como InstallAnywhere, entre otros. (8)

Entre las principales características de HSQLDB se encuentran:

- ❖ Es un SGDB escrito completamente en Java.
- ❖ Posee un completo motor de base de datos relacional.
- ❖ El tiempo de arranque del motor de base de datos es mínimo y las operaciones SQL tales como SELECT, INSERT, DELETE y UPDATE son ejecutadas a alta velocidad.
- ❖ Utiliza la sintaxis SQL estándar.
- ❖ Cumple con la integridad referencial (claves foráneas).
- ❖ Procedimientos almacenados en Java.
- ❖ Es posible la creación Triggers.
- ❖ Las tablas guardadas en disco son manejables a alta velocidad pudiendo llegar a tener capacidades de 256GB.

2.6. XML

XML (del inglés Extensible Markup Language) es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML.

XML simplifica el movimiento de datos entre sistemas diferentes, sus capacidades de "metadatos" y "metalenguaje" facilita el despliegue dentro de las aplicaciones complejas. Con él se pueden transportar enormes cantidades de datos bien organizados juntos con descriptores que indican los elementos y estructura de datos que contiene el documento. Sirve para estructurar, almacenar e intercambiar información. No ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.

Como norma en el XML se establece que los tags deben ser "declarados" antes de que puedan ser utilizados para marcar los elementos de texto. Estas declaraciones suelen estar contenidas en DTD (del inglés Document Type Definition) separados en los documentos de datos, estos ficheros establecen los vocabularios que especifican cómo se debe utilizar cada tags. Además cuando se usa XML como lenguaje de metadatos, los DTD pueden funcionar como diccionarios de tipos de datos. (García 2006)

Entre las ventajas de utilizar XML se encuentran:

- ❖ Es un estándar basado en un conjunto de reglas para la definición de etiquetas semánticas que organizan los documentos en diferentes secciones.
- ❖ Es una arquitectura más abierta y extensible. Los identificadores pueden crearse de manera sencilla y ser adaptados.
- ❖ La codificación del contenido en XML consigue que la estructura de la información resulte más accesible. Además, la independencia entre el contenido de los datos y la presentación de los mismos hace de XML un formato adecuado para el desarrollo de un sistema.
- ❖ Existen abundantes herramientas para trabajar con información representada con el formato XML. Estas herramientas facilitan el acceso, tratamiento y transformación de los datos.

2.7. Metodologías para el desarrollo del software

En la actualidad la construcción de una aplicación informática eficiente que cumpla con los requerimientos planteados es una tarea intensa y difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente.

Una metodología de desarrollo del software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Hoy en día existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo y seleccionar la adecuada que posibilite obtener los resultados óptimos es uno de los principales problemas a que se enfrentan los equipos de desarrollo.

2.7.1. El proceso Unificado de desarrollo de Software (RUP).

El Proceso Unificado de Desarrollo (conocido como RUP por sus siglas en inglés) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo. Es, además, una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante el ciclo de vida del software, con el objetivo de abarcar tanto pequeños como grandes proyectos. (KRUCHTEN 2001)

RUP posee tres características esenciales: (Jacobson I, Booch G, Rumbaugh J, 2000)

- ❖ Dirigido por casos de uso: Los casos de usos representan el hilo conductor que orienta las actividades de desarrollo. Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades del usuario.
- ❖ Centrado en la arquitectura: Establecer una arquitectura candidata al inicio es un paso muy importante, pues será ella la que guíe el desarrollo del sistema. Esto permitirá el desarrollo en paralelo, minimizar la repetición de trabajos e incrementar la probabilidad de reutilización de componentes.
- ❖ Iterativo e incremental: El desarrollo iterativo brinda la posibilidad de que los elementos sean integrados progresivamente, facilita el rehúso y resulta un producto más robusto pues los errores se van corrigiendo en cada iteración.

Las nociones de casos de uso y de Escenarios utilizadas en RUP han demostrado ser una manera excelente de capturar los requerimientos funcionales y asegurarse que direccionan el diseño, la implementación y la prueba del sistema, logrando así que el sistema satisfaga las necesidades del usuario. Una característica a destacar de RUP es que describe como controlar, rastrear y monitorear los cambios, en ambientes en los cuales el cambio es inevitable, para permitir un desarrollo iterativo exitoso.

2.8. El Lenguaje Unificado de Modelado (UML)

La falta de estandarización en la forma de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema.

UML (del inglés Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

UML ha nacido como un lenguaje, pero es mucho más que un lenguaje de programación. En realidad se ha diseñado y construido un lenguaje que ha nacido con una madurez sólida si se le compara, incluso con los últimos desarrollos de HTML, C#, Java, Ajax, XML, los lenguajes por excelencia del mundo de la Internet. (Larman 1999).

UML ayuda a los usuarios a entender la realidad desde un punto de vista de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades de dinero en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de los módulos que construirán el software. (RUMBAUHG 2000)

Entre los rasgos principales que han contribuido a hacer de UML el estándar de la industria en la actualidad, se puede mencionar: (Jacobson. I, Booch. G, Rumbaugh. J, 2000)

- ❖ Permite modelar sistemas utilizando técnicas orientadas a objetos.
- ❖ Adecuado a las necesidades de conectividades actuales y futuras.
- ❖ Es un lenguaje muy expresivo que cubre las vistas necesarias para desarrollar y luego desplegar los sistemas.
- ❖ Ampliamente utilizado por la industria del software.
- ❖ Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- ❖ Modela estructuras complejas.

- ❖ Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración, que sirve para evaluar el estado de las máquinas.

2.9. Visual Paradigm como herramienta CASE

Las herramientas CASE (Ingeniería de Software Asistida por Ordenador del inglés Computer Aided Software Engineering) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Entre sus principales objetivos se encuentra:

- ❖ Mejorar la productividad en el proceso de desarrollo del software.
- ❖ Aumentar el nivel de calidad del sistema construido.
- ❖ Reducir el tiempo y coste de desarrollo.
- ❖ Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- ❖ Ayudar a la reutilización del software, portabilidad y estandarización de la documentación.
- ❖ Gestionar de forma global todas las fases de desarrollo del software con una misma herramienta.
- ❖ Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Visual Paradigm es una herramienta CASE que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios. (Jacobson. I, Booch. G, Rumbaugh. J, 2000)

Emplea las últimas notaciones del Lenguaje Unificado de Modelado, ingeniería inversa, generación del código, exportación e importación de archivos XML, entre otros. Es una tecnología libre, multiplataforma, fácil de instalar y actualizar que se encuentra disponible en varios idiomas.

Las características gráficas y cómodas de Visual Paradigm facilitan la realización de los diagramas de modelados grandes y complejos que siguen el estándar de UML durante

el ciclo de vida de desarrollo del software. Su arquitectura persistente mejora en gran medida la velocidad y uso de memoria del entorno donde se ejecuta.

De forma general, esta herramienta ofrece:

- ❖ Entorno de creación de diagramas para UML 2.0
- ❖ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- ❖ Disponibilidad de múltiples versiones, para cada necesidad.
- ❖ Disponibilidad en múltiples plataformas.

2.10. Entornos de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado o IDE (del inglés Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación, que puede dedicarse a uno o varios lenguajes de programación y ha sido empaquetado como un programa de aplicación compuesto por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica de forma amigable.

2.10.1. NetBeans

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Su uso es gratuito y sin restricciones al ser un producto libre.

El IDE NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las API del NetBeans y un archivo especial (manifest file) que lo identifica como módulo.

A partir de la versión 6.0 se incluyen mejoras significativas y nuevas características, incluyendo una infraestructura de editor completamente reescrita, soporte para idiomas adicionales, incorpora nuevas características de productividad, y un proceso de instalación simplificado.

Entre las características del IDE se encuentran: (Sun Microsystems, 1994-2008)

- ❖ **Código abierto patrocinado por Sun:** NetBeans viene asimismo con la ventaja de haber sido creado, respaldado y convertido en código abierto por Sun Microsystems lo que implica que sea un entorno de desarrollo de clase empresarial con un soporte completo.
- ❖ **Integración de múltiples herramientas y protocolos:** Presenta extensas posibilidades para el desarrollo multiplataforma. Muchos desarrolladores están encontrando atractivo a NetBeans porque tiene soporte para Java Enterprise Edition (JEE), por su facilidad de uso, su cumplimiento con los estándares, y su extensibilidad a través de distintas plataformas; lo que posibilita que la creación de aplicaciones Java de tipo empresarial sea fácil y rápido.
- ❖ **El soporte al modelado mejora la productividad del desarrollador:** Posee soporte para UML. Este aspecto de NetBeans convierte el desarrollo de aplicaciones Java para múltiples plataformas en algo rápido y eficiente. El soporte bidireccional permite implementar y sincronizar rápidamente los modelos con los cambios en el código a medida que la aplicación avanza por los ciclos de desarrollo.

2.11. Conclusiones

Para la construcción de la herramienta propuesta se decide desarrollarla en la plataforma Java (Java 2 Edición Estándar – J2SE –) utilizando el IDE NetBeans 6.8, siguiendo la metodología RUP. Los diagramas serán modelados en la herramienta CASE Visual Paradigm. Los ficheros que se generen en la aplicación serán estructurados utilizando en formato XML.

CAPÍTULO 3 PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

3.1. Introducción

En el presente capítulo se describen los procesos de negocio, actores del negocio y trabajadores del negocio. Se expone, además, el Diagrama de casos de uso del Negocio y se detallan todos los casos de uso del negocio a través de sus descripciones textuales.

En un segundo momento se presentan los requisitos funcionales y no funcionales a tener en cuenta y finalmente se presenta la solución desde la perspectiva del sistema, se definen sus actores, se exhibe el Diagrama de casos de uso del Sistema y luego sus descripciones textuales.

3.2. Modelo del Negocio

El modelo de negocio describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización. Tiene como objetivo comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprende los problemas actuales de la organización e identifica las mejoras potenciales. Asegura que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y sepan derivar los requisitos del sistema que va a soportar la misma.

3.2.1. Actores y trabajadores del negocio

Un actor del negocio representa un individuo, grupo, organización, máquina o sistema que interactúa con el negocio y se beneficia de su existencia.

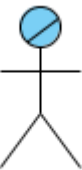
Actores del Negocio	Justificación
 Director de Hidrocarburos	Coordina y dirige el proceso de confección del INRRPG. Beneficiario directo de los resultados obtenidos a partir de las solicitudes de los informes de recursos y reservas.

Tabla 1: Actores del Negocio

Los trabajadores del negocio representan personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtienen ningún resultado de valor.



Trabajadores del Negocio	Justificación
 <p>Contratista</p>	Participa en el proceso de Gestionar informe de datos primarios de Prospecto, Gestionar Informe de datos primarios de Yacimiento, Gestionar datos primarios de reserva calculada, Gestionar datos primarios de Producción.
 <p>Operador del Inventario</p>	Participa en la confección del Inventario Nacional de Recursos y Reservas de Petróleo y Gas, apoyándose en la información entregada por los Contratistas.

Tabla 2: Trabajadores del Negocio

La siguiente figura representa el Modelo de Objetos del Negocio.

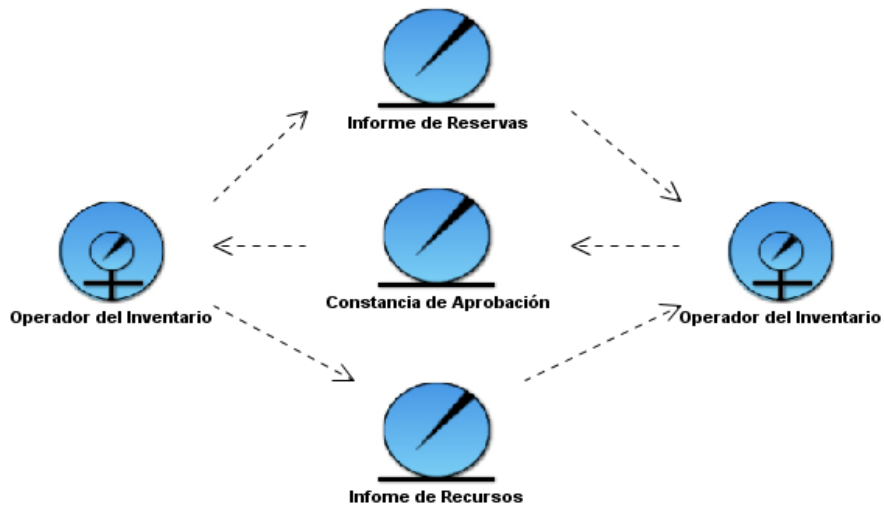


Figura 1: Modelo de Objetos del Negocio

3.2.2. Procesos del Negocio

El proceso de negocio comienza cuando la ONRM emite, a las entidades y contratistas, la orden de entregar la información relacionada con el movimiento de los recursos y reservas ocurridos durante el año calendario en operaciones.

Una vez emitida la solicitud, las entidades y contratistas deben confeccionar los informes de Recursos y los informes de Reservas y entregarlos antes del 1 de marzo del año en curso.

La ONRM tendrá 30 días, hasta el 1 de abril, para revisar y evaluar la información entregada; pudiendo realizar observaciones en caso de encontrar errores. Particularmente para los datos primarios de prospectos o de yacimientos se hace necesario validar los parámetros de cálculo con el objetivo de asegurar la veracidad de los datos.

Las entidades y contratistas tienen hasta el 25 de abril para corregir los datos incorrectos y entregar el documento final sin errores. La ONRM controlará que se realicen las correcciones de acuerdo a sus observaciones en el plazo establecido para ello.

A partir de ese momento, la ONRM tendrá hasta el 1 de mayo para aprobar la información entregada y notificar la validez de la misma a las entidades y contratistas. Una vez concluido lo anterior se procede a registrar los datos desde los documentos presentados por los contratistas hacia el sistema que se utiliza en esta área de la ONRM, a saber, el Sistema de Gestión de Datos Geológicos.

Luego de introducidos los datos, la propia aplicación genera, finalmente, el 30 de junio del mismo año con dicha información, el Inventario Nacional de Recursos y Reservas de Petróleo y Gas.

3.2.3. Diagrama de casos de uso del Negocio

El diagrama de casos de uso del Negocio describe las relaciones que existe entre un caso de uso de Negocio y un Actor del Negocio.

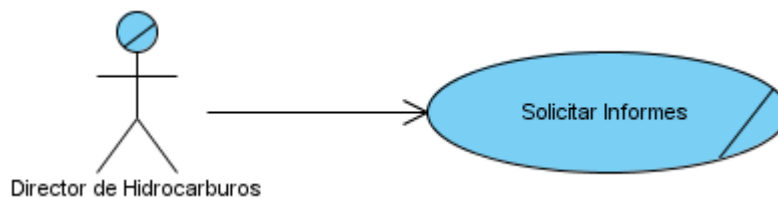


Figura 2: Diagrama de casos de uso del Negocio

3.2.4. Diagrama de Actividades

Los diagrama de actividades ayudan a describir en detalle lo que pasa dentro del negocio y permite examinar los roles específicos que juegan las personas (trabajadores del negocio) y las actividades que realizan, también identifica qué funciones deberá asumir el producto del software y quiénes serán los actores del futuro sistema.

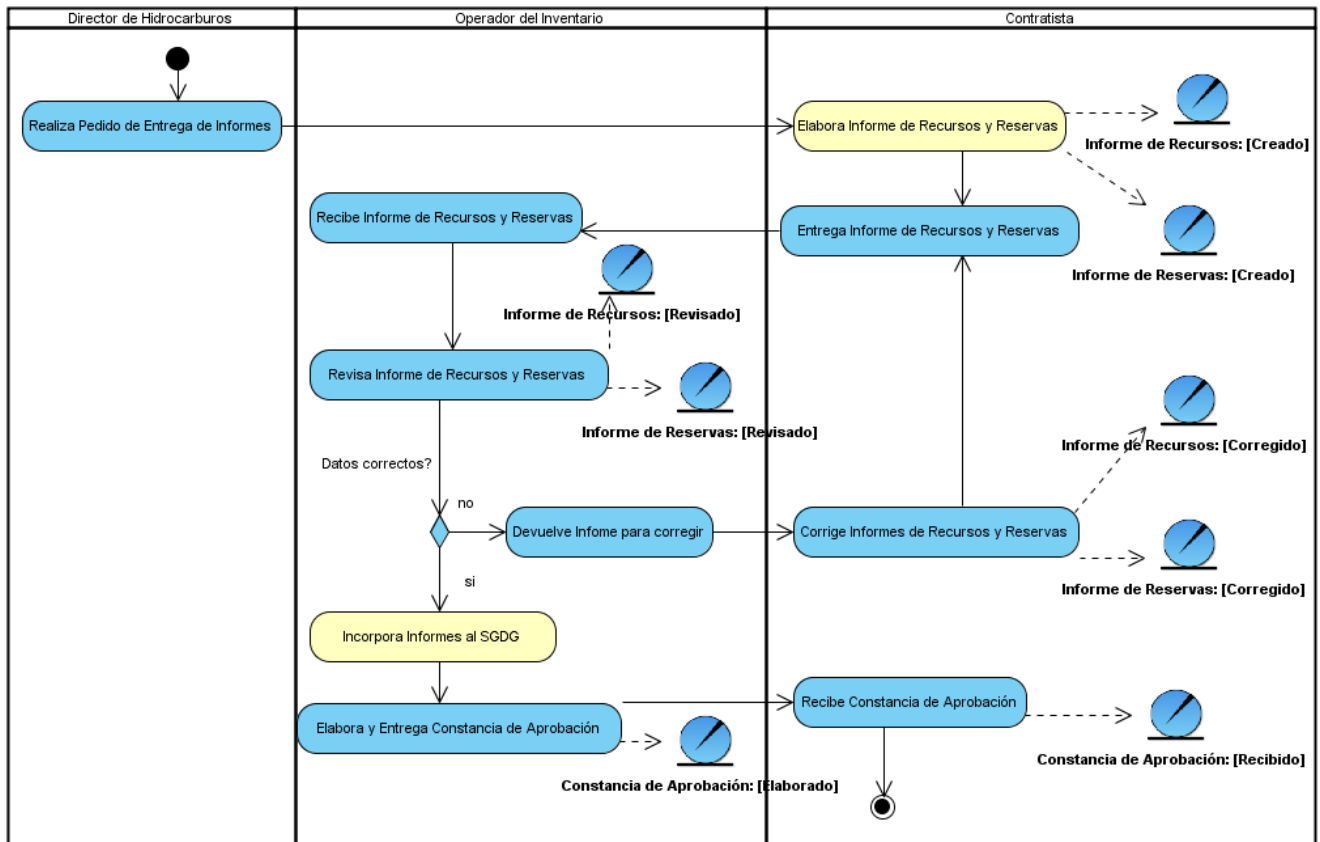


Figura 3: Diagrama de Actividad del caso de uso del Negocio Solicitar Informe

3.2.5. Descripción textual de los casos de uso del Negocio

A continuación se muestra la descripción textual del caso de uso del Negocio Solicitar Informe.

Caso de Uso	Solicitar Informe
Actores	Director de Hidrocarburos

Trabajadores	Operador del Inventario, Contratista	
Resumen	El caso de uso comienza cuando el Director de Hidrocarburos solicita la información necesaria a los contratistas para elaborar el Inventario Nacional de Recursos y Reservas de Petróleo y Gas; y termina el caso de uso cuando es aprobada la información.	
Precondiciones	Ninguna.	
Curso Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Presenta solicitud de entregar los informes de recursos y reservas.	2. Los contratistas confeccionan los informes de recursos y los informes de reservas.	
	3. Los contratistas entregan los informes de recursos y reservas elaborados a la Dirección de Hidrocarburo.	
	4. Se revisa la información entregada por los contratistas.	
	5. Se incorpora la información entregada por los contratistas al Sistema de Gestión de Datos Geológicos.	
	6. Se elabora y entrega una constancia de aprobación que refleja la correcta entrega de la información.	
	7. Los contratistas reciben la constancia de aprobación de los informes, finalizando el caso de uso.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
	5.1 En caso de error se devuelve a los contratistas los informes de recursos y los informes de reservas para su corrección; una vez corregidos los	

	errores, el caso de uso regresa a la acción 3.
Poscondiciones: Un informe es entregado, aprobado y registrado en el Sistema de Gestión de Datos Geológicos.	

Tabla 3: Descripción Textual del caso de uso del Negocio Solicitar Informe

3.3. Requisitos Funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Como requisitos funcionales de la solución propuesta se tienen:

R1. Gestionar datos primarios de Prospecto

- R1.1 Registrar datos primarios de Prospecto
- R1.2 Modificar datos primarios de Prospecto
- R1.3 Eliminar datos primarios de Prospecto

R2. Gestionar datos primarios de Yacimiento

- R.2.1 Registrar datos primarios de Yacimiento
- R.2.2 Modificar datos primarios de Yacimiento
- R.2.3 Eliminar datos primarios de Yacimiento

R3. Gestionar datos de reservas calculadas

- R3.1 Registrar datos de reservas calculadas
- R3.2 Modificar datos de reservas calculadas
- R3.3 Eliminar datos de reservas calculadas

R4. Gestionar datos de Producción

- R4.1 Registrar datos de Producción
- R4.2 Modificar datos de Producción
- R4.3 Eliminar datos de Producción

R.5. Importar Configuración

R.6. Generar Informe de Recursos y Reservas

R6.1 Imprimir Informe de Recursos y Reservas

R7. Exportar Informe de Recursos y Reservas a formato XML

R8. Importar Informe de Recursos y Reservas desde formato XML

3.4. Requisitos No Funcionales

Los requisitos no funcionales representan propiedades o cualidades que el producto debe tener lo que hace del mismo que sea más atractivo, más usable, rápido y más confiable.

Requerimientos de usabilidad

1. El sistema debe poder ser usado por cualquier persona que posea conocimientos básicos de computación (trabajo con ventanas, trabajo con menú, trabajo con el mouse).
2. El sistema debe poseer una interfaz entendible y amigable, de fácil acceso y manipulación y que facilite la localización de las diferentes funcionalidades presentes en ella.
3. La información deberá estar disponible en todo momento.

Apariencia o interfaz externa

1. El sistema debe tener una apariencia profesional con un diseño gráfico sencillo de tonalidades de colores claros que facilite la localización de las funciones del sistema.

Requerimientos de portabilidad y operatividad.

1. La aplicación debe ser compatible con los Sistemas Operativos Microsoft Windows 2000 NT, Microsoft Windows XP y cualquier distribución de GNU/Linux.

Requerimientos de software del sistema.

1. Las computadoras que utilizarán el software deben tener instalado:
 - a) Microsoft Windows 2000 NT, Microsoft Windows XP Profesional o cualquier distribución de GNU/Linux.
 - b) Máquina Virtual de Java (JVM) en su versión 1.6 o superior.

Requerimientos de hardware del sistema.


1. Las computadoras que utilizarán la aplicación deberán contar con un microprocesador con velocidad de procesamiento superior a un 1 GHz¹⁰ y memoria RAM¹¹ de 256 MB o superior.

3.5. Descripción del Sistema Propuesto

Un caso de uso constituye una técnica utilizada para describir el comportamiento del sistema, a través de un documento narrativo que define la secuencia de acciones que obtienen resultados de valor para un actor que utiliza un sistema para completar un proceso, sin importar los detalles de la implementación.

Los actores se definen como los roles que puede tener un usuario; pueden ser humanos, otros sistemas, máquinas, hardware, etc. que interactúan con un sistema para de esta forma intercambiar datos.

3.5.1. Descripción de los Actores

 <p>Contratista</p>	Interviene en el proceso de Gestionar los datos primarios de Prospectos, Gestionar los datos primarios de Yacimiento, Gestionar los datos primarios de reservas calculadas, Gestionar los datos primarios de Producción, exportar los Informes de Recursos y Reservas a ficheros en formato XML, Generar un informe con la información a entregar a la ONRM, Importar un informe desde ficheros en formato XML
--	--

¹⁰ Gigahercio: es un múltiplo de la unidad de medida de frecuencia hercio (Hz) y equivale a 109 Hz.

¹¹ Memoria de Acceso Aleatorio


	y finalmente cargar la configuración necesaria para que el sistema funcione.
 <p>Responsable Inventario</p>	Interviene en el proceso de confección del Inventario importando informes desde ficheros en formato XML y cargando la configuración necesaria para que el sistema funcione.

Tabla 4: Descripción de los Acores del Sistema

3.5.2. Diagrama de casos de uso del Sistema

A continuación se muestra el Diagrama de casos de uso del Sistema donde están representados los distintos actores que interactúan con el sistema mediante los del sistema y luego la descripción textual de cada uno de estos. Las imágenes correspondientes a los prototipos de interfaz de usuario se pueden encontrar en el Anexo III.

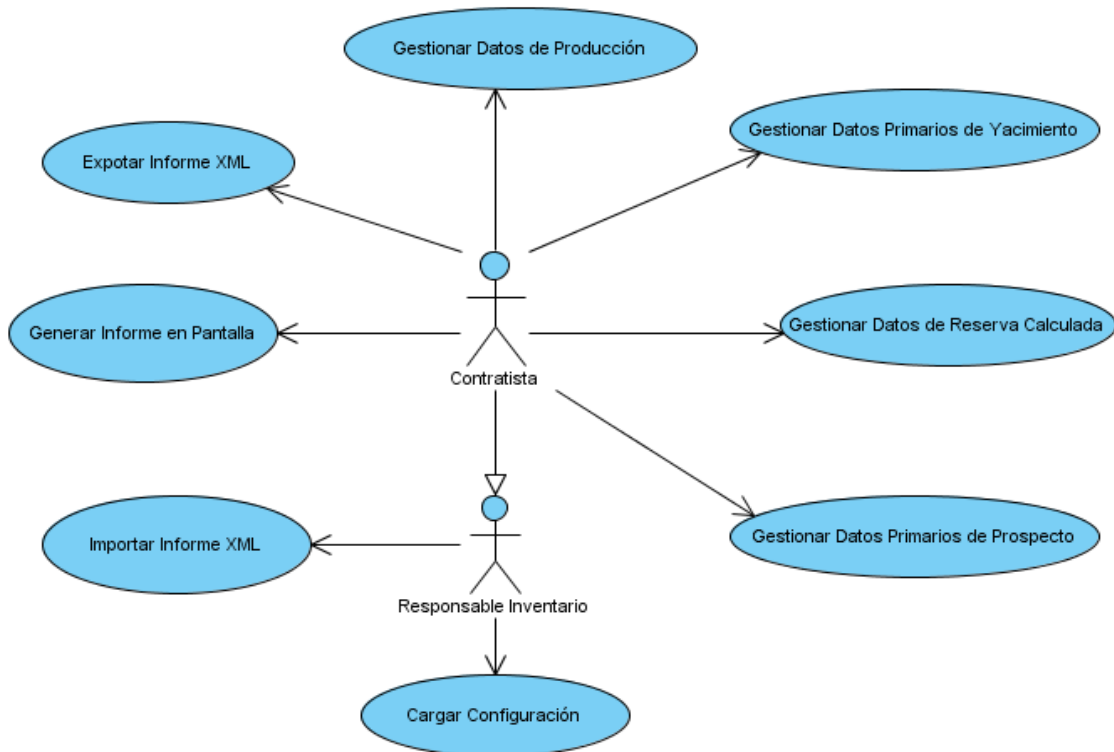


Figura 4: Diagrama de casos de uso del Sistema

3.5.3. Descripción textual de los casos de uso del Sistema

Caso de Uso	Gestionar datos primarios de Prospecto	
Actores	Usuario (Contratista)	
Propósito	Capacidad del usuario para gestionar los datos primarios de prospecto.	
Resumen	El caso de uso comienza cuando el usuario necesita gestionar los datos primarios de prospecto para confeccionar el Informe de Recursos y Reservas que entregará a la ONRM para la elaboración del INRRPG. El caso de uso termina cuando el usuario completa las operaciones a realizar (actualizar, eliminar, registrar) con los datos primarios del prospecto.	
Referencias	R1, R1.1, R1.2, R1.3	
Precondiciones	El sistema debe tener cargada la configuración.	
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El usuario selecciona del menú principal Gestión de datos la opción datos primarios de Prospecto o en la barra de herramientas la opción Prospecto.	2. El sistema muestra en pantalla un panel con la información necesaria para gestionar los datos primarios del prospecto.	
3. El usuario selecciona de la lista de prospecto mostrada un prospecto.	4. El sistema muestra en pantalla los datos asociados al prospecto seleccionado por el usuario y habilita las distintas opciones que permitirán gestionar los datos primarios del prospecto.	
5. De las opciones que muestra el sistema, el usuario selecciona una de ellas.	6. En dependencia de la opción seleccionada por el usuario, el sistema ejecuta una operación: <ul style="list-style-type: none"> ❖ El usuario selecciona la opción Eliminar datos primarios del Prospecto. [Ver sección Eliminar datos primarios del Prospecto] 	

	<ul style="list-style-type: none"> ❖ El usuario selecciona la opción Registrar Intervalo, se muestra la pantalla IUGestionarDPProspectoDato. [Ver sección Registrar Intervalo] ❖ El usuario selecciona la opción Editar Intervalo, seleccionando previamente el intervalo que desea modificar, se muestra la pantalla IUGestionarDPProspectoDato. [Ver sección Editar Intervalo]. ❖ El usuario selecciona la opción Eliminar Intervalo, seleccionando previamente el intervalo que desea eliminar [Ver sección Eliminar Intervalo] ❖ El usuario selecciona la opción Guardar Cambios, se guarda los datos primarios del prospecto seleccionado. [Ver sección Guardar Cambios] ❖ El usuario selecciona la opción Vista en Informe [Ver sección Vista en Informe]. ❖ El usuario selecciona la opción Cerrar, se cierra la pantalla que gestiona la información del prospecto seleccionado [Ver sección Cerrar].
Sección “Eliminar datos primarios del Prospecto”	
1. El usuario selecciona la opción Eliminar datos primarios del Prospecto.	2. El sistema muestra la alerta de confirmación “¿Deseas eliminar los

	datos del prospecto seleccionado?”, para asegurar que el usuario desea eliminar dicha información.
3. El usuario confirma su decisión de eliminar la información.	4. El sistema elimina los datos primarios del prospecto seleccionado. 5. El sistema muestra el mensaje “Datos eliminados”. 6. El sistema refleja los cambios en pantalla.
Cursos Alternos	3.1 El usuario cancela la decisión de eliminar la información. El sistema no elimina los datos primarios del prospecto seleccionado.
Sección “Registrar Intervalo”	
1. El usuario selecciona la opción Registrar Intervalo.	2. El sistema muestra la pantalla IUGestionarDPProspectoDato, con los campos necesarios para llenar los datos del intervalo a registrar.
3. El usuario completa todos los campo que se muestra en la pantalla y elije la opción Aceptar.	4. El sistema comprueba que todos los campos estén llenos y acorde a las restricciones. 5. El sistema guarda los datos del intervalo registrado, oculta la pantalla IUGestionarDPProspectoDato y refleja los cambios en el listado de intervalos en pantalla.
Cursos Alternos	3.1 El usuario selecciona la opción Cancelar y el sistema cierra la pantalla IUGestionarDPProspectoDato. 5.1 En caso de contener error los datos introducidos por el usuario, o faltar algún campo por llenar, el sistema muestra el mensaje “Error en los datos entrados.

	Por favor verifíquelos”, y resalta en rojo los campos que son incorrectos.
Sección “Eliminar Intervalo”	
1. De la tabla que contiene la lista de intervalos registrados, el usuario selecciona el(los) intervalo(s) que desea eliminar.	2. El sistema resalta la(s) fila(s) seleccionada(s) por el usuario de la tabla mostrada.
3. El usuario selecciona la opción Eliminar Intervalo para eliminar el(los) intervalo(s) seleccionado(s).	4. El sistema muestra la alerta de confirmación “¿Deseas eliminar los intervalos seleccionados?”, para asegurar que el usuario desea eliminar dicha información.
5. El usuario confirma su decisión de eliminar la información.	6. El sistema elimina el(los) intervalo(s) seleccionado(s) por el usuario. 7. El sistema actualiza la tabla que contiene el listado de intervalos disponibles.
Curso Alternos	3.1 El usuario cancela la decisión de eliminar la información. El sistema no elimina el(los) intervalo(s) seleccionado(s). 4.1 El sistema muestra la alerta “Seleccione los intervalos a eliminar”, en caso del usuario no haber seleccionado al menos un intervalo de la lista de intervalos disponibles.
Sección “Editar Intervalo”	
1. De la tabla que contiene la lista de intervalos registrados, el usuario selecciona el intervalo que desea modificar.	2. El sistema resalta la fila seleccionada por el usuario de la tabla mostrada.
3. El usuario selecciona la opción Editar Intervalo para modificar los datos del intervalo seleccionado.	4. El sistema muestra la pantalla, IUGestionarDPProspectoDato, en la cual aparece los campos con los datos del intervalo seleccionado por el usuario

	para que el mismo realice las modificaciones necesarias.
5. El usuario realiza los cambios necesarios y posteriormente selecciona la opción Aceptar.	6. El sistema comprueba que todos los campos estén llenos y acorde a las restricciones. 7. El sistema guarda los cambios realizados, oculta la pantalla IUGestionarDPProspectoDato y refleja los cambios en el listado de intervalos en pantalla.
Cursos Alternos	5.1 El usuario selecciona la opción Cancelar y el sistema cierra la pantalla IUGestionarDPYacimientoDatos. 7.1 En caso de contener error los datos introducidos por el usuario, o faltar algún campo por llenar, el sistema muestra el mensaje “Error en los datos entrados. Por favor verifíquelos”, y resalta en rojo los campos que son incorrectos.
Sección “Guardar Cambios”	
1. El usuario selecciona la opción Guardar Cambios para guardar los datos generales del Prospecto seleccionado.	2. El sistema comprueba que todos los campos estén llenos y acorde a las restricciones. 3. El sistema guarda los datos generales del Prospecto seleccionado y muestra el mensaje “Cambios guardados”.
Cursos Alternos	3.1 En caso de contener error los datos introducidos por el usuario, o faltar algún campo por llenar, el sistema muestra el mensaje “Error en los datos entrados. Por favor verifíquelos”, y resalta en rojo los campos que son incorrectos.
Sección “Vista en Informe”	

1. El usuario selecciona la opción Vista en Informe.	2. El sistema muestra en pantalla una página en forma de informe con la información del Prospecto seleccionado.
--	---

Sección "Cerrar"

1. El usuario selecciona la opción Cerrar.	2. El sistema no guarda los cambios pendientes en los datos generales, oculta el panel actual y finaliza el caso de uso.
--	--

Prioridad

Secundario

Interfaz

Poscondiciones	La información referente a los datos primarios del Prospecto ha sido gestionada (eliminada, registrada, modificada).
-----------------------	--

Tabla 5: Descripción del caso de uso Gestionar datos primarios de Prospecto

3.6. Conclusiones

En el presente capítulo se describió detalladamente el negocio el cual se modeló a través del Diagrama de casos de uso del Negocio y luego se precisaron las características del sistema, los requerimientos funcionales y no funcionales concluyendo que el software a desarrollar es de fácil utilización.

CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

4.1. Introducción

El presente capítulo describe la solución en términos de diagrama de clases del diseño dividido por secciones de casos de uso para facilitar una mejor comprensión, quedarán definidos los estándares de la interfaz de la aplicación y la concepción general de la ayuda, se presentará el diseño de la base de datos a través del diagrama Entidad-Relación y el diagrama de clases persistentes. Se define además, el modelo de despliegue originado por la selección de los artefactos más importantes para el sistema donde se precisan los componentes que conforman la estructura física de la aplicación.

4.2. Principios de Diseño

Definir principios de diseño en el desarrollo de una aplicación permite que la misma se convierta en una herramienta útil y atractiva para el usuario. Los principios de diseño no solo definen la apariencia estética de la interfaz, sino que posibilitan también que la misma brinde adecuadamente la información que le sea útil al usuario.

Como principios de diseño del sistema se tienen:

- ❖ Brindar una interfaz sencilla, de forma tal que cualquier persona con un mínimo dominio de la computación pueda trabajar con la aplicación si n presentar dificultades notables.
- ❖ Garantizar la legibilidad de manera que exista contraste entre los colores de los textos y el fondo y, por otra parte, el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- ❖ Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.
- ❖ Los mensajes que son mostrados al usuario deben ser precisos y de fácil comprensión.
- ❖ Los Menús y las etiquetas de botones deben comenzar con la palabra más importante.

4.2.1. Estándares de la Interfaz de la aplicación

Barra de Título

- ❖ El título debe ubicarse a la izquierda y los botones minimizar, maximizar y cerrar, ubicados del lado derecho.
- ❖ No deben usarse colores degradados u otros que interrumpan la visibilidad de los elementos anteriores.

Barra de Menú

- ❖ Los menús se ordenan según la importancia de las funcionalidades que brindan.
- ❖ El primer menú es “Archivo” y en él la última opción es “Salir”.
- ❖ El menú Ayuda está compuesto por dos elementos: “Temas de ayuda” y un cuadro de diálogo "Acerca de".
- ❖ Incluye atajos para los elementos del menú más frecuentemente usados.
- ❖ Para una mejor simplicidad de la barra de menú se emplean submenús y/o líneas divisorias para lograr un acceso rápido.

Punteros del Ratón

- ❖ Se usan los punteros estándares del sistema operativo en el cual se esté ejecutando la aplicación.

4.2.2. Concepción general de la ayuda

Temas de Ayuda

- ❖ Estructura de la aplicación: Se describe la estructura de la aplicación.
- ❖ Funcionalidades: Se explica de forma detallada cada una de las funcionalidades que brinda el sistema, a través del menú principal de la aplicación.
- ❖ Respuestas a Preguntas Frecuentes (FAQ).

Acerca de

- ❖ Se describen las características generales del sistema atendiendo a: versión en producción, desarrolladores, datos de contacto, año de elaboración y propietario.

4.3. Diagrama de clases del diseño

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas; muestran lo que el sistema puede hacer y cómo puede ser construido. Cuando se crea un diagrama de clases, se está modelando una parte de los elementos y relaciones que configuran la vista de diseño del sistema.

A continuación se muestra el diagrama de clases del diseño para el caso de uso Gestionar datos primarios de Prospecto. Los restantes diagramas de clases del diseño pueden ser consultados en el Anexo III.

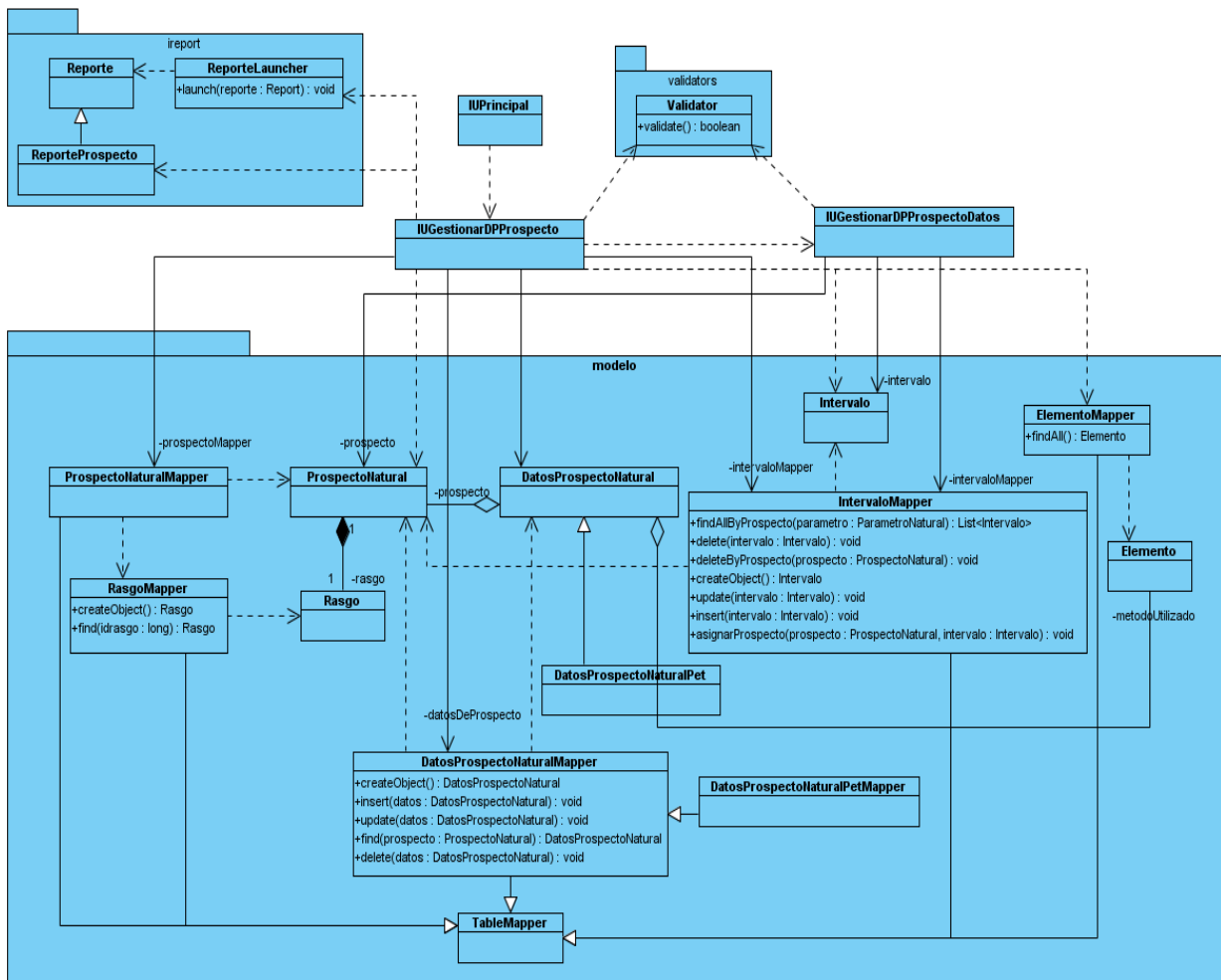


Figura 5: Diagrama de clases del CU Gestionar datos primarios de Prospecto.

4.4. Diseño de la base de datos

El primer paso para la construcción de una base de datos es definir su estructura, de forma tal que permita un adecuado mecanismo para almacenar los datos y posteriormente recuperarlos. Para lograr un buen diseño de la base de datos es necesario seguir un conjunto de pasos que comienzan con definir las clases persistentes, refinarlas y clasificarlas junto a sus atributos, lo que permitirá realizar el diagrama de clases persistentes y posteriormente la conversión de las clases al medio de almacenamiento. (Ver Anexo V).

4.4.1. Diagrama Entidad – Relación

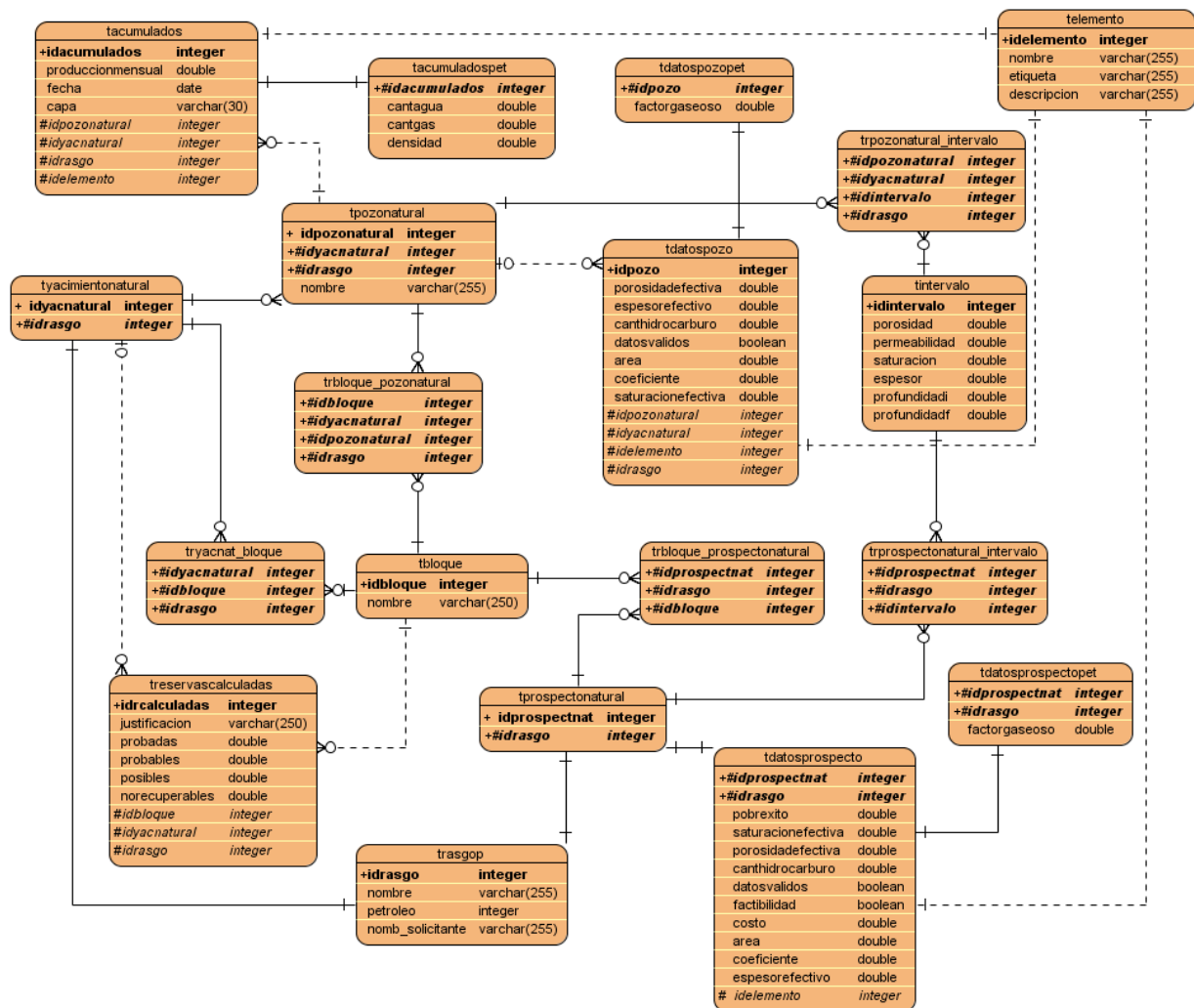


Figura 6: Diagrama Entidad – Relación

4.4.2. Diagrama de clases persistentes

Todas las clases identificadas durante el desarrollo del diseño no tienen que ser necesariamente persistentes, la persistencia de una clase está dada por la capacidad de la misma para mantener su valor en el espacio y en el tiempo; contrario a las clases temporales que son manejadas y almacenadas por el sistema en tiempo de ejecución, dejando de existir al finalizar el programa.

En la siguiente figura se muestra el diagrama de clases persistentes correspondiente al sistema planteado.

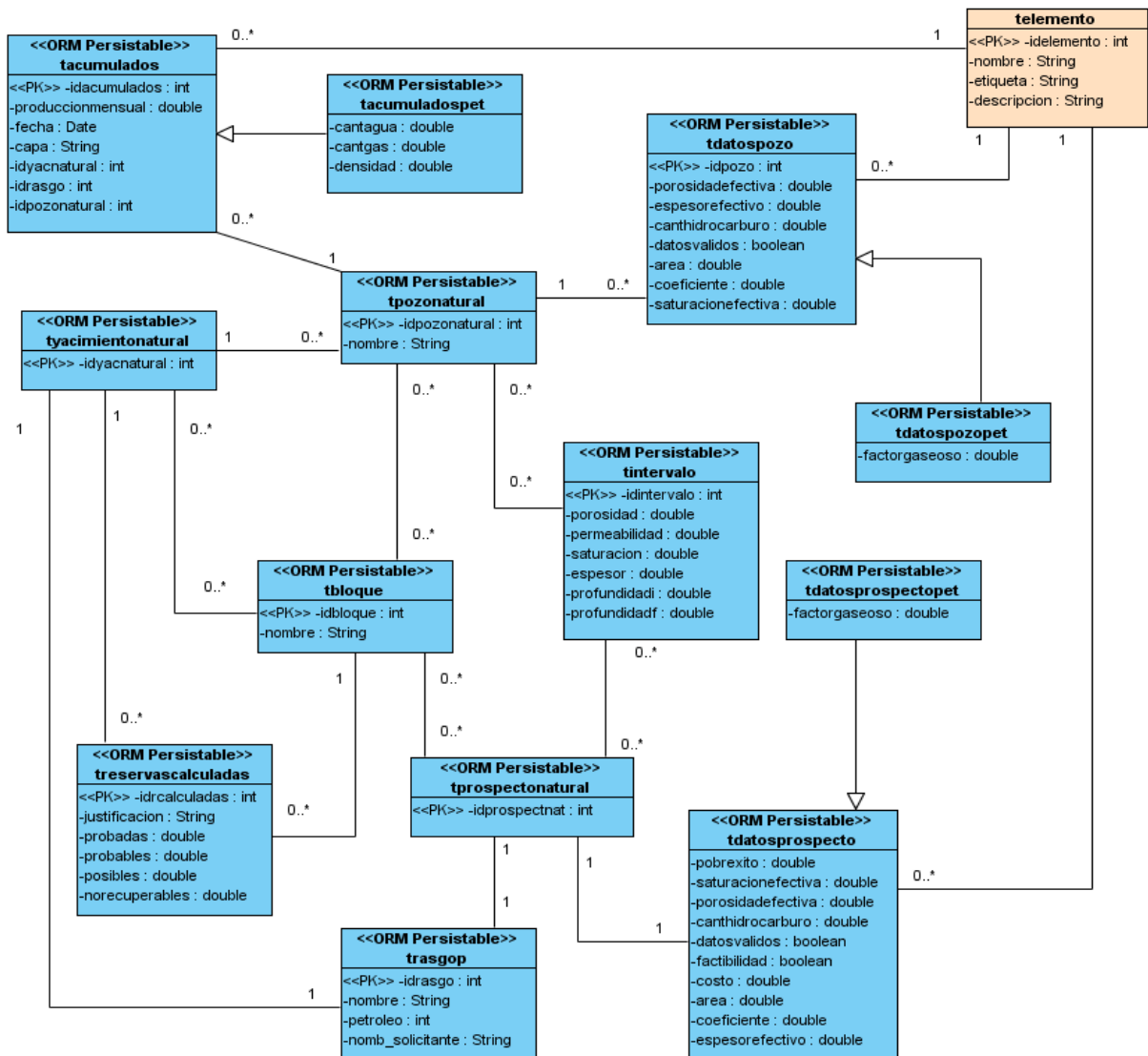


Figura 7: Diagrama de Clases Persistentes

4.5. Generalidades de la implementación

En la fase de construcción se implementa el sistema, es decir, se crea el código correspondiente al resultado de la fase de diseño, siguiendo los patrones y la arquitectura escogida.

El modelo de implementación permite planificar las integraciones de sistemas necesarias en cada iteración, distribuye el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue, implementa las clases y subsistemas encontrados durante el diseño y posibilita probar los componentes individualmente para después integrarlos.

4.5.1. Modelo de componentes

Los componentes son la parte modular del sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces con su realización. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos, éstos pueden ser un fichero de código fuente, *scripts*, ficheros de código binario, ejecutables, etc. Los diagramas de componentes son utilizados para modelar la vista estática del sistema, mostrando la organización y las dependencias lógicas entre los componentes. (Ver Anexo IV).

4.5.2. Modelo de despliegue

El Modelo de Despliegue describe la distribución física del sistema en términos de cómo las funcionalidades se distribuyen entre los nodos de computación sobre los que se va a instalar el sistema; mostrando las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

En la siguiente figura se muestra el modelo de despliegue para el sistema planteado.

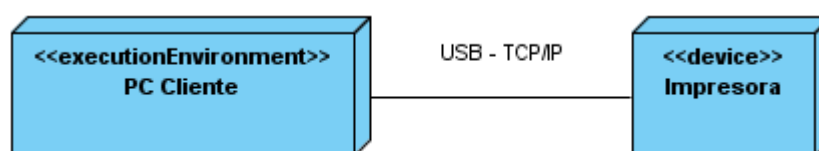


Figura 8: Modelo de Despliegue

4.6. Prueba del sistema propuesto

El desarrollo del software implica una serie de actividades en las que la posibilidad de que aparezcan errores humanos es muy común. Los errores pueden comenzar a manifestarse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta, y así en los posteriores pasos del diseño y desarrollo.

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación, estas tienen diferentes clasificaciones, algunas de ellas pueden ser:

- ❖ **Pruebas de Verificación:** Se comprueba el cumplimiento de las especificaciones del diseño.
- ❖ **Pruebas de Validación:** Se encargan de velar por el cumplimiento de los requisitos del análisis.
- ❖ **Pruebas de Caja Blanca:** Se conoce el código y se trata de ejecutar cada uno de los elementos del mismo.
- ❖ **Pruebas de Caja Negra:** Solamente se conoce la interfaz y se trata de probar cada uno de los elementos que componen a la misma.

Cualquier tipo de prueba tiene el objetivo principal de encontrar errores o defectos en el software.

Para que las pruebas aplicadas a un software tengan éxito es necesario realizar casos de pruebas con probabilidad de descubrir los errores en el sistema, a través de técnicas que guíen el proceso de la prueba.

4.6.1. Pruebas de caja negra

Para probar el sistema propuesto se utiliza la prueba de Caja Negra, la cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que la salida producida es correcta, ya que no

es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

Muchos autores consideran que estas pruebas permiten encontrar: [Pressman, 1998][MYE, 1979][Beizer,1995]

- ❖ Funciones incorrectas o ausentes.
- ❖ Errores de interfaz.
- ❖ Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- ❖ Errores de rendimiento.
- ❖ Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el sistema según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del sistema, a fin de verificar que el mismo funcione correctamente.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están: [Pressman, 2000]

- ❖ **Técnica de la Partición de Equivalencia:** esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del sistema.
- ❖ **Técnica del Análisis de Valores Límites:** esta técnica prueba la habilidad del sistema para manejar datos que se encuentran en los límites aceptables.
- ❖ **Técnica de Grafos de Causa-Efecto:** es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el sistema, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.

La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. [Pressman, 1997]

Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

- ❖ Si un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.
- ❖ Si una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango.
- ❖ Si una entrada requiere un valor presente en un conjunto de valores, aparecen 2 clases de equivalencia: en el conjunto o fuera de él.
- ❖ Si una entrada es booleana, hay 2 clases: si o no.

Aplicando estas directrices se ejecutan casos de pruebas para cada elemento de datos del campo de entrada a desarrollar, donde los casos de prueba se seleccionan de forma que ejerciten el mayor número de atributos de cada clase de equivalencia a la vez.

A continuación se muestra los casos de pruebas para el caso de uso “Exportar Informe XML” utilizando la técnica de Partición de Equivalencia. (Ver Anexo VI para los restantes casos de prueba de los diferentes casos de uso).

4.6.2. Caso de uso: Exportar Informe XML.

Descripción general del caso de uso

El caso de uso comienza cuando el usuario necesita enviar a la ONRM el informe de los recursos y las reservas; y termina cuando el sistema genera el informe en formato XML.

Condiciones de Ejecución

El sistema debe tener cargada la configuración.

Secciones a probar en el caso de uso

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC1. Exportar Informe XML	<p>EC 1.1. Seleccionar Exportar</p>	<p>Al acceder a la opción “Exportar Informe XML” del menú Archivo o a través de la barra de herramientas, se muestra la pantalla “Exportar Informe a formato XML”. Esta ventana permite exportar (guardar) la información de los recursos y reservas a un archivo XML. Se muestra la opción de seleccionar el archivo destino, así como la información a guardar en el mismo. Adicionalmente se tiene el botón “Exportar Informe” y “Cancelar”.</p>	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.
	<p>EC 1.2. Guardar En</p>	<p>Al seleccionar el botón “Guardar En” se muestra un explorador de archivos y carpetas que permite seleccionar el archivo XML dónde se guardará la información de los recursos y las reservas. Una vez seleccionado el archivo su ruta completa aparecerá en el campo “Dirección”.</p>	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.
	<p>EC 1.3. Cancelar Guardar En.</p>	<p>Al seleccionar el botón “Cancelar” del explorador de archivo que aparece cuando se selecciona el botón “Guardar En”, se cierra el explorador de archivo para lo cual el campo “Dirección” no cambia su valor.</p>	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.

			<p>XML.</p> <ul style="list-style-type: none"> ❖ Guardar En.
	EC 1.4. Exportar Informe.	Luego de completarse todos los datos requeridos para exportar la información de los recursos y reservas al archivo XML, se selecciona el botón “Exportar Informe” y se genera un archivo XML con la información en la dirección seleccionada. Luego de esto se muestra el mensaje de información “Se guardó el Informe XML de forma correcta” para indicar el éxito de la operación.	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.
	EC 1.5. Cancelar operación.	Al seleccionar el botón “Cancelar”, se debe cerrar la ventana “Exportar Informe a formato XML”, cancelando la operación.	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.
	E.C 1.6. Datos incorrectos o Incompletos.	Al seleccionar el botón “Exportar Informe”, se comprueba que los datos entrados sean correctos y estén completos o acordes, en caso de error, se resalta en rojo los campos incorrectos y se muestra el correspondiente mensaje de alerta. (“El archivo seleccionado no es correcto” para el caso del campo dirección, y “Datos incorrectos” cuando hay datos incompletos o incorrectos).	<ul style="list-style-type: none"> ❖ Ventana Principal. ❖ Menú Archivo. ❖ Exportar Informe a XML.

Tabla 6: Secciones a probar en el caso de uso Exportar Informe XML

Descripción de las variables

No.	Nombre del campo	Clasificación	Requerido	Descripción
1	Dirección	Campo de Texto	Si	Ruta al archivo XML donde se guardará la información.
2	Todo	Radio Button	Alternado	Opción que indica que se exporta toda la información. (Alternado con la opción Custom)
3	Custom	Radio Button	Alternado	Opción que indica que se exporta parte de la información (Alternado con la opción Todo)
4	Contenido	Casilla de selección	Opcional	Indica el contenido a exportar en el archivo XML. (Solo se selecciona en caso de seleccionarse la opción Custom)

Tabla 7: Descripción de las variables para el caso de uso Exportar Informe XML

Matriz de datos (SC 1. Exportar Informe XML)

Escenario	Dirección	Todo	Custom	Contenido	Respuesta del Sistema	Resultado de la prueba
EC 1.1. Seleccionar Exportar	N/A	N/A	N/A	N/A	Al acceder a la opción "Exportar Informe XML" se muestra la pantalla "Exportar Informe a formato XML" con	

					todos los campos necesarios.	
EC 1.2. Guardar En	V	N/A	N/A	N/A		

Tabla 8: Matriz de Datos. SC1. Exportar Informe XML

4.7. Conclusiones

En el presente capítulo se arribó a un sistema completamente diseñado y construido en términos de clases del diseño. Se generaron además cada uno de los artefactos y diagramas referentes al flujo de trabajo de implementación culminando la modelación completa de la solución propuesta.

CONCLUSIONES

Una vez concluida la investigación y luego de obtenidos los resultados, es posible resaltar una serie de conclusiones que se enumeran a continuación:

1. Las tecnologías, herramientas, lenguajes y framework seleccionados para el desarrollo de la aplicación que centra la investigación obedecen a criterios de selección de tecnologías libres y multiplataforma, de acuerdo con las políticas que impulsa la universidad y el país en sentido general.
2. La puesta en explotación de la aplicación desarrollada proporcionará mayor rendimiento al Sistema de Gestión de Datos Geológicos y el procesamiento de los datos resultará más fidedigno debido a la reducción de errores por actividad humana, por otra parte, agilizará el tiempo de entrega y la manipulación de los datos obtenidos de los contratistas.
3. El resultado de la investigación cobra doblemente valor al considerar la no existencia de soluciones semejantes a la desarrollada.
4. El diseño y la arquitectura de información de la aplicación se presentan de manera sencilla, fácil de usar, intuitiva y con balance visual de colores que proporcionan un entorno agradable al usuario final.
5. Todos los requisitos funcionales y no funcionales capturados en el momento correspondiente fueron debidamente implementados.
6. El sistema cuenta con una ayuda general que permite obtener información de cada una de las acciones que puede realizar.
7. El diseño de las pruebas de caja negra permitirá en un momento posterior validar los requisitos funcionales capturados y comprobar que el sistema realiza las acciones que debe realizar en el momento y el lugar en el que las debe realizar.

El autor de la presente investigación recomienda:

1. Establecer un período de capacitación para cada usuario nuevo que deba utilizar el sistema para el desarrollo de sus actividades.
2. Continuar la identificación de requisitos funcionales en aras de ampliar las posibilidades de trabajo con la herramienta desarrollada.
3. Implementar las pruebas diseñadas y documentar los defectos en función de lograr un producto más robusto y libre de errores.

REFERENCIAS BIBLIOGRÁFICAS

1. **RANGEL.** (2006). "Nuevas Tecnologías de la Información y la Comunicación (NTIC) Ideas para su categorización." [En línea]. Disponible en: <http://anabelcarylásticos.blogspot.com/>.
2. **ASTROMÍA.** Geología, 2005. [2008]. [En línea]. Disponible en: <http://www.astromia.com/glosario/geologia.htm>
3. **ORNM.** (2008). "Oficina Nacional de Recursos Minerales." [En línea]. Disponible en: <http://www.onrm.minbas.cu/?q=node/21>.
4. **ONRM, Dirección de Hidrocarburos.** Reglamento Clasificación de Recursos y Reservas de Petróleo y Gas. La Habana: s.n.
5. **Ricardo Alarcón de Quesada.** 1995. [En línea] 23 de Enero de 1995. [En línea]. Recuperado el 15 de enero del 2010. Disponible en: <http://www.onrm.minbas.cu/files/Legislacion/LEGMINERA/LEYES/LEY%20MINA/S/LEY-76-94%20Ley%20de%20Minas.pdf>.
6. **RAE.** Diccionario de la Lengua Española – Vigésima Segunda Edición. Lenguaje. [En línea]. Disponible en: <http://buscon.rae.es/drael/>
7. **C. G. Hernández** (2006). Módulo Alojamiento del Sistema Automatizado para la Gestión de Información de la Misión Milagro, Universidad de las Ciencias Informáticas. Ciudad de la Habana.
8. **BRAINWORX.SA.** Software Guru, 2006.
9. **Sun Microsystems.** 1994-2008. Sun Microsystems. [En línea] 1994-2008. Disponible en: http://www.sun.com/emrkt/innercircle/newsletter/spain/0207spain_feature.html. Recuperado el: 16 de enero del 2010. MINBAS, Comisión de Redacción. 2007. Propuesta para la promulgación de una nueva ley de petróleo y gas natural. 2007.
10. **LÓPEZ, A. R.** Sistema Asistente para la Generación de Horarios de Cursos.: Computación, Electrónica, Física e Innovación Puebla, Universidad de las Américas Puebla, 2008. p.
11. **CARREÑO, Anabel** (2008). Las TIC's en el Diseño Curricular Bolivariano. [En línea]. Recuperado el 27 de abril del 2010. Disponible en: <http://anabelcarylásticos.blogspot.com/>

BIBLIOGRAFÍA

1. **STAIR, Ralph M.**, et al (2003). Principles of Information Systems, Sixth Edition. Thomson Learning, Inc., pp. 132. ISBN 0-619-06489-7
2. **RIORDAN, Rebecca** (1999). Designing Relational Database Systems. Microsoft Press. ISBN 0-7356-0634-X
3. **DATE, C.J.** An Introduction to Database Systems. 7th Edition. Reading: Addison-Wesley Publishing Company, 1999.
4. **HSQL Database Engine**. HyperSQL Database. 100% Java Database. [En línea] 2010. Disponible en: <http://hsqldb.org/>.
5. Wikipedia, la enciclopedia libre. HSQLDB. [En línea]. Disponible en: <http://en.wikipedia.org/wiki/HSQLDB>
6. **FLANAGAN, David**. Java en pocas palabras. Segunda edición. s.l.: O'Reilly & Associates. págs. 3-8. ISBN 1-56592-262-X.
7. **GOSLING James, JOY Bill, STEELE Guy, y BRACHA Gilad**, The Java language specification, tercera edición. Addison-Wesley, 2005. ISBN 0-321-24678-0.
8. **CORNEJO, J. E. G.** (2001). Arquitectura en Capas. Un camino hacia los procesos distribuidos.
9. **JACOBSON I, BOOCH G, RUMBAUGH J**. El Proceso Unificado de Desarrollo de Software. Addison-Wesley, 2000. ISBN 84-7829-036-2
10. **CASTILLA, I.C.U.d.**, Prácticas de la ingeniería de software, Una Herramienta CASE para ADOO. 2007.
11. **LARMAN, C.** (1999). UML y Patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall, México 1999. ISBN: 970-17-0261-1
12. **RUMBAUHG, J. I. JACOBSON** (2000). El Lenguaje Unificado de Modelado. Manual de referencia. Madrid, Pearson Educación.
13. **KRUCHTEN, P.** The Rational Unified Process: An Introduction (2nd Edition). Boston, 2001.
14. **GARCÍA ARENAS, María Isabel**, Introducción a XML, 2006. [En línea]. Recuperado el: 16 de enero del 2010. Disponible en: <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml>.

15. **SALAZAR GÓMEZ, Lisset; DUEÑAS NARANJO, Yarely.** Sistema automatizado para la captura de información referente al Balance Nacional de Recursos y Reservas de Petróleo de la Oficina Nacional de Recursos Minerales. Universidad de las Ciencias Informáticas. Ciudad de la Habana, 2008.
16. **MARTELL FERNÁNDEZ, Vladimir; FIGUEROA HIDALGO, Daira.** Sistema automatizado para la gestión de los datos del Balance Nacional de Recursos y Reservas de Petróleo y Gas de la Oficina Nacional de Recursos Minerales. Universidad de las Ciencias Informáticas. Ciudad de la Habana, 2008.
17. **BUSCHMANN Frank, R. M., ROHNERT Hans, SOMMERLAD Peter y STAL Michael.** Pattern-Oriented Software Architecture – A System of Patterns. 1996. p.
18. **TAYLOR Richard, n. M., ANDERSON Kenneth, WHITEHEAD James, ROBBINS Jason, NIES Kari, OREIZY Peyman y DUBROW Deborah.** A Component- And Message-Based Architectural Style for GUI Software. 17TH International Conference on Software Engineering, Seattle, 1995. p.
19. **ASSOCIATION, I. S.** Recommended practice for architectural description of software-intensive systems 2000. [En línea]. Disponible en: www.standards.ieee.org/reading/ieee/std_public/description/se/1471-2000_desc.html
20. **GAMMA Erich, HELM Richard, JOHNSON Ralph, VLISSIDES John.** Elements of Reuseable Object-Oriented Software, New York, 1997.
21. **FOWLER Martin, RICE David, FOEMMEL Matthew, HEATT Edward, MEE Robert, STAFFORD Randy.** Patterns of Enterprise Application Architecture. Addison Wesley, ISBN 2002. 0-321-12742-0.
22. **HAMILTON Kim, MILES Russell.** Learning UML 2.0. O'Reilly, 2006. ISBN 0-596-00982-8.
23. **ÖVERGAARD Gunnar, PALMKVIST Karin.** Use Cases Patterns and Blueprints. Addison Wesley Professional, 2004. ISBN 0-13-145134-0.
24. **DOMÍNGUEZ GENIZ Amalio Javier.** Aprenda Reportes con ayuda del plugin IREPORT para Netbeans y MySQL, en N Diapositivas. [En línea]. Recuperado el Disponible en 27 de mayo del 2010. Disponible en: <http://ajdgeniz.wordpress.com>

ANEXO III DIAGRAMAS DE CLASES DEL DISEÑO

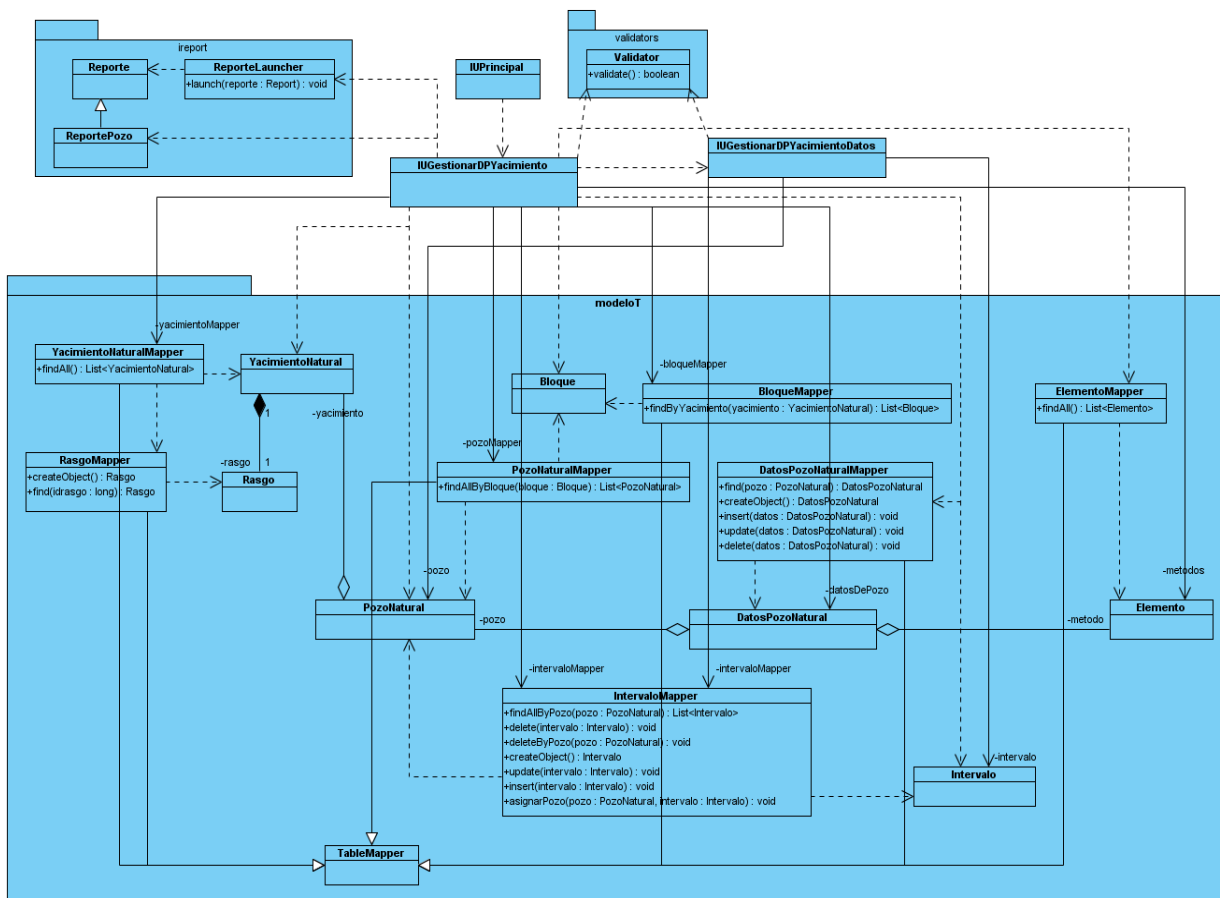


Figura 9: Diagrama del CU Gestionar datos primarios de Yacimiento

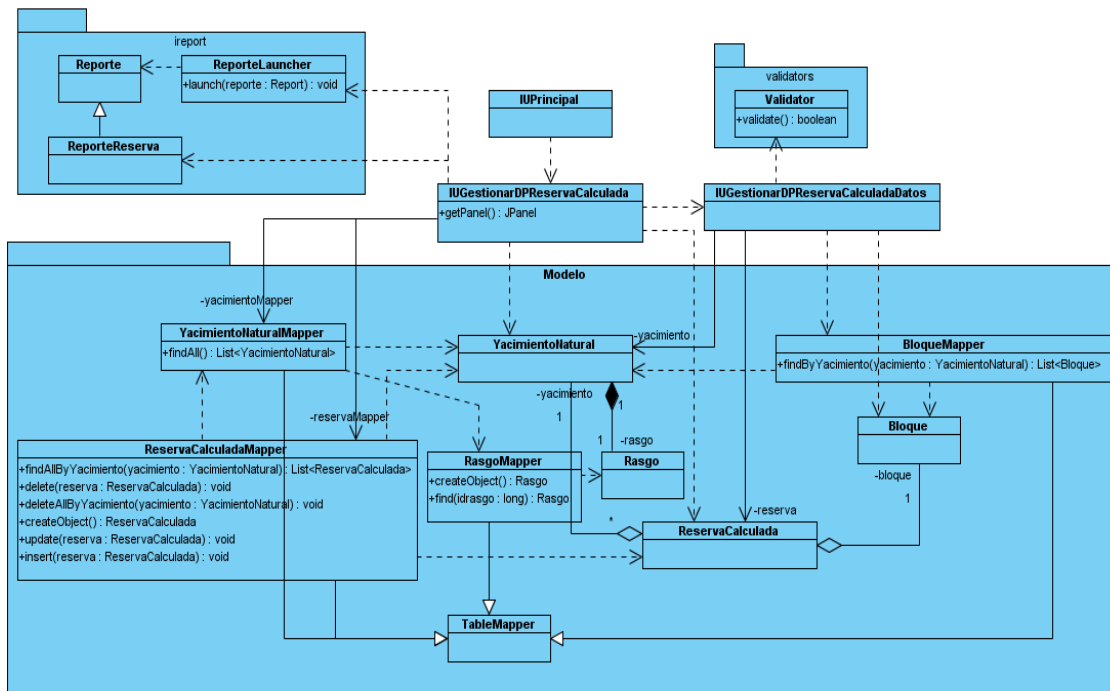


Figura 10: Diagrama del CU Gestionar datos de reserva calculada

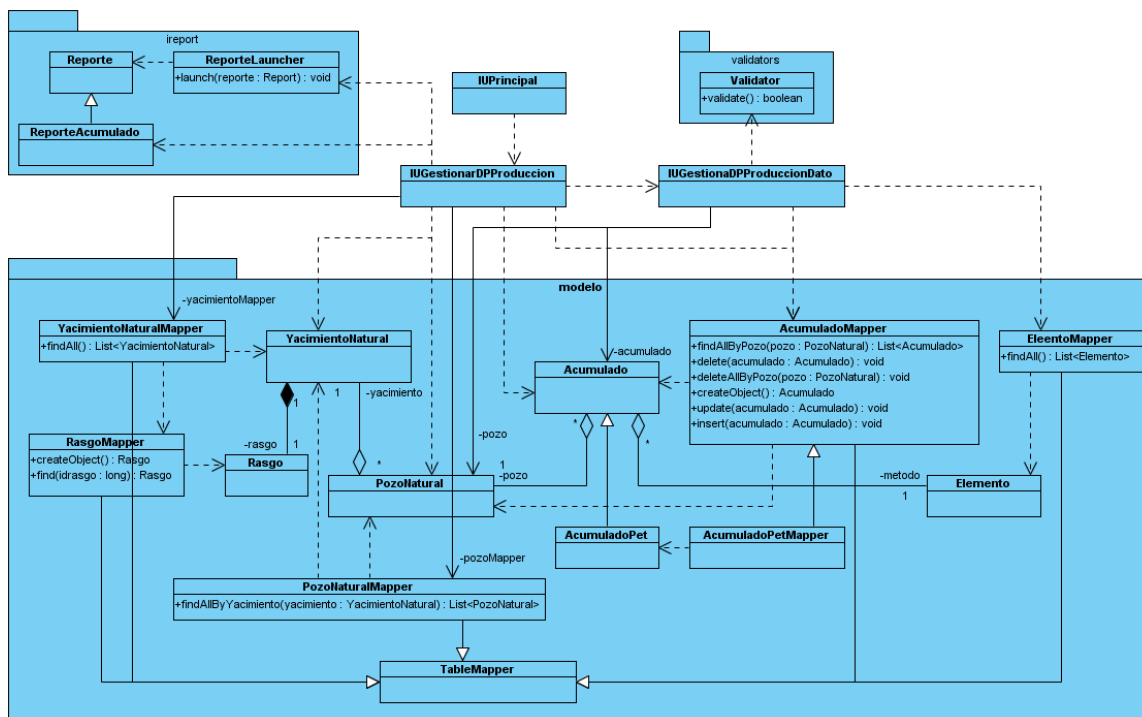


Figura 11: Diagrama del CU Gestionar datos de Producción

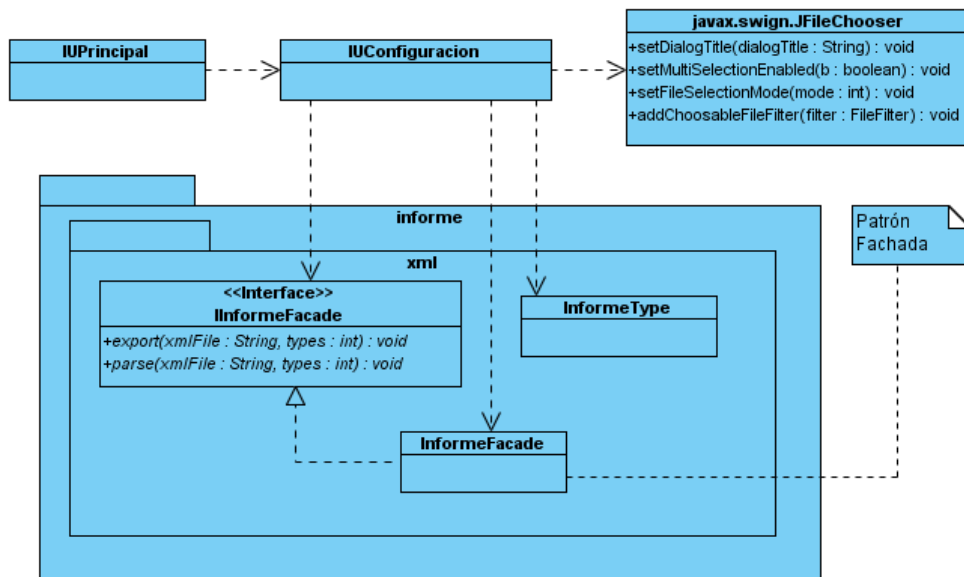


Figura 12: Diagrama del CU Cargar Configuración

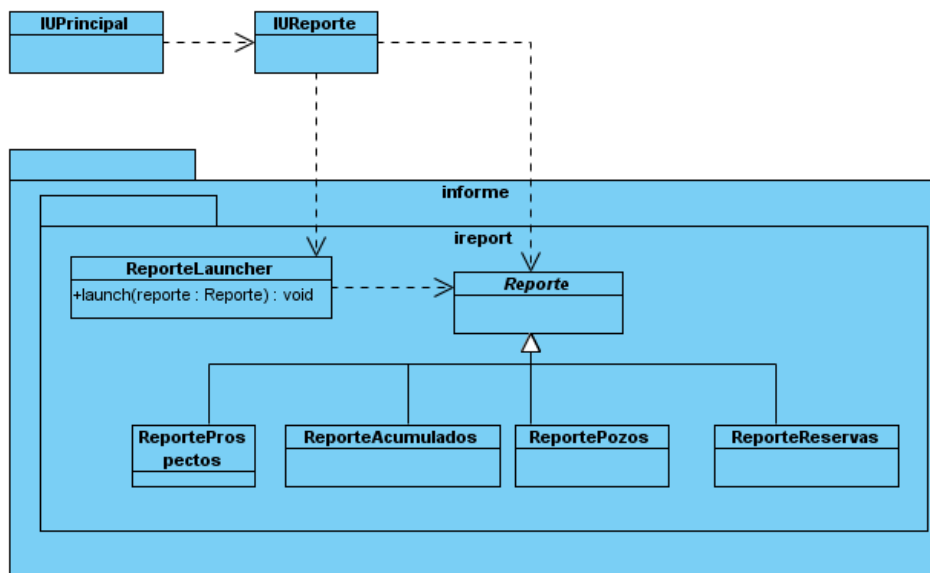


Figura 13: Diagrama del CU Generar Informe en Pantalla

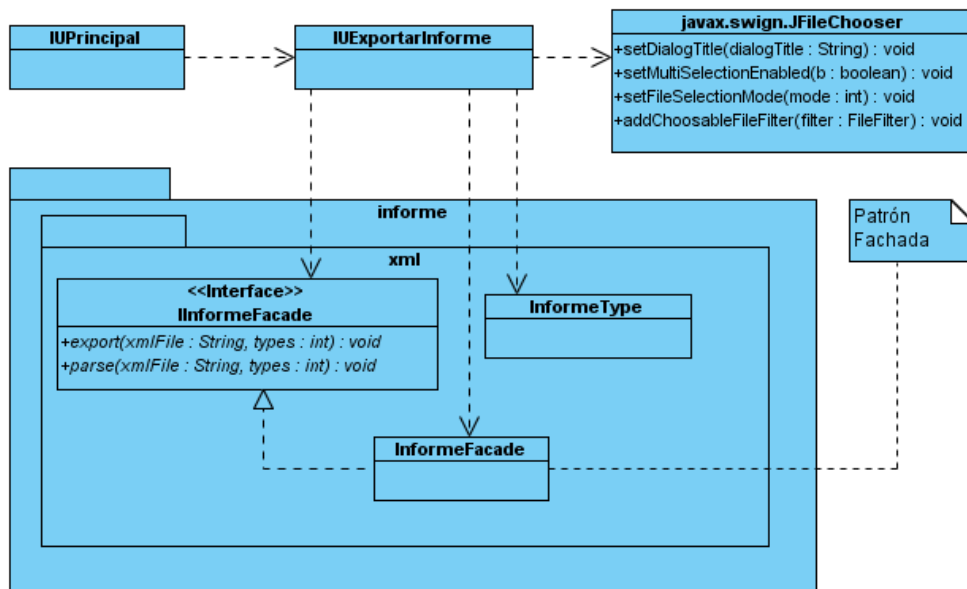


Figura 14: Diagrama del CU Exportar Informe XML

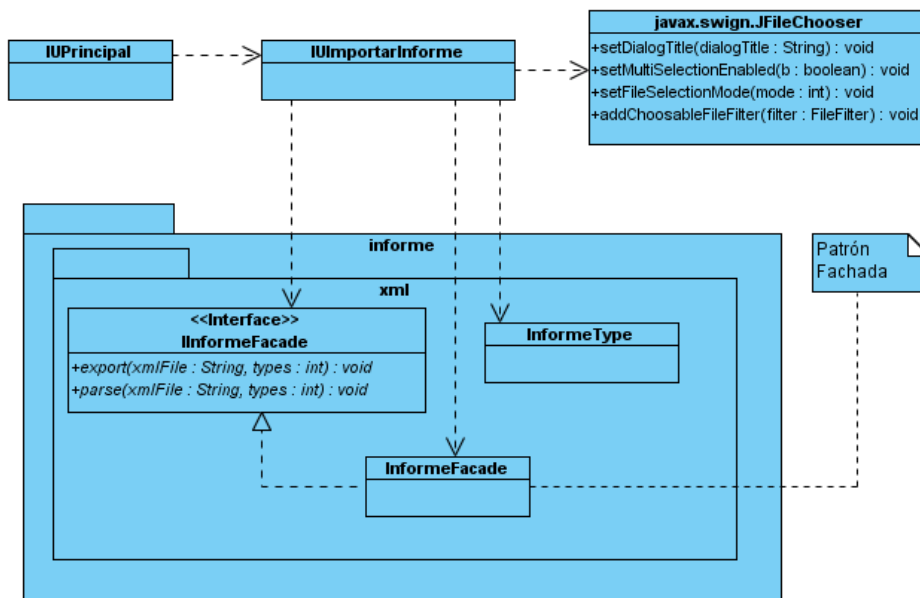


Figura 15: Diagrama del CU Importar Informe XML

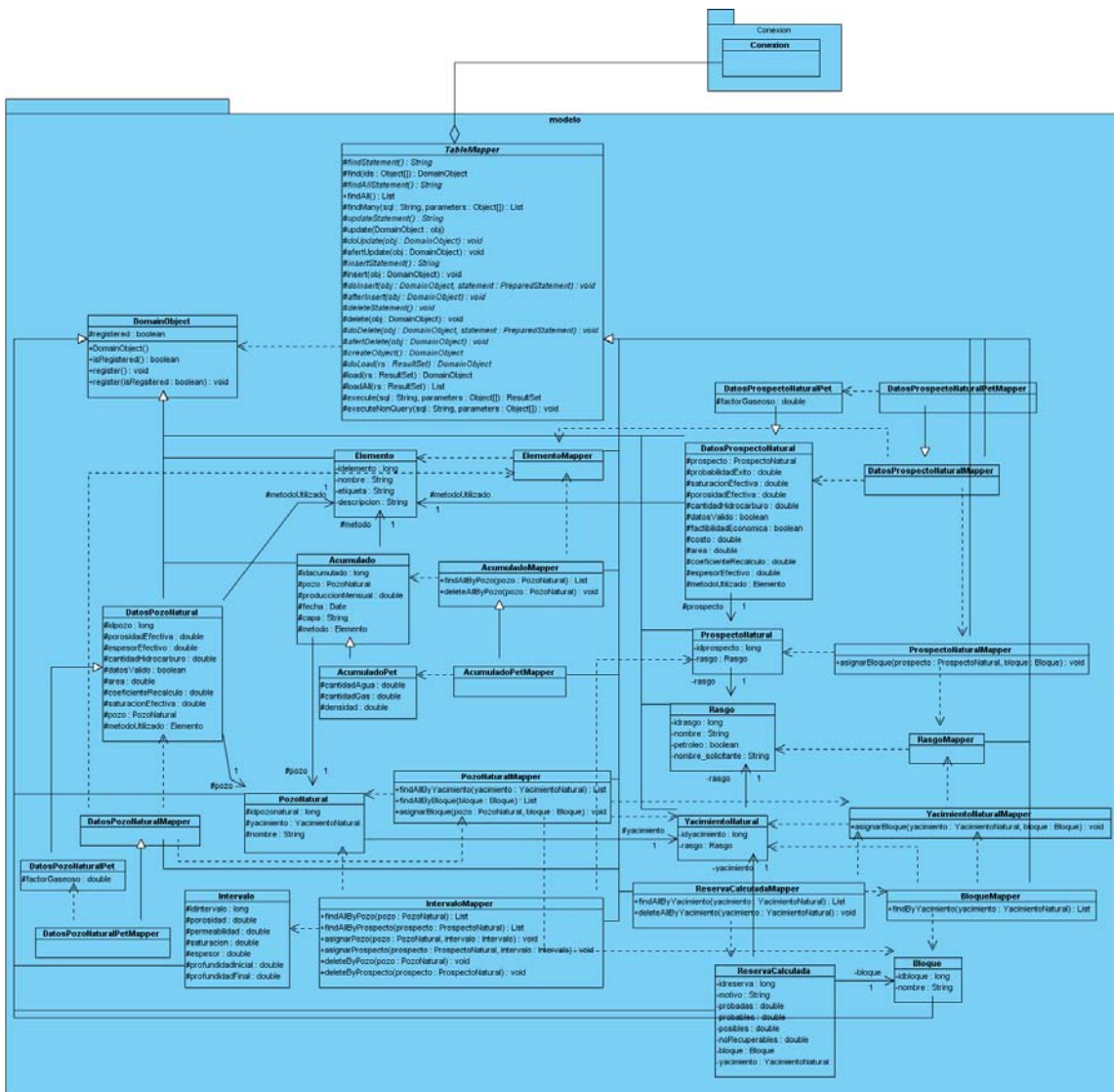


Figura 16: Diagrama de clases del diseño del Paquete "Modelo"

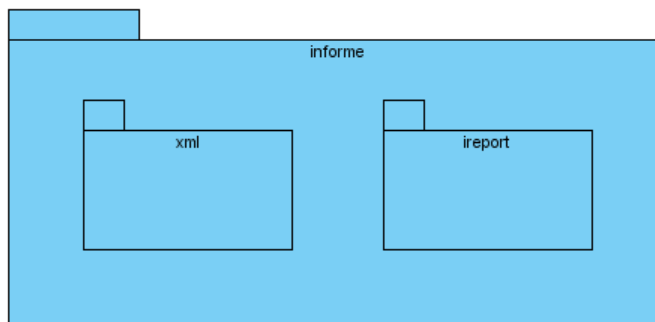


Figura 17: Diagrama de clases del diseño del Paquete “Informe”

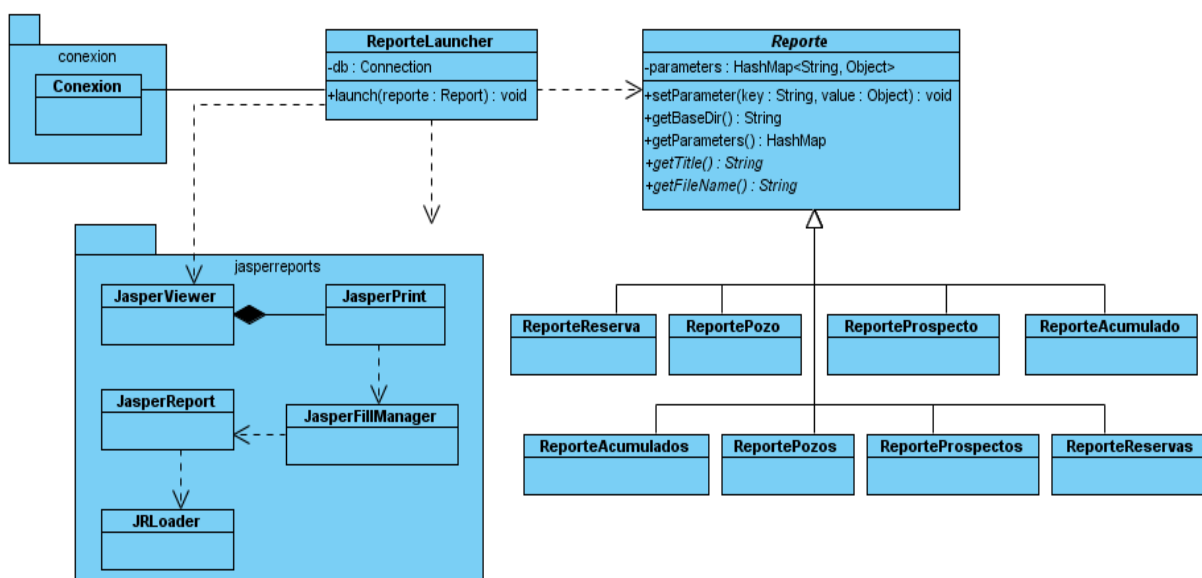


Figura 18: Diagrama de clases del diseño del Paquete “ireport”

