



UNIVERSIDAD DE LA CIENCIAS INFORMÁTICAS
PROYECTO PARA LA INFORMATIZACION DEL CONOCIMIENTO GEOLÓGICO

Sistema para el Registro e Importación de Definiciones y Bases de Datos Geológicas.

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

Autores: David Rodríguez Aldana.
Alexis Amaro Placeres.

Tutor: Ing. Jorge Luis Diéguez Escalona.

Junio del 2008

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Proyecto de Informatización del Conocimiento Geológico (PICG); así como a la Oficina Nacional de Recursos Minerales (ONRM) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los 18 días del mes de junio de 2008.

Firma del Autor

Firma del Autor

Firma del Tutor

OPINIÓN DEL TUTOR

Como tutor del Trabajo de Diploma “Sistema para el Registro e Importación de Definiciones y Bases de Datos Geológicas”, luego de haber culminado la realización del mismo, considero que los autores David Rodríguez Aldana y Alexis Amaro Placeres han desarrollado un conjunto de habilidades que les permitirán darle solución adecuadamente a cualquier tipo de necesidad de informatización que se presente en su vida profesional.

Durante la realización del presente trabajo los estudiantes han demostrado un alto grado de preocupación y responsabilidad ante el cumplimiento en tiempo de las tareas que se les programaron. Han trabajado coordinadamente dando muestras de poseer responsabilidad y compromiso en la realización de su tesis. Su desempeño ha manifestado que han desarrollado un valioso nivel de asimilación e investigación, llegando a alcanzar un profundo conocimiento y una gran capacidad para la toma de decisiones correctas.

Los estudiantes manifestaron laboriosidad y preocupación a lo largo del cumplimiento de las tareas programadas, logrando que los resultados obtenidos estuviesen acorde con los objetivos trazados aunque deben mejorar la coordinación de su trabajo en equipo.

Por otra parte, el elemento investigativo del documento, estuvo desde el inicio muy bien orientado y estructurado, basado en una gran cantidad de bibliografía actualizada. Cada contenido se ha expuesto con claridad y aporta grandes conocimientos al lector. Vale destacar que el proceso realizado es una tarea completamente nueva en nuestra universidad por lo que han realizado un trabajo admirable que normalmente es objeto de mucho trabajo y poca planificación, logrando obtener resultados satisfactorios.

Por todo lo anteriormente expresado consideramos que los estudiantes están aptos para ejercer como Ingenieros Informáticos; y proponemos que se le otorgue al Trabajo de Diploma la calificación de _____.

Tutor: Ing. Jorge Luis Dieguez Escalona

Fecha

A mis padres y a mis hermanos que siempre han estado apoyándome en esta tarea.

Alexis

Dedico éste; mi primer paso hacia el futuro a mis padres y mi familia.

David

Gracias a mis padres, a mis hermanos que me han apoyado a lo largo de este trabajo, también agradecer a mis amigos que han estado ahí cuando los he necesitado y en especial a mi compañero de tesis ya que su aporte ha sido imprescindible para la culminación de este trabajo.

Alexis

Gracias a mi hermana Tita, que luchó tanto conmigo. Que no se queden atrás mis colegas que no han dejado de apoyarme durante toda la carrera; gracias a todos y suerte con sus tareas.

David

RESUMEN

Con el desarrollo de la informática y el aumento de su impacto social, son cada vez más las instituciones u organizaciones que optan por incorporar aplicaciones que gestionen su información y automaticen muchas de sus tareas, logrando así una mayor dinámica en sus procesos de negocio.

El Proyecto de Informatización del Conocimiento Geológico (PICG) perteneciente a la Facultad #9 de la Universidad de la Ciencias Informáticas (UCI) en coordinación con la Oficina Nacional de Recursos Minerales (ONRM) esta llevando a cabo el desarrollo de diferentes sistemas para informatizar los servicios y procesos que se realizan en esta última, en aras de contribuir enormemente al desarrollo económico del país en el sector geominero. Dentro de estos sistemas se encuentra el relacionado con el tema de la captación de datos geológicos, actualmente llamado Geodato^{IC} dentro de la ONRM.

El presente trabajo de diploma tiene como objetivo realizar una investigación del proceso de registro e importación de Definiciones y Bases de Datos Geológicas llevado a cabo dentro de la ONRM cuyos resultados favorezcan el desarrollo de un software que permita la ejecución de dichos procesos de una forma mucho más eficiente y centralizada.

	Índice
Introducción	1
Métodos investigativos:	3
Capítulo I Fundamentación Teórica	5
1.1 Conceptos fundamentales.	5
1.1.1 Base de Datos Relacional con Atributos y Registros Variables	5
1.1.2 Rasgos.	5
1.1.3 Tipos de Rasgos.....	6
1.1.4 Variables Virtuales.....	6
1.1.5 Variables Elementales.	8
1.1.6 Perfiles de Trabajo.....	9
1.2 Sistemas implementados dentro del entorno del problema	10
1.2.1 GPDClasif.....	10
1.2.1.1 Concepción general del sistema.	10
1.2.1.2 Características.....	10
1.2.1.3 Deficiencias.	10
1.3 La importación de Bases de Datos.	11
1.3.1 DBConvert for Access & PostgreSQL.	11
1.3.2 Kexi.	12
Conclusiones del Capítulo.....	12
Capítulo II Tecnologías y Herramientas Utilizadas	13
2.1 Lenguajes de programación.	13
2.1.1 C++	14
2.1.2 C#	14
2.1.3 Java.....	15
2.2 Entornos de desarrollo integrado.	17
2.2.1 Eclipse.....	17
2.2.2 NetBeans IDE.....	18
2.3 Sistemas gestores de bases de datos	19
2.3.1 Firebird.	19
2.3.1 PostgreSQL.....	20
2.4 Metodología de desarrollo de software utilizada.	23
2.5 Roles y artefactos	24
2.5.1 Rol: Analista del proceso de negocio.	24
2.5.2 Rol: Analista del sistema.....	24

2.5.3 Rol: Diseñador de bases de datos.....	24
2.6 Patrones de arquitectura.....	25
2.6.1 Concepto de patrón.....	25
2.6.2 Patrones de arquitectura.....	25
2.6.3 Patrones de diseño.....	26
2.6.3.1 GRASP.....	27
2.7 Lenguaje unificado de modelado.....	28
2.8 Visual Paradigm for UML.....	29
Conclusiones del Capítulo.....	30
Capitulo III Descripción de la Solución Propuesta.....	31
3.1 Descripción de los procesos de negocio propuestos.....	31
3.1.1 El proceso de registro de Definiciones Geológicas.....	31
3.1.2 El proceso de importación de Bases de Datos Geológicas.....	31
3.2 Actores del negocio.....	32
Especialista.....	32
Especialista del Campo.....	32
3.3 Diagrama de casos de uso del negocio.....	32
3.4 Descripción textual de los casos de uso del negocio.....	33
3.4.1 Solicitar Registro de Definiciones.....	33
3.4.2 Solicitar Certificación de Base de Datos.....	34
3.5 Requerimientos funcionales.....	35
3.6 Requerimientos no funcionales.....	36
Usabilidad.....	36
Rendimiento.....	36
Software.....	36
Portabilidad.....	36
Seguridad.....	36
3.7 Descripción del sistema propuesto.....	36
3.7.1 Actores del Sistema.....	36
Especialista del Centro de Datos.....	36
3.7.2 Diagrama de Casos de Uso del Sistema.....	36
3.7.3 Descripción textual de los Casos de Uso del Sistema.....	37
3.7.3.1 Autenticar Usuario.....	37
3.7.3.2 Registrar Definiciones.....	39
3.7.3.3 Certificar Base de Datos.....	41

Conclusiones del Capítulo.....	42
Capitulo IV Construcción de la Solución Propuesta	43
4.1 Diagramas de clases.....	43
4.2 Principios del diseño	47
4.2.1 Estándares en la interfaz de usuario	48
4.2.3 Tratamiento de excepciones.....	48
4.2.4 Estándares de codificación.....	48
4.3 Diseño de la base de datos.....	49
4.3.1 Diagrama de clases persistentes.....	49
4.3.2 Modelo de datos.....	49
4.4 Modelo de implementación.	51
4.4.1 Diagrama de despliegue.....	51
4.4.2 Diagrama de Componentes.....	51
Conclusiones del Capítulo.....	52
Capitulo V Estudio de Factibilidad.....	53
5.1 Estimación del esfuerzo	53
5.1.1 Cálculo de Puntos de Caso de Uso sin Ajustar.....	53
5.1.2 Factor de Peso de los Actores sin ajustar (UAW).	53
5.1.3 Factor de Peso de los Casos de Uso sin ajustar (UUCW).	54
5.1.4 Cálculo de Puntos de Casos de Uso ajustados.	55
5.1.5 Factor de complejidad técnica (TCF).	55
5.1.6 Factor de ambiente (EF).....	56
5.2 Beneficios tangibles e intangibles	58
5.3 Análisis de costos y beneficios.....	59
5.4 Valoración de sostenibilidad según la dimensión socio-cultural, económica, ambiental y tecnológica.....	59
Conclusiones del capítulo	60
Conclusiones	61
Recomendaciones	62
Glosario de Términos.....	63
Referencias Bibliográficas.....	64
Bibliografía.....	65
Anexos.....	I

Introducción

La organización y estandarización de los datos espaciales es tarea que acometen cada día más países, a tal punto que la Organización de Naciones Unidas (ONU) está creando la serie de normas ISO 19100 con la finalidad de la estandarización internacional en el campo de la información geográfica para que:

- Apoye la comprensión y uso de la información geográfica
- Incremente la disponibilidad, acceso, integración y compartimiento de la información geográfica.
- Permita la interoperabilidad de los sistemas geoespaciales.
- Facilite el establecimiento de la infraestructura geoespacial a nivel local, regional y global

Para ello los países crean la Infraestructura de Datos Espaciales, la que es usada por las entidades y organizaciones relacionadas con la geominería, la agricultura, la meteorología, la hidrografía, etc. En este frente se destacan los Estados Unidos, Inglaterra y casi todos los países desarrollados económicamente y avanzan otros países como Colombia.

Cuba participa desde el año 2000 en eventos relacionados con esta temática y trabaja para crear su propia Infraestructura de Datos Espaciales para la cual la ONU ha coordinado ayuda y asesoría con Colombia y la Universidad de Wisconsin. Dentro de las tareas de Informatización de la Sociedad que dirige el Ministerio de Informática y Comunicaciones se ha creado la Secretaría Ejecutiva de la Infraestructura de Datos Espaciales de la República de Cuba encontrándose dichos trabajos en la fase inicial (definición de políticas y organización del trabajo) y se trabaja en la promulgación de un documento legal que regule estos asuntos. Durante el desarrollo del Programa se mantendrán contactos con esta Secretaría y se aplicarán todas las regulaciones que se orienten.

La actividad geológica para minerales sólidos y aguas minerales dirigida por la Unión de Empresas Geológicas inició en 1989 un trabajo con el fin de digitalizar el uso y conservación de la información, este empeño no llegó a su fin, pues la unificación con

la Unión de Minería y Sal y el inicio del período especial paralizaron lo proyectado. (Geominera Oriente)

El trabajo consistía en: Una “Base de Datos” Ramal con un “manipulador” general encargado de la codificación, captación, validación, recuperación y mantenimiento de los datos; enlazado a un conjunto de “módulos” independientes encargados de los cálculos y aplicaciones de las distintas especialidades (Laboratorio, Topografía, Geofísica, Geoquímica, Geología, Estadística).

El trabajo se inició y por las razones antes expuestas quedó en la situación siguiente: Se diseñó completamente la base de datos, se programó el manipulador, se definieron las interfaces del manipulador con cada uno de los módulos y se entregó la responsabilidad de cada módulo a cada uno de los equipos formados, dirigidos por especialistas de las diferentes Empresas (Laboratorio: Reynaldo Drake, Geofísica: Guillermo Rodríguez, Geología: César Rosales, Geoestadística: Ma. Esperanza Argudín, Coordinador General Héctor Camejo). No se ha podido obtener ninguna documentación de estos trabajos.

A inicios de la década del 90 se comenzaron a confeccionar bases de datos digitales. Al no realizarse bajo una dirección metodológica única, éstas se elaboraron en formas muy variadas y usando distintos programas; esta situación se ha extendido hasta la actualidad que aún quedan en ejecución trabajos iniciados en el año 2002. A finales de ese año el Instituto de Geología y Paleontología (IGP) se dio a la tarea de resolver la situación existente y uniformar y controlar la elaboración de las Bases de datos; en el propio IGP se ejecutaba en esos momentos el SIGEOL, trabajo finalizado en agosto/2003 el cual demostró y sirve de ejemplo de trabajo en lo referente a la estandarización de la información, creación y uso de base de datos para la confección de mapas geológicos, materializando con ello el inicio a nivel de la Unión Geominera el esfuerzo por la estandarización de la información geológica. (Geominera Oriente)

Como resultado de lo antes expuesto se deriva la siguiente **problemática**: No existencia de un Sistema óptimo que cuente con la suficiente documentación ni que registre Definiciones Geológicas e importe Bases de Datos Geológicas de forma única y centralizada en la Oficina Nacional de Recursos Minerales (ONRM). De esta problemática se deriva el **problema a resolver** de este trabajo: ¿Cómo lograr la validación e importación de Bases de Datos Geológicas así como también el registro de nuevas Definiciones Geológicas de forma única, centralizada y segura en la ONRM?

Una vez identificado el problema que da pie para la realización de este trabajo se definen como **objeto de estudio** las definiciones de la Tecnología de Base de Datos de Registros y Atributos Variables y el proceso de importación de Bases de Datos desde el Centro de Datos de la ONRM, como **campo de acción** la automatización de la captación, validación y almacenamiento de Bases de Datos Geológicas y como **idea a defender** un sistema basado en el diseño que se propone realizará de forma única y centralizada los procesos de registro e importación de definiciones y bases de datos geológicas.

Este trabajo propone una solución a dicho problema para la cual nos hemos trazado como **objetivo fundamental** llevar a cabo una Metodología de Desarrollo de Software que deje sentada las bases para desarrollar un módulo informático único a nivel nacional capaz de validar e importar Bases de Datos geológicas al Centro de Datos de la ONRM así como también generar las Definiciones de la Tecnología de Base de Datos de Registros y Atributos Variables.

Para dar cumplimiento a este objetivo se ha dividido el mismo en cuatro responsabilidades mucho más específicas:

- Análisis exacto y detallado de la Tecnología de Base de Datos de Registros y Atributos Variables.
- Estudio acerca de la importación de Bases de Datos Geológicas al Centro de Datos de la ONRM.
- Realización de todo el Proceso Unificado de Desarrollo de Software hasta los flujos de análisis y diseño para el sistema propuesto.

Métodos investigativos:

- **Teóricos:**

Análisis y síntesis

Se hace necesario analizar profundamente el proceso de recopilación de datos geológicos para descomponer cada uno de los factores implicados en el mismo. Una vez desglosados todos y cada uno de dichos factores se debe encontrar las relaciones entre los mismos y sus características mediante un proceso de síntesis; ya que un proceso de análisis debe ser sucedido por uno de síntesis.

Histórico y lógico

Es imprescindible saber los sucesos que anteceden a la problemática que ha conducido a la realización de este trabajo, para así comprender de una mejor manera la situación y los procesos que ocurren actualmente.

Modelación

Es muy útil el hecho de poder representar fenómenos de la vida diaria en esquemas y modelos para su mejor comprensión y descomposición. Este es un método sumamente útil para cualquier problemática, este trabajo no escapa de su alcance ya que se hace necesaria la representación en esquemas y diagramas de los procesos involucrados en el problema.

- **Empíricos:**

Observación

Con la observación de los procesos que envuelven cada situación es posible realizar una síntesis más eficaz de los mismos así como una modelación detallada del problema. Con la observación es posible captar de manera más rápida cada uno los componentes externos de un problema.

Capítulo I

Fundamentación Teórica

A continuación se hará un análisis de los diferentes conceptos y terminologías que involucra el problema en cuestión, los procesos que son inherentes al negocio así como también se hará un estudio de los actuales mecanismos utilizados en la Oficina Nacional de Recursos Minerales (ONRM) para registrar definiciones geológicas.

1.1 Conceptos fundamentales.

Para la digitalización de datos geológicos se ha hecho necesario un mecanismo capaz de agrupar los diferentes conceptos relacionados a dichos datos para así darle al proceso una visión mucho más centralizada y genérica. Como solución a lo antes expuesto se ha concebido la Tecnología de Base de Datos Relacional con Atributos y Registros Variables, que será conocida a partir de ahora como tecnología BDRAV, dicha tecnología cuenta con los elementos que dan la base teórica de este trabajo.

1.1.1 Base de Datos Relacional con Atributos y Registros Variables.

Es una tecnología desarrollada por el Grupo de Servicios Informáticos de la Empresa Geominera Oriente como parte del Sistema Geodato^{IC}. Esta representa la estructura relacional con la que se almacena los datos geológicos en una Base de Datos (BD), dadas sus características genéricas esta tecnología puede ser utilizada para almacenar todo tipo de de datos sean o no geológicos. A continuación se podrá apreciar los diferentes elementos que la forman.

1.1.2 Rasgos.

Es la entidad que representa un grupo de elementos georeferenciados del mundo real que contiene información geológica y tiene una representación geométrica dada, contiene los atributos de ubicación de acuerdo con el tipo geométrico establecidos en la ISO 19107 (Polígono, Línea, Punto, Poli línea o Segmento) y atributos adicionales que constituyen las llamadas variables del rasgo. (Geominera Oriente)

Ejemplo: Pozos de Perforación, Puntos de observación, Trincheras, etc.

Los rasgos fueron definidos a partir de los modelos lógicos y conceptuales realizados por diferentes grupos de diseño de las Empresas Geomineras, estos se incorporaron como datos al GeoDato^{IC}. Los atributos que caracterizan al rasgo y se nombran Variables Reales son:

Identificador. Valor numérico que identifica al rasgo dentro de la base de datos.

Nombre. Nombre del rasgo en el mundo real de hasta 12 caracteres.

Clase Geométrica. Clase geométrica a que pertenece el rasgo (Punto, línea, polígono).

Pertenencia. Identificador de rasgo al que puede pertenecer el rasgo (Un pozo puede pertenecer a un polígono, o un punto de muestreo puede formar parte de una trinchera).

1.1.3 Tipos de Rasgos.

Es una entidad perteneciente al grupo de entidades representadas por el Rasgo de la cual heredan sus atributos y pueden agregarse otros que constituyen las variables del Tipo de Rasgo. Es decir los Rasgos son meta clases cuyas instancias son los Tipos de Rasgos. (Geominera Oriente)

Estableciendo una analogía con la programación orientada a objetos (POO) los objetos o instancias de las clases incluidas en la generalización Rasgo se agrupan dentro de la generalización Tipo de Rasgo.

Ejemplo: Rasgo: Pozo de perforación y Tipo de rasgo: Pozo estructural.

Los atributos que caracterizan (**Variables Reales**) al Tipo de Rasgo son:

- **Identificador.** Valor numérico que identifica al Tipo de rasgo dentro de la Base de datos.
- **Tipo de Rasgo.** Código identificador del tipo de rasgo.

1.1.4 Variables Virtuales.

Son Variables Virtuales o simplemente Variables, los atributos normalizados de las clases persistentes que definen las entidades diseñadas en los modelos lógicos. (Geominera Oriente)

Ejemplo: Fecha de perforado el rasgo Pozo de perforación. Profundidad del muestreo.

Clases de variables

Las variables virtuales pueden ser:

Simples. Son las que se definen en tiempo de diseño.

Codificadas. Son las que toman el nombre de un clasificador de caracteres.

Ej. La variable Sílice esta definida como mineral en el clasificador de minerales.

Atributos de las variables virtuales

Las variables quedan completamente definidas por los siguientes atributos, que son utilizados en el GeoDato^{IC}:

Código. Es un número entero único para todo el programa GEODATO (VR_Invariable).

Nombre. Nombre corto de la variable con hasta 30 caracteres (VR_NomVariable).

Descripción. Descripción de la variable de hasta 70 caracteres (VR_NomLVariable).

Etiqueta. Etiqueta de hasta 10 caracteres la cual identifica la variable (VR_Etiqueta).

Tipo. Tipo de la variable (VR_Tipovvariable).

Obligatoriedad. Fija si la variable es obligatoria o no para el GeoDato^{IC} (VR_Obligatoria).

Agrupamiento de las variables virtuales

Las Variables Virtuales fueron agrupadas bajo los Dominios de datos que se corresponden con las especialidades que se encargaron de los modelos lógicos entregados:

- Geoquímica.
- Identificación geológica.
- Físico Mecánico.
- Ubicación.
- Petrografía Mineralogía.
- Punto Geológico.

- Geología Paleontología.
- Geofísica.
- Topografía.

1.1.5 Variables Elementales.

Las variables elementales son variables que describen a las variables virtuales a diferencias de las virtuales propiamente dichas que describen rasgos, tipos de rasgos y perfiles de trabajo. Sus atributos son los mismos que éstas. En la siguiente tabla se muestran los tipos de variables elementales. (Geominera Oriente)

Variable Elemental	Bytes	Dominio
Boolean	1	S ó N, 0 ó 1
Flotante	8	entre -1.79×10^{308} y 1.79×10^{308}
Entero pequeño	2	Entre (-32,768) y (32,767)
Clasificada	2	Toma valores de hasta 3 clasificadores de caracteres y es del tipo entero entre (-32,768) y (32,767)
Fecha	2	Fechas entre 1/1/1900 y 6/6/2079
Entero	4	Entre (-2,147,483,648) y (2,147,483,647)
TEXTO		Variable de hasta (2,147,483,647) caracteres
Real	4	Datos numéricos entre -3.40×10^{38} y 3.40×10^{38}
StringS08	8	Cadena de hasta 8 caracteres
StringS20	20	Cadena de hasta 20 caracteres

StringS70	70	Cadena de hasta 70 caracteres
StringS256	256	Cadena de hasta 256 caracteres
Imagen		Contiene imágenes de hasta (2,147,483,647) bytes
Tiempo	8	Tiempos y fechas entre 1/1/1753 00:00:00 y 31/12/9999 00:00:00

Ejemplo: La variable virtual profundidad para que quede completamente definida tiene a la variable elemental Unidad de medida.

1.1.6 Perfiles de Trabajo.

Es la representación digital dentro del GeoDato^{IC} de un conjunto de datos en un documento geológico. Contienen el conjunto de variables contenidas en una fuente de información dada dentro de los documentos de soporte físico de los datos primarios (Tablas, informes, registros, etc.). En estos perfiles se recogen las variables por niveles jerárquicos, se establecen sus restricciones, orden de captación y llaves de relación de los niveles jerárquicos. Con ellos puede crearse el formulario de captación en correspondencia con la fuente del dato y asegura la vista de usuario de los datos idéntica a su organización en el soporte original. (Geominera Oriente)

En resumen los Perfiles de Trabajo no son más que toda la documentación resultante al estudiar o catalogar un Rasgo.

Ejemplo: Documentación geológica de pozo. (Desde, Hasta, Documentación).

Punto de medición geofísica. (DGBouguer, Delta, GanmaTotal).

Tipos de Perfiles de trabajo

Los perfiles de trabajo, por la cantidad de perfiles se dividen en:

De registro único. (Ej. Documentación de Afloramiento)

MultiRegistro. (Ej. Documentación Geológica de Pozo)

1.2 Sistemas implementados dentro del entorno del problema

En la actualidad la ONRM cuenta con un sistema encargado de registrar las Definiciones de la tecnología de Base de Datos Relacional con Atributos y Registros Variables (BDARV). Para la importación de bases de datos actualmente no hay ningún sistema implementado; esta es una de las solicitudes del cliente como nueva funcionalidad de la solución al problema y uno de los aportes que el desarrollo de este trabajo brinda al sistema propuesto.

1.2.1 GPDClasif.

GPDClasif es una aplicación integrada al programa GeoDato^{IC} del Instituto de Geología y Paleontología (IGP).

1.2.1.1 Concepción general del sistema.

Es una aplicación de escritorio que permite de una forma sencilla la incorporación de nuevas Definiciones de la tecnología BDARV a la Base de Datos del Centro de Datos de la ONRM. Permite de igual manera gestionar dichas Definiciones.

1.2.1.2 Características.

- 1- Desarrollada en Delphi.
- 2- Trabaja sobre una Base de Datos de SQL Server 2000.
- 3- No es muy sencilla de usar.
- 4- Sus funcionalidades están divididas según las Definiciones de la tecnología BDARV.
- 5- Gestiona las definiciones por Separado.

1.2.1.3 Deficiencias.

Aunque desde hace algunos años esta aplicación está siendo usada en la ONRM posee ciertas deficiencias que hacen indispensables su sustitución. En primer lugar GPDClasif no posee ningún tipo de documentación, ya que fue desarrollada sin un correcto Proceso de Desarrollo de Software, esto hace que su soporte, mantenimiento

y actualizaciones sean prácticamente imposible. Está completamente sobre una plataforma propietaria por lo que no cumple con las aspiraciones de nuestro país para migrar a software libre. Las Definiciones se almacenan en la misma Base de Datos en la que están todos los datos registrados de la ONRM, esto es una de las deficiencias más grandes que tiene GPDClasif porque cada vez que un especialista sale al campo para registrar datos debe llevar consigo una copia de toda la base de datos cuando solamente necesita las Definiciones actualizadas.

1.3 La importación de Bases de Datos.

Actualmente a raíz de los continuos cambios de tecnología para el almacenamiento de datos el problema de la importación de Bases de Datos de una tecnología a otra se ha hecho común alrededor del mundo.

En este problema en particular se hace necesario importar Bases de Datos provenientes del Gestor Access de Microsoft hacia Gestores libres, es por ello que se propuso la tarea de buscar otras aplicaciones implementadas en el ámbito internacional que resuelvan esta problemática, para tener una idea del proceso a realizar.

1.3.1 DBConvert for Access & PostgreSQL.

DBConvert for Access & PostgreSQL (DBConvert) fue desarrollado por DMSoft Technologies en el 2001 y es un convertidor de Bases de Datos el cual convierte de MS Access (mdb) a PostgreSQL y viceversa con solo configurar algunas opciones en una aplicación guiada paso a paso. DBConvert soporta Unicode que nos permite mostrar el lenguaje de símbolos especiales correctamente. Mejora la usabilidad, trae nuevos diseños y diferentes máscaras que contienen soporte Multilenguaje. DBConvert convierte llaves primarias, índices y llaves foráneas. Con él también es posible renombrar campos e índices antes de la conversión de la Base de Datos. Facilita la detección de errores y posee la propiedad de salvar datos hacia un script de PHP o hacia un archivo dumb lo cual permite hacer restricciones sobre la Base de datos de PostgreSQL. Posee gran velocidad en la conversión de Bases de Datos. Una de las propiedades claves de esta herramienta es que permite filtrar datos durante la conversión, esto nos da más control sobre el proceso de migración de datos. (DMSoft Technologies)

Es una herramienta excelente, poderosa y de un entorno muy amigable, pero no es de código abierto y por lo tanto no es factible como punto de partida para el sistema a implementar.

1.3.2 Kexi.

Kexi es un aplicación informática integrada para el manejo de datos, incluida dentro de la suite ofimática KOffice. Permite diseñar e implementar bases de datos, insertar y procesar datos y hacer consultas sobre los mismos. Kexi puede conectarse con distintos servidores de bases de datos, como por ejemplo PostgreSQL y MySQL. (Kexi, 2006)

También puede trabajar sin un servidor utilizando el motor de bases de datos embebido SQLite, lo que permite al usuario almacenar los datos y el diseño de la base de datos en un único archivo informático. Se pueden crear formularios para proporcionar un interfaz a medida al usuario para trabajar con los datos.

Con Kexi se pueden realizar consultas de datos de forma gráfica, imprimir o pre visualizar informes, programar scripts con los lenguajes Python y Ruby, y tiene un lenguaje de macros similares a las de Microsoft Access. Todos los objetos de base de datos (tablas, consultas, formularios, etc.) se almacenan en el archivo base de datos, facilitando compartir la aplicación completa, datos y diseño, como un único archivo normal.

Si bien Kexi no convierte Bases de Datos de Microsoft Access hacia otros formatos; este tiene la capacidad de importar dichos archivos y trabajar sobre ellos; por eso precisamente, además de que es de código abierto, se prestara especial atención en esta aplicación ya que una vez importados, y con la posibilidad de modificarlos, estos archivos pueden ser exportados hacia otros Gestores.

Conclusiones del Capítulo

Una vez analizados los conceptos y aspectos fundamentales que forman la base teórica de este trabajo están creadas las condiciones para analizar las diferentes tecnologías que se utilizaron para dar solución a los problemas existentes.

Capítulo II

Tecnologías y Herramientas Utilizadas

Para el desarrollo de cualquier Sistema Informático es necesario tener en cuenta las Tecnologías, Metodologías y Herramientas que intervienen en dicho proceso. En este capítulo se expondrán una serie de herramientas y tecnologías que se tuvieron en cuenta para la realización de este trabajo; ¿Cuáles fueron las escogidas? ¿Por qué fueron escogidas? Las respuestas a estas preguntas también serán plasmadas en este apartado.

2.1 Lenguajes de programación.

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos. (Gutierrez)

Un lenguaje de programación es cualquier lenguaje artificial, el cual, se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. Se asume que las instrucciones así escritas son traducidas luego a un código que la máquina pueda “comprender”. El proceso de traducción es realizado normalmente por la computadora, usando un programa especializado para tal fin. (Ministerio del Poder Popular para Ciencia y Tecnología)

De acuerdo con lo antes expuesto se concluye que un lenguaje de programación no es más que un lenguaje que se utiliza para que los usuarios se comuniquen con la computadora y este a su vez es interpretado esta.

El sistema a implementar deberá ser una aplicación destinada al uso de una sola persona, el Especialista Geólogo del Centro de Datos, en una oficina ya que no necesitará ser utilizada en ningún otro ámbito fuera, otra razón para no desarrollar una aplicación web en este sentido es que el sistema propuesto no tiene porque interactuar con ningún personal fuera de la ONRM, por estas razones se descartó la opción de utilizar tecnologías Web para su desarrollo y se ha centrado la atención en las Aplicaciones de Escritorio; a continuación se dará una síntesis de los distintos lenguajes de programación que se tuvieron en cuenta para este trabajo.

2.1.1 C++

El C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C.

Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes como ROOT ([enlace externo](#)). Las principales características del C++ son las facilidades que proporciona para la programación orientada a objetos y para el uso de plantillas o programación genérica (*templates*).

C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos trae (obliga a hacerlo casi todo manualmente al igual que C) lo que "dificulta" mucho su aprendizaje. (Tecnología, Colegio San José SS.CC.)

En la actualidad, C++ es un lenguaje versátil, potente y general, mantiene las ventajas del C en cuanto a riqueza de operadores y expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original.

A pesar de su merecida reputación C++ arrastra algunas deficiencias de la programación estructurada (PE), como lo es el uso de variables globales que atenta seriamente al encapsulamiento de datos promovido por la programación orientada a objetos; razones como esta y la ausencia de un Entorno de Desarrollo Integrado libre lo suficientemente potente en comparación con otros que se analizarán más adelante hacen que este lenguaje no sea el más indicado para el desarrollo de este trabajo.

2.1.2 C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la Asociación de Constructores de Computadoras Europea (ECMA) e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes.

C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 2 de Febrero del 2008 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, tipos anónimos y el Language Integrated Query (LINQ).

Aunque C# forma parte de la plataforma.NET, esta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado el que provee el Framework de DotGNU - Mono es que no generen programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX / Linux.

Actualmente C# se encuentra entre los 10 lenguajes más utilizados, a pesar de su corta historia. (Tiobe Software)

Este lenguaje posee a priori características que lo convierten en el idóneo para el desarrollo de cualquier Sistema Informático pero el Proyecto Mono que es hasta ahora el único compilador libre para C# es aún muy joven y se prefirió optar por tecnologías con un mayor tiempo dentro de la comunidad mundial de programadores.

2.1.3 Java.

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La implementación original y de referencia del compilador, la máquina virtual y las librerías de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también

implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, 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 los programas Java todavía no es software libre). (Sitio oficial de Sun Microsystems, 2008)

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PCs de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecute en cualquier PC.

En las primeras versiones de la plataforma Java existían importantes limitaciones en las Interfaces de Programación de Aplicaciones (APIs) de desarrollo gráfico. Desde la aparición de la librería Swing la situación mejoró substancialmente y posteriormente con la aparición de librerías como SWT hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo y usabilidad sea relativamente sencillo.

Es precisamente Java el Lenguaje que se ha seleccionado para desarrollar este sistema por su versatilidad y la comodidad con la que se programa en él, además ofrece un sin número de bibliotecas externas para utilizar en disímiles problemáticas que puedan surgir en el desarrollo de un Sistema. Java es también con respecto a los otros Lenguajes analizados el único que cuenta con Entornos de Desarrollos Integrados (IDEs) libres con potentes herramientas para el trabajo con interfaces gráficas, precisamente a continuación se expondrán los diferentes IDEs para Java que se tuvieron en cuenta.

2.2 Entornos de desarrollo integrado.

Un Entorno de Desarrollo Integrado (IDE) es un entorno de programación que ha sido empaquetado como una aplicación; la cual esta compuesta por un Editor de Código, un Compilador, un Depurador y un Editor de Interfaz Gráfica. Todo de lenguaje de Programación posee al menos un IDE, y estos a su vez pueden utilizar uno o varios Lenguajes de Programación. Los IDEs constituyen herramientas definitivas el desarrollo de un Sistema Informático. Ahora se analizará algunos IDEs relacionados con Java, Lenguaje de programación escogido para desarrollar este sistema.

2.2.1 Eclipse.

Eclipse es un entorno de desarrollo integrado de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plug-in permite escribir cualquier extensión deseada en el ambiente

El Kit de Desarrollo de Software (SDK) de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código. El IDE también hace uso de un espacio de trabajo, en este caso un grupo de metadatos en un espacio para archivos planos, permitiendo modificaciones externas a los archivos en tanto se refresque el espacio de trabajo correspondiente.

Eclipse es sin dudas un IDE excepcional, posiblemente el mejor existente para Java su versatilidad y potencia son incalculables, pero tiene un talón de Aquiles que lo hace menos indicado para este trabajo, el caso es que en el trabajo con Interfaces Gráficas para la creación de Aplicaciones de Escritorio es aún rudimentario y engorroso, muchos de los componentes gráficos debes ser creados a mano y eso entorpece el trabajo del programador; queda descartado por tanto el Eclipse.

2.2.2 NetBeans IDE.

NetBeans se refiere a una plataforma para el desarrollo de aplicaciones de escritorio usando Java y a un entorno de desarrollo integrado (IDE) desarrollado usando la Plataforma NetBeans.

La plataforma 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 APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).

- Administración de ventanas.
- Framework basado en asistentes (diálogo paso a paso).

Se escogió específicamente por este IDE ya que posee las herramientas más sólidas para el trabajo con Interfaces Gráficas. Además es de código abierto, lo que ofrece al programador la ventaja de poder personalizar el IDE para una mejor satisfacción de sus necesidades.

2.3 Sistemas gestores de bases de datos

El sistema debe ser capaz de manejar y almacenar grandes volúmenes de Información por lo cual se necesitará una Base de Datos para facilitar dichos procedimientos. Seguidamente se expondrán características de los diferentes Sistemas Gestores de Bases de Datos que se tuvieron en cuenta para el desarrollo la Base de Datos del sistema.

Un Sistema Gestor de Bases de Datos (SGDB) es un conjunto de Programas que permiten crear y mantener una Base de Datos, asegurando la integridad, seguridad y la confidencialidad de datos almacenados.

2.3.1 Firebird.

Firebird es un sistema de administración de base de datos relacional (o RDBMS) SQL de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en el 2000. Su código fue reescrito de C a C++.

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Soporte de transacciones y claves foráneas.
- Es medianamente escalable.
- Buena seguridad basada en usuarios/roles.
- Diferentes arquitecturas, entre ellas el Firebird empotrado (*embedded server*) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird.

- Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como CD-ROM.
- Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad y restauraciones periódicas.
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Completo lenguaje para la escritura de disparadores (*triggers*) y procedimientos almacenados.
- Capacidad de almacenar elementos BLOB (*Binary Large Objects*).

Existen dos tipos de servidor Firebird para ser instalados: **Classic** y **Super Server**. Si bien tienen varias diferencias menores entre sí, la principal consiste en que el Super Server maneja hilos de ejecución individuales para cada conexión. Por lo tanto para un número reducido de conexiones el recomendado sería el Classic porque consumirá menor cantidad de recursos.

Los propios desarrolladores de Firebird recomiendan lo siguiente a la hora de decidirse por uno de estos servidores:

- En plataformas Windows seleccionar el **Super Server**.
- En Linux simplemente elegir cualquiera, según las conexiones estimadas. En la mayoría de las situaciones no se notará diferencias en la ejecución.

Podría considerarse un tercer tipo, el **Embedded**. Éste consiste en una única Biblioteca de Enlace Dinámico (DLL), de unos 2 MB de tamaño, que contiene todo el servidor. De esta forma se puede tener un DBMS completo disponible y distribuible junto con aplicaciones de usuario.

Éste es un gestor muy potente y que presume de una gran velocidad a la hora de realizar consultas pero no ha sido seleccionado por razones que se expondrán seguidamente.

2.3.2 PostgreSQL.

PostgreSQL es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD.

Como muchos otros proyectos open source (código abierto), el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el *PostgreSQL Global Development Group* (PGDG).

Su instalación es ilimitada. Con PostgreSQL, nadie puede ser demandado por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software. No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento. Posee flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento. Existen varias herramientas gráficas de alta calidad para administrar las bases de datos y para hacer diseño de bases de datos para es gestor. Muchas organizaciones, incluyendo grandes corporaciones, instituciones gubernamentales y pequeños negocios en línea usan PostgreSQL para manejar sus datos más valiosos y aplicaciones de misión crítica. (Molina, Soluciones Linux, 2008)

Beneficios

- Seguridad:

PostgreSQL ha agregado permisos a esquemas, funciones y otros objetos y nuevas opciones de configuración para aumentar la granularidad del control del administrador sobre la seguridad.

- Administración:

PGAdmin y phpPgAdmin, dos de las tres GUIs (Interfaces Gráficas) más populares para PostgreSQL tendrán nuevas versiones para PostgreSQL, haciendo más comfortable que nunca el cambio para administradores de bases de datos.

- Facilidad de uso:

Las Interfaces Gráficas hacen que las Bases de Datos de PostgreSQL sean más fáciles de usar.

- Escalabilidad:

Está liberado bajo la licencia BSD, lo que significa que cualquiera puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente. Es más, la licencia BSD permite redistribuir el código modificado o no como software cerrado, en contraposición a la licencia GPL que fuerza a que las modificaciones sean publicadas también bajo la GPL.

Ahora PostgreSQL soporta el estándar de comunicaciones TCP/IP IPv6, y el protocolo Rendezvous de Apple para funcionamiento sin configuración. Posee una gran escalabilidad, ya que es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. (Molina, Soluciones Linux, 2008)

PostgreSQL al igual que NetBeans es extensible, su código fuente esta disponible para todos, por lo que se puede modificar a gusto del desarrollador, esta es un característica que no posee Firebird y que hace que este gestor sea muy indicado para la solución que se busca debido a que el proceso de importación de bases de datos requiere de ciertas alteraciones en el funcionamiento del gestor. La estrategia de almacenamiento de filas MVCC le brinda concurrencia a la hora de manejar grandes volúmenes de información lo cual lo hace muy apropiado para la solución propuesta. El uso de las Funciones dentro de este gestor hace muy flexible el trabajo en función de las operaciones que se realizan sobre la Base de Datos, estas funciones pueden ser para ejecutarse con los derechos del usuario ejecutor o con derechos de usuario previamente definidos; esto le brindara al sistema una mayor seguridad para los datos almacenados, además de que estas pueden implementarse en un sin número de lenguajes de programación. PostgreSQL es orientado a objetos, esto permite que cada uno de los componentes de la estructura de la Base de Datos pueda ser tratado como un objeto, característica de suma importancia en el sistema que se implementará ya que facilitará enormemente la Importación de Bases de Datos Geológicas, además de poseer una gran variedad de tipos e incluso tipos y operadores definidos por el usuario, cosa que no posee Firebird y es una característica sumamente útil para el proceso de importación. Es por tanto este gestor el indicado para el sistema.

2.4 Metodología de desarrollo de software utilizada.

Definidas las herramientas que darán soporte a la base teórica es necesario definir la Metodología de Desarrollo de Software que guiará el proceso de automatización, en este caso se optó por el Proceso Unificado de Rational (RUP).

El RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. (España, 2008)

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.
- **Mantenimiento:** una vez instalado el producto, el usuario realiza requerimientos de ajuste, esto se hace de acuerdo a solicitudes generadas como consecuencia del interactuar con el producto.

Características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.

- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

En lo que se refiere a la metodología esta comprende tres frases claves: Dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental, lo cual le brinda una sin igual eficacia en cuanto a Calidad de Software se refiere.

2.5 Roles y artefactos

Teniendo en cuenta los roles que define la metodología RUP, se ha decidido que los roles que se desempeñarán serán: analista del proceso de negocio, analista del sistema y diseñador de bases de datos; los artefactos serán los que define la metodología para estos roles.

2.5.1 Rol: Analista del proceso de negocio.

El analista del proceso de negocio conduce y coordina el caso de uso del negocio que modela contorneando y delimitando la organización que es modelada; por ejemplo, el establecer que actores del negocio y casos de uso del negocio existen y como trabajan entre ellos. El analista del proceso de negocio es responsable de la arquitectura del negocio.

2.5.2 Rol: Analista del sistema.

Conduce y coordina los requerimientos y los Casos de Uso modelando y delimitando la funcionalidad del sistema y delimitando el sistema; por ejemplo, estableciendo que actores y casos de uso existen y como interactúan.

2.5.3 Rol: Diseñador de bases de datos.

El rol Diseñador de Base de Datos define las tablas, índices, vistas, constraints, triggers y otros objetos específicos de la base de datos necesarios para almacenar, recuperar y eliminar objetos persistentes.

2.6 Patrones de arquitectura

2.6.1 Concepto de patrón.

Según Christopher Alexander:

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

2.6.2 Patrones de arquitectura.

Los patrones de arquitectura describen un problema particular y recurrente de diseño, que aparece en contextos de diseño específico, y presenta un esquema genérico demostrado con éxito para su solución.

Algunos patrones de arquitectura son:

- 1- Arquitecturas de tubería-filtros.
- 2- Modelo-Vista-Controlador (MVC).
- 3- Arquitectura en Capas.
- 4- Arquitecturas orientadas a objetos.
- 5- Arquitectura Basada en Componentes.
- 6- Arquitecturas de Pizarra o Repositorio.
- 7- Arquitecturas Orientadas a Servicios (SOA en inglés).
- 8- Arquitecturas Basadas en Eventos.
- 9- Arquitecturas Basadas en Recursos.

De acuerdo con la definición y la función que realizan estos patrones se ha decidido utilizar en el diseño del sistema las siguientes:

Arquitectura en capas: Este patrón define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar qué se puede reutilizar, y proporciona una estructura que nos ayuda a tomar decisiones sobre qué partes comprar y qué partes construir.

Los principales estilos de arquitecturas estratificadas de las aplicaciones distribuidas contemporáneas son:

- Arquitecturas de dos niveles
- Arquitecturas de tres niveles
- Arquitecturas de n niveles

A continuación se muestra una especificación del estilo escogido para este patrón en específico.

Arquitectura en tres capas: La comunidad de software desarrolló la noción de una arquitectura de tres niveles. La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación.

La capa intermedia (lógica de aplicación) es básicamente el código al que recurre la capa de presentación para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación siempre que esta presente una interfaz claramente definida a la capa de presentación.

La tercera capa contiene los datos necesarios para la aplicación. Estos datos consisten en cualquier fuente de información, incluido una base de datos de empresa como Oracle o Sybase, un conjunto de documentos XML o incluso un servicio de directorio como el servidor LDAP. Además del tradicional mecanismo de almacenamiento relacional de bases de datos, existen muchas fuentes diferentes de datos de empresa a las que pueden acceder las aplicaciones.

2.6.3 Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones

anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Este conjunto de buenas prácticas son fácilmente adaptables a muchos problemas dentro de la programación orientada a objetos; a continuación se analizan los patrones que se utilizaran en el diseño del sistema.

2.6.3.1 GRASP

En diseño orientado a objetos, GRASP son patrones generales de software para asignación de responsabilidades. Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software.

Seguidamente se expondrán las variantes de este patrón seleccionadas para el sistema en cuestión:

Creador

El patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto, o
- Usa directamente las instancias creadas del objeto, o
- Almacena o maneja varias instancias de la clase

Este caso en particular es sumamente útil para este trabajo ya que para el registro de las definiciones geológicas es imprescindible tener claro en todo momento quien tiene la responsabilidad de crear los objetos de cada definición; ejemplo: cada rasgo tiene la responsabilidad de crear sus variables virtuales, y estas a su vez crearan sus variables elementales.

Controlador

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

Este patrón se acopla al patrón de arquitectura en tres capas anteriormente descrito las clases pertenecientes al controlador son las identificadas por dicho patrón de arquitectura como las contenidas en la capa de lógica de la aplicación.

2.7 Lenguaje unificado de modelado.

Lenguaje Unificado de Modelado (UML) es el Lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el Object Management Group (OMG). Es un Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (Zero)

UML es un "Lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema, para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

Además de haberse convertido en un estándar de facto, UML es un estándar industrial promovido por el Grupo de Gestión de Objetos (OMG) al mismo nivel que el estándar de Arquitectura Común de Intermediarios en Peticiones a Objetos (CORBA) para intercambio de objetos distribuidos. Para la revisión de UML se formaron dos "corrientes" que promovían la aparición de la nueva versión desde distintos puntos de vista. Finalmente se impuso la visión más industrial frente a la académica. Recientemente se ha publicado la versión 2.0 en la que aparecen muchas novedades y cambios que, fundamentalmente, se centran en resolver carencias prácticas. Además, esta versión recibe diversas mejoras que provienen del lenguaje SDL. (Gracia, 2005)

Lógicamente UML es de uso obligatorio si se utiliza RUP, a continuación se expondrá la Herramienta de Modelado basada en UML que se utilizará en este trabajo.

2.8 Visual Paradigm for UML.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (León Pavón, 2007)

Lista de características:

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo.
- Interoperabilidad con modelos UML2.
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generación de código y despliegue de para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación
- Generador de informes para generación de documentación
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas de MS Visio.

- Editor de figuras.

Esta es una potente herramienta para modelado y en el caso en particular de este trabajo viene muy bien su integración con Java, así como su completamiento de código y la generación de Bases de Datos. Posee además un entorno muy amigable y fácil. Es importante señalar que esta herramienta soporta Windows y Linux, además de ser libre.

Conclusiones del Capítulo

Se ha concluido con un paso vital para la realización de un Sistema Informático, ahora puede decirse que se cuenta con los elementos suficientes para abordar la solución propuesta para la problemática existente en la Oficina Nacional de Recursos Minerales.

El Proceso Unificado de Software de Rational (RUP) propone un desarrollo iterativo e incremental a través de sus flujos de trabajo, cada iteración comprende parte de las cuatro fases que propone RUP: inicio, elaboración, construcción y transición. Siempre que se finaliza una iteración tiene como resultado una versión del software acompañada de un conjunto de artefactos. A continuación se muestran los artefactos resultantes correspondientes a las iteraciones de los flujos de negocio y levantamiento de requisitos en la fase de inicio.

3.1 Descripción de los procesos de negocio propuestos

Para la realización de este trabajo fue necesario centrar especial atención sobre los procesos de registro de Definiciones Geológicas y de importación de Bases de Datos que tienen lugar en la ONRM como parte de su sistema de captación de datos geológicos.

3.1.1 El proceso de registro de Definiciones Geológicas.

Es un proceso que comienza al surgir o ser tomado en cuenta un nuevo elemento del mundo real como un dato de interés geológico; inmediatamente se hace necesario georeferenciar dicho elemento, lo cual no es más que el proceso de tomar todos los datos geológicos que contiene como definiciones geológicas las cuales servirán para el posterior almacenamiento de datos específicos de cada uno hallazgos realizados, una vez determinadas las definiciones para este nuevo elemento se solicita el ingreso de estas a la Base de Datos de Definiciones Geológicas.

El especialista geólogo que trabaja en el Centro de Datos es el encargado de registrar los nuevos datos hacia la Base de Datos de Definiciones Geológicas. El nuevo elemento pasa a ser considerado un Rasgo o Tipo de Rasgo, depende del caso, y todos los datos geológicos que él contiene son las diferentes Variables ya sean Virtuales o Elementales.

Una vez almacenados todos los datos referentes al nuevo Rasgo, el sistema se encarga de publicar las nuevas actualizaciones realizadas a la Base de datos.

3.1.2 El proceso de importación de Bases de Datos Geológicas.

Los datos de cada registro geológico son almacenados en Bases de Datos por los especialistas que trabajan en el campo; una vez terminado el proceso de recopilación de dichos datos, este especialista se presenta en la ONRM para realizar la debida incorporación de los mismos al Centro de Datos.

La Base de Datos debe pasar por un proceso de certificación en el cual primeramente se importará la Base de Datos en cuestión ya que la misma se encuentra en un formato ajeno al sistema del Centro de Datos, seguidamente se verificará que no se hallan alterado ninguna de las Definiciones Geológicas involucradas, que los datos estén en el formato adecuado y que la estructura de la Base de Datos no halla sido alterada. Este proceso de certificación es llevado a cabo por el especialista que trabaja en el Centro de Datos. Si este proceso falla la Base de Datos es rechazada y los datos deben ser tomados otra vez.

Si todo transcurrió sin contratiempos la Base de Datos es importada hacia el Centro de Datos, terminando así el proceso.

3.2 Actores del negocio

Un actor del negocio es todo aquel que interactúe con el mismo recíprocamente, pone las demandas o esta interesado en su rendimiento. A continuación se enuncian quienes son los actores del negocio para ese trabajo y la respectiva justificación sobre su elección.

Especialista

Es el encargado de crear el documento que contiene las definiciones pertinentes para el nuevo Rasgo o Tipo de Rasgo a registrar. Cuando dicho documento esta listo este actor se presenta ante el Especialista del Centro de Datos y solicita el ingreso del nuevo elemento a la Base de Datos de Definiciones Geológicas.

Especialista del Campo

Es el geólogo que trabaja directamente en las zonas donde se realizan los estudios (campo), el mismo se encarga de recoger los datos resultantes de dichos estudios en Bases de Batos que contienen las definiciones propias para el elemento que se esta estudiando. Una vez recogidos los datos necesarios este actor se presenta en la ONRM y solicita la certificación de la Base de Datos que contiene a los mismos.

3.3 Diagrama de casos de uso del negocio

El Diagrama de Casos de Uso del Negocio permite de manera visual y representativa mostrar al usuario los diferentes procesos involucrados en el ámbito del problema a resolver, a continuación se muestra en la Figura 3.1 la imagen correspondiente al dicho diagrama, adaptado a las condiciones del problema en cuestión.

Diagrama de Casos de Uso del Negocio

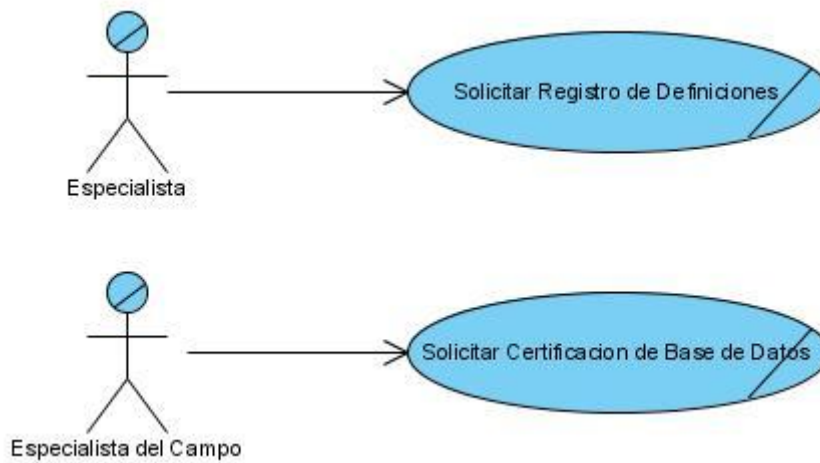


Figura 3.1 – Diagrama de Casos de Uso del Negocio.

Las condiciones para describir paso a paso cada una de las acciones que involucran estos procesos ya están

creadas, seguidamente se muestran las descripciones textuales de los casos de uso del negocio.

3.4 Descripción textual de los casos de uso del negocio.

Dentro de cada caso de uso hay un conjunto de acciones que se realizan en un orden lógico y son llevadas a cabo por diferentes roles participantes en el negocio. El artefacto propuesto por RUP para representar dichas acciones son las descripciones textuales y a continuación se presentan las correspondientes al negocio del problema.

3.4.1 Solicitar Registro de Definiciones.

Nombre del caso de uso del negocio:	Solicitar Registro de Definiciones.
Actores del negocio:	Especialista
Propósito:	Registrar nuevas definiciones geológicas de tipo Rasgo y Tipo de Rasgo; así como todas las concernientes a estos dos conceptos.
Resumen:	Se solicita el registro de nuevas Definiciones para la captación de Datos Geológicos.
Casos de uso asociados:	-
Flujo de trabajo	
Acción del actor	Respuesta del negocio
1 El Especialista solicita el registro de un nuevo Rasgo o Tipo de Rasgo.	2 El Especialista del Centro de Datos solicita la lista de Definiciones correspondiente al nuevo Rasgo o Tipo de Rasgo.
3 El Especialista entrega la lista de Definiciones.	

	4 El Especialista del Centro de Datos registra en la Base de Datos las nuevas Definiciones.
Prioridad:	
Mejoras:	
Cursos alternos: -	

3.4.2 Solicitar Certificación de Base de Datos.

Nombre del caso de uso del negocio:	Solicitar Certificación de Base de Datos.
Actores del negocio:	Especialista del Campo (inicia).
Propósito:	Verificar el estado de las Bases de Datos entrantes para su posterior importación hacia el Centro de Datos.
Resumen:	El caso de uso se inicia cuando el Especialista del Campo presenta al Especialista del Centro de Datos una Base de Datos para su validación. Una vez que el Especialista del Centro de Datos finaliza el proceso de validación de la Base de Datos esta es importada o no hacia el Centro de Datos. El caso de uso finaliza una vez que importa la Base de Datos o se notifica al Especialista del Campo el fallo del proceso.
Casos de uso asociados:	-
Flujo de trabajo	

Acción del actor		Respuesta del negocio
1 El Especialista del Campo entrega un archivo de Base de Datos con los datos recogidos en el campo para su incorporación al Centro de datos.		2 El Especialista del Centro de Datos importa el archivo y analiza que la estructura de la Base de Datos sea la correcta. La estructura es correcta. 3 El Especialista de Centro de Datos importa la Base de Datos hacia el Centro de datos.
Flujo alternativo de trabajo		
Acción del actor		Respuesta del negocio
3 El Especialista de Campo recibe una notificación de que la base de datos ha sido rechazada.		2 El Especialista del Centro de Datos importa el archivo y analiza que la estructura de la Base de Datos sea la correcta. La estructura es incorrecta.
Prioridad:	Crítico	
Mejoras:	El proceso previo a la importación de los datos geológicos se agilizará significativamente. Desaparecerá el riesgo de captar datos incorrectos o Bases de Datos alteradas.	

La representación del diagrama de casos de uso del negocio conjuntamente con las descripciones textuales de cada uno de los casos de uso involucrados sirven como base principal para la selección de los requerimientos que se muestran a continuación.

3.5 Requerimientos funcionales

- 1- Autenticar Usuario.
- 2- Registrar Definiciones Geológicas.
 - 2.1- Registrar Rasgos.
 - 2.2- Registrar Tipos de Rasgos.
 - 2.3- Registrar Perfiles de Trabajo.
 - 2.4- Registrar Variables Virtuales.
 - 2.5- Registrar Variables Elementales.
- 3- Validar las Bases de Datos entrantes al Centro de Datos.
- 4- Importar del módulo de campo Bases de Datos Geológicas hacia el Centro de Datos.

3.6 Requerimientos no funcionales.

Usabilidad

El sistema será utilizado por un único usuario; el especialista geólogo del Centro de Datos de la ONRM.

Rendimiento

El rendimiento del sistema no debe ser seriamente afectado por bajos recursos de hardware.

Software

Todas las herramientas que se utilicen en el desarrollo del sistema deben estar respaldadas por licencias bajo las condiciones de software libre.

Portabilidad

El sistema debe funcionar sobre plataformas Unix, como parte del interés del país en migrar hacia sistemas operativos basados en estas plataformas.

Seguridad

Permitir que a la información solo acceda quien está autorizado para ello y para el uso a que está autorizado.

Identificar a la persona a la que se autoriza, a quien se le concede permiso para determinadas tareas.

Conservar la integridad de la información, es decir, asegurarse de que ésta no se ha transformado durante su almacenamiento o transporte.

Permitir que la contraseña del usuario se almacene y viaje por la red de forma encriptada.

3.7 Descripción del sistema propuesto

3.7.1 Actores del Sistema.

Un actor del sistema es todo aquel que interactúa directamente con este, se beneficia de él pero no pertenece al mismo. Seguidamente se mostrará quienes interactúan con el sistema propuesto.

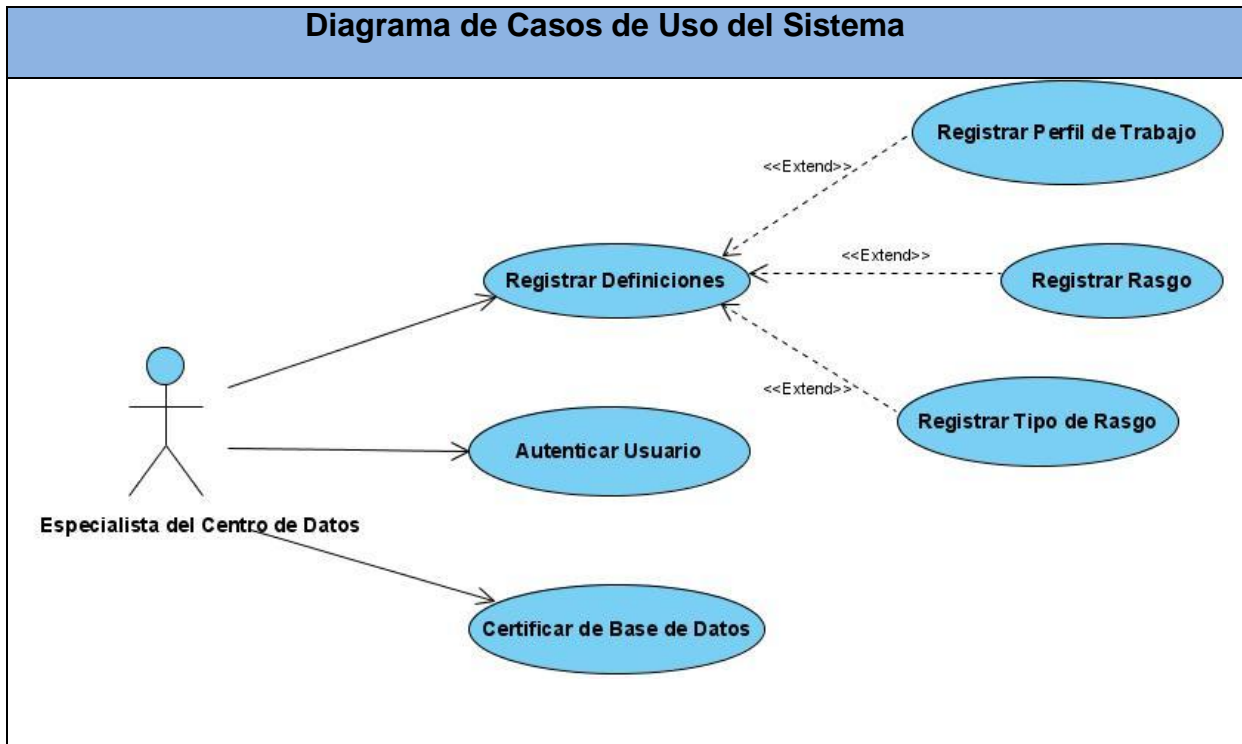
Especialista del Centro de Datos.

Es la única persona que interactúa con el sistema, efectuando todas las operaciones que están disponibles en este.

3.7.2 Diagrama de Casos de Uso del Sistema.

Este diagrama muestra la interacción de los usuarios definidos del sistema y los casos de uso del mismo.

Figura 3.2 – Diagrama de Casos de Uso del Sistema. Seguidamente se procede a describir los casos de uso del sistema.



3.7.3 Descripción textual de los Casos de Uso del Sistema.

Al igual que sucede en el negocio, en los casos de uso del sistema intervienen una serie de acciones realizadas por los actores del sistema las cuales son respondidas por este ultimo; cada una de ellas se describen textualmente para mostrar lograr un mejor acercamiento al funcionamiento del sistema.

3.7.3.1 Autenticar Usuario.

Uso:	Autenticar Usuario
Actores:	Especialista del Centro de Datos
Resumen:	El Especialista del Centro de Datos introduce los datos correspondientes para hacer uso de la aplicación.
Precondiciones:	-
Referencias	R1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
2 El Especialista del Centro de Datos introduce nombre de usuario y contraseña.	1 El sistema muestra la interfaz de autenticación. 3 El sistema verifica que el nombre de usuario y la contraseña introducidos estén correctos. 4 El sistema muestra la interfaz principal.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>3 El sistema verifica que el nombre de usuario y la contraseña introducidos estén correctos. Los datos están incorrectos.</p> <p>4 El sistema muestra un mensaje "Usuario o contraseña incorrectos."</p>
Pos condiciones	La Interfaz principal es mostrada al usuario.

3.7.3.2 Registrar Definiciones.

Caso de Uso:	Registrar Definiciones.	
Actores:	Especialista del Centro de Datos. (inicia)	
Resumen:	El Especialista del Centro de Datos registra las nuevas Definiciones Geológicas que servirán para posteriores estudios de objetivos geológicos en la ONRM.	
Precondiciones:	1 El especialista debe estar autenticado.	
Referencias	R2, R2.1, R2.2, R2.3, R2.4, R2.5, R2.6	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1	El Especialista del Centro de Datos accede a la opción "Registrar" del Menú Principal.	2 El sistema muestra las opciones correspondientes a la selección.
3	El Especialista del Centro de Datos puede: <ul style="list-style-type: none"> • Registrar Rasgo. • Registrar Tipo de Rasgo. • Registrar Perfil de Trabajo. 	
Sección "Registrar Rasgo"		
Acción del Actor		Respuesta del Sistema
1	El Especialista del Centro de Datos selecciona la opción "Registrar Rasgo".	2 El sistema muestra la interfaz correspondiente.
3	El Especialista del Centro de Datos introduce los datos correspondientes a: <ul style="list-style-type: none"> • Rasgo. • Variables Virtuales. • Variables Elementales 	4 El sistema comprueba que los datos estén correctos. 5 El sistema guarda los datos especificados y genera una actualización de las Definiciones.
Sección "Registrar Tipo de Rasgo"		
Acción del Actor		Respuesta del Sistema
1	El Especialista del Centro de Datos selecciona la opción "Registrar Tipo de Rasgo".	2 El sistema muestra la interfaz correspondiente.
3	El Especialista del Centro de Datos introduce los datos correspondientes a: <ul style="list-style-type: none"> • Tipo de Rasgo • Rasgo • Variables Virtuales. • Variables Elementales. 	4 El sistema comprueba que los datos estén correctos. 5 El sistema guarda los datos especificados y genera una actualización de las Definiciones.
Sección "Registrar Perfil de Trabajo"		
Acción del Actor		Respuesta del Sistema
1	El Especialista del Centro de Datos selecciona la opción "Registrar Perfil de Trabajo".	2 El sistema muestra la interfaz correspondiente.
3	El Especialista del Centro de Datos introduce los datos correspondientes a: <ul style="list-style-type: none"> • Perfil de Trabajo 	4 El sistema comprueba que los datos estén correctos. 5 El sistema guarda los datos especificados y genera una

<ul style="list-style-type: none"> • Variables Virtuales. • Variables Elementales. 	<p>actualización de las Definiciones.</p>
Flujos Alternos	
Sección “Registrar Rasgo”	
Acción del Actor	Respuesta del Sistema
	<p>4 El sistema comprueba que los datos estén correctos. Existe algún dato incorrecto.</p> <p>5 El sistema muestra un mensaje “Verifique los datos.”</p>
Sección “Registrar Tipo Rasgo”	
Acción del Actor	Respuesta del Sistema
	<p>4 El sistema comprueba que los datos estén correctos. Existe algún dato incorrecto.</p> <p>5 El sistema muestra un mensaje “Verifique los datos.”</p>
Sección “Registrar Perfil de Trabajo”	
Acción del Actor	Respuesta del Sistema
	<p>4 El sistema comprueba que los datos estén correctos. Existe algún dato incorrecto.</p> <p>5 El sistema muestra un mensaje “Verifique los datos.”</p>
Pos condiciones	Nuevas Definiciones Geológicas son incorporadas a la Base de Datos.

3.7.3.3 Certificar Base de Datos.

Caso de Uso:	Certificar Base de Datos	
Actores:	Especialista del Centro de Datos (inicia)	
Resumen:	El Especialista del Centro de Datos realiza el proceso de Certificación a una Base de Datos Geológica para su posterior importación al Centro de Datos.	
Precondiciones:	<ol style="list-style-type: none"> 1. El especialista debe estar autenticado. 2. El especialista debe recibir el archivo de Base de Datos proveniente del campo. 	
Referencias	R3, R4	
Prioridad	Crítico.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1	El Especialista del Centro de Datos selecciona la opción "Importar Base de Datos" del menú.	2 El sistema muestra la interfaz correspondiente.
3	Busca el archivo a importar.	4 Importa la Base de Datos.
5	El Especialista del Centro de Datos selecciona la opción "Validar".	6 Valida la Base de Datos. El proceso es exitoso.
7	Especialista del Centro de Datos selecciona la opción "Importar hacia Centro de Datos".	8 Envía la Base de Datos al módulo de captación del Centro de Datos.
Flujos Alternos		
Acción del Actor		Respuesta del Sistema
		6 Valida la Base de Datos. El proceso falla.
		7 Muestra un mensaje "Base de Datos alterada."
Pos condiciones	Nuevos datos geológicos son incorporados al Centro de Datos de la ONRM.	

Conclusiones del Capítulo

Los artefactos resultantes en los flujos de trabajo de negocio y requisitos crean una base sólida en el sistema en construcción, generando varios prototipos de confrontación con el cliente para evitar malos entendidos en los flujos de trabajo que le suceden.

Capítulo IV

Construcción de la Solución Propuesta

El diseño es la parte del proceso de desarrollo de software que tiene como objetivo principal definir la estructura del sistema en estudio, siempre basándose en los requisitos funcionales que fueron seleccionados en las etapas anteriores.

En este capítulo se muestran los principales artefactos UML obtenidos en los flujos de trabajo de diseño e implementación según RUP. Finalizando se presentarán el modelo de implementación mediante el diagrama de componentes y de despliegue que resultaron del diseño realizado de cada uno de los casos de uso del sistema.

4.1 Diagramas de clases

- Paquete Autenticar Usuario. (Figura 4.1)
- Paquete Importar Base de Datos. (Figura 4.2)
- Paquete Registrar Rasgo. (Figura 4.3)
- Paquete Registrar Tipo de rasgo. (Figura 4.4)
- Paquete registrar Perfil de trabajo. (Figura 4.5)

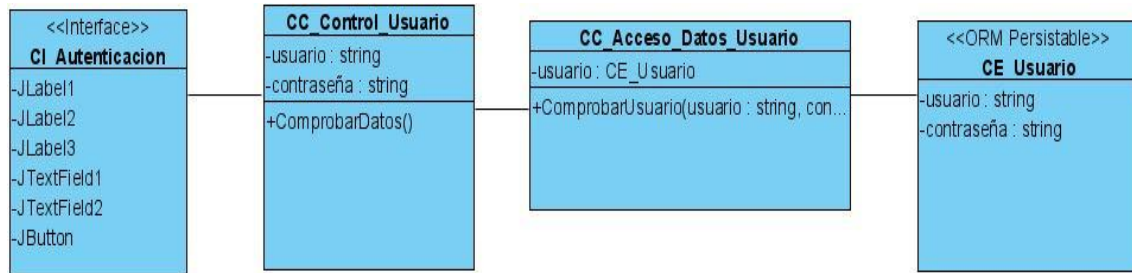


Figura 4.1 – Diagrama de Clases Autenticar Usuario.

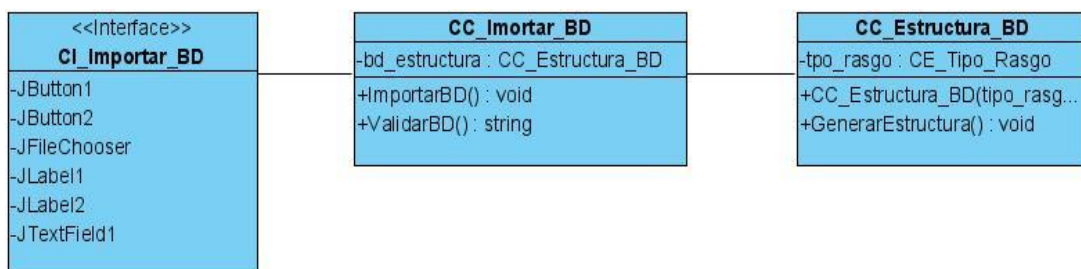


Figura 4.2 – Diagrama de Clases Importar Base de Datos.

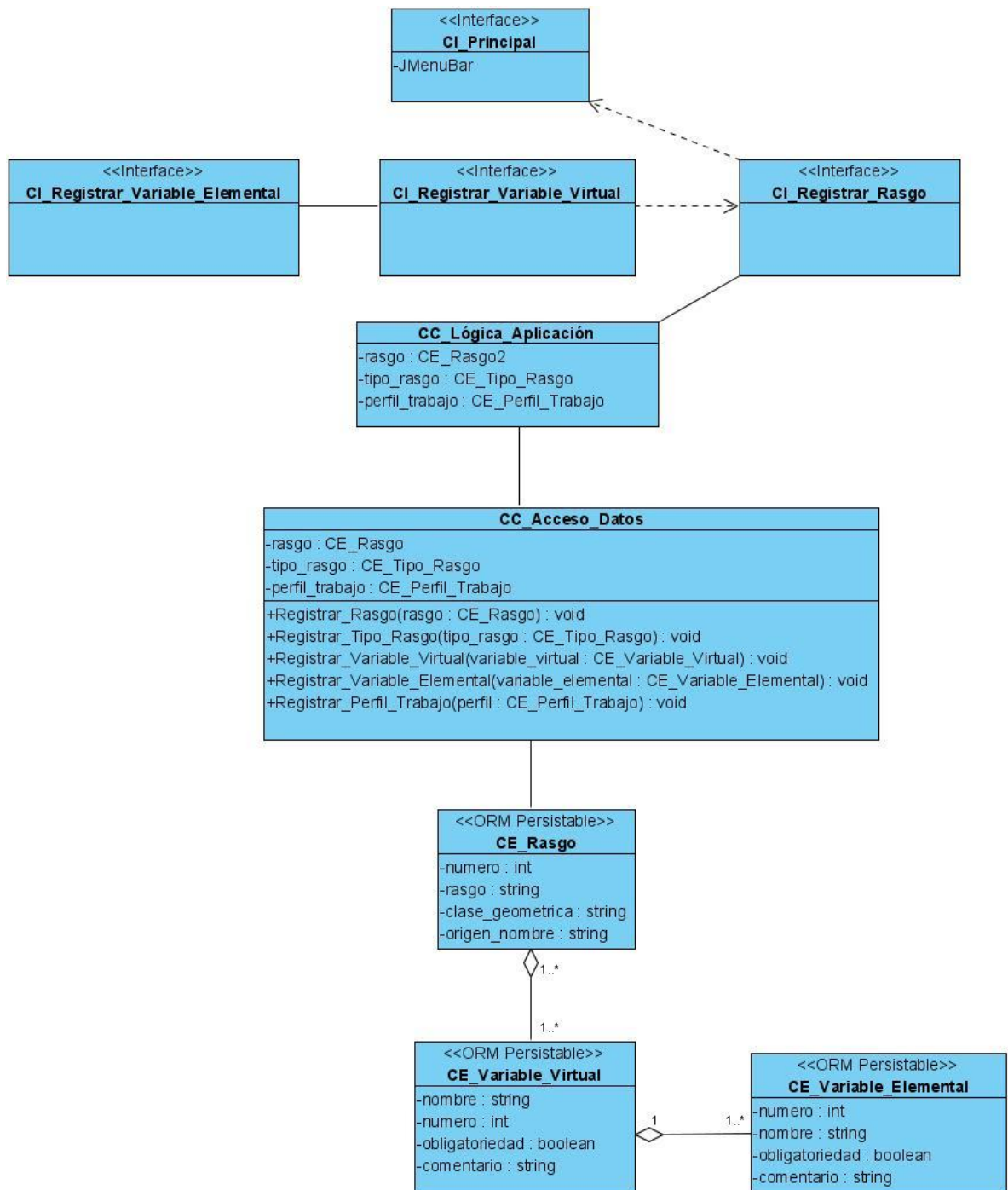


Figura 4.3 – Diagrama de Clases Registrar Rasgo.

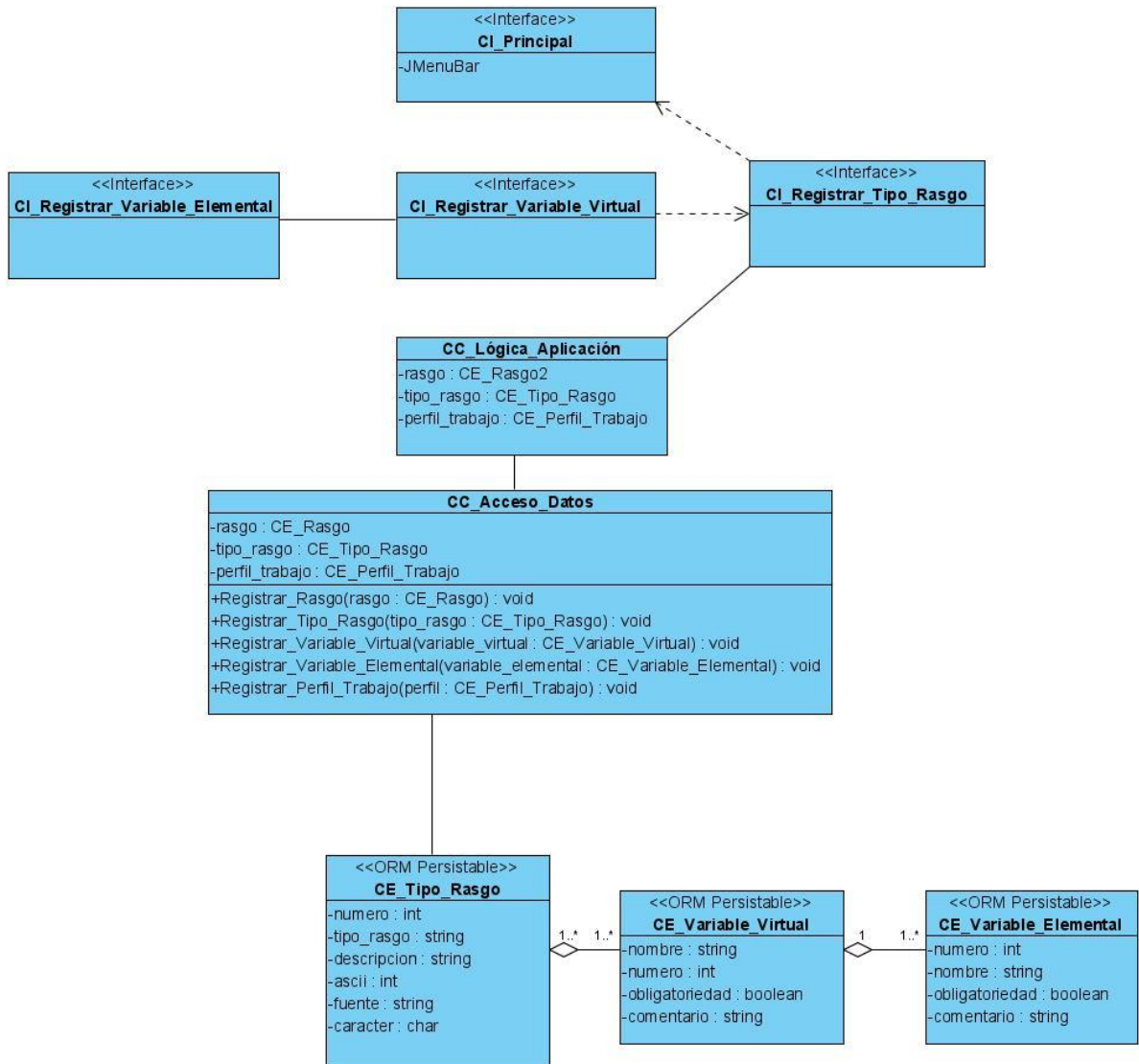


Figura 4.4 – Diagrama de Clases Registrar Tipo de Rasgo.

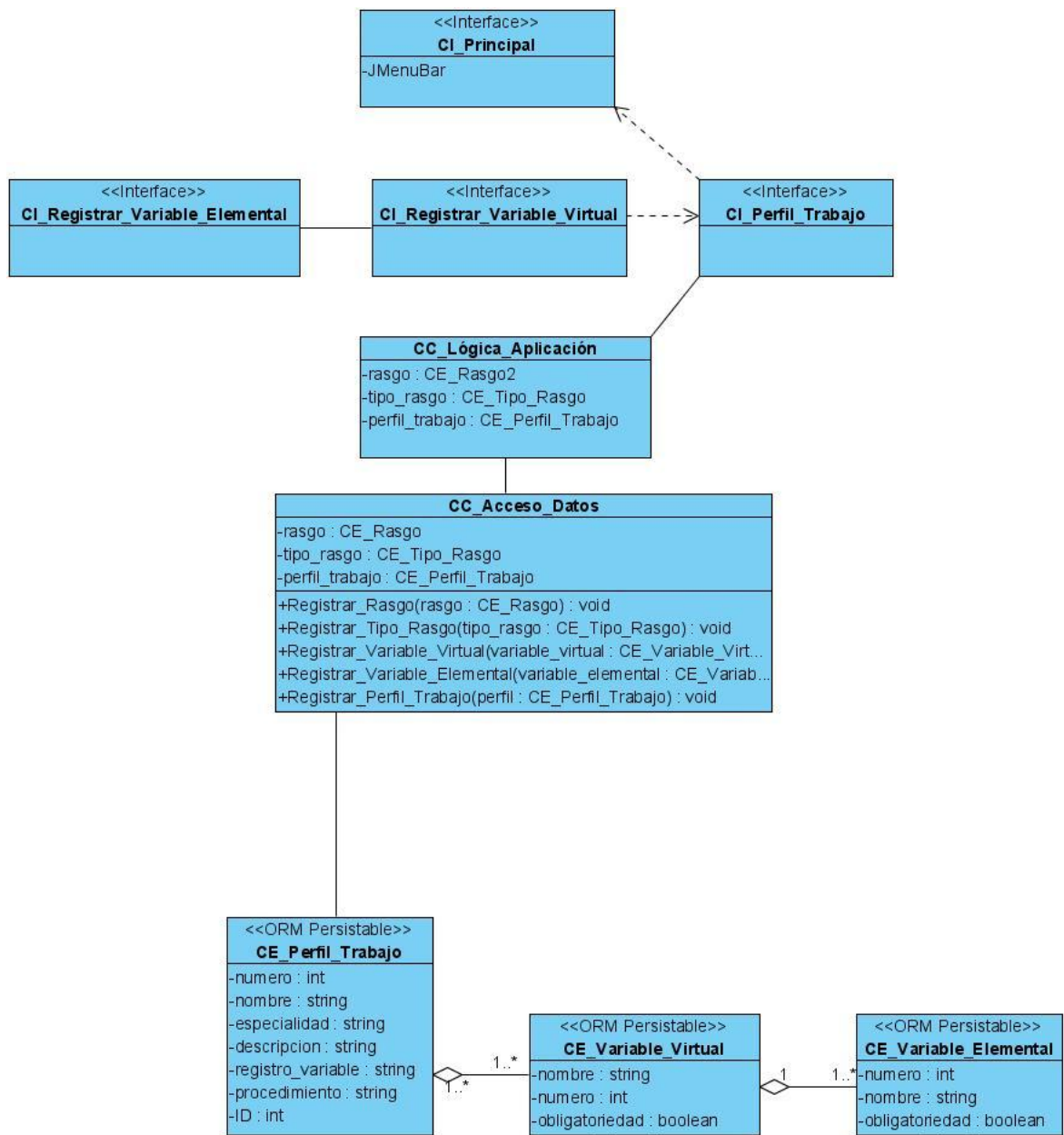


Figura 4.5 – Diagrama de Clases Registrar Perfil de Trabajo.

4.2 Principios del diseño

4.2.1 Estándares en la interfaz de usuario

La interfaz del sistema como pertenece a una aplicación de escritorio se desarrollará guiándose por el actual sistema en uso de la ONRM, adicionándole además nuevos componentes como parte de la nueva concepción del mismo. El logotipo de la entidad estará presente en todas las formas del sistema y se contara con un menú que permita realizar todas las operaciones que dan solución a la problemática.

4.2.3 Tratamiento de excepciones.

El tratamiento de los errores que puedan aparecer durante el uso del software comienza desde los procedimientos almacenados que si detectan algún error detienen su ejecución y retornan al estado anterior, esto es conocido como *RollBack*, luego retorna un código de error a la aplicación que se encargará de manipularlo. Para el trabajo de errores en la interfaz de administración desarrollada para el sistema se definirá una clase Error que contiene todos los mensajes de error que pueden mostrarse al usuario. Cada vez que exista un punto crítico donde pueda aparecer un error se instancia la clase Error, y esta muestra el correspondiente mensaje a modo de información al usuario.

4.2.4 Estándares de codificación.

La estandarización del código de la aplicación se realizará siguiendo las reglas definidas para todas las aplicaciones del Proyecto de Informatización del Conocimiento Geológico (PICG) garantizando un eficiente mantenimiento y la reutilización del código de la aplicación.

Se separará y comentará el código de forma tal que se hiciera más inteligible y se separaron los niveles mediante tabulaciones. Para nombrar las variables se seguirá la regla de escribir los identificadores con letras minúsculas, utilizando como separador para las palabras el carácter “_” tratando de usar nombres sugerentes a la acción de la variable. En el caso de las clases y sus métodos no se usarán abreviaciones y las palabras continuas deben comenzar con mayúsculas y separadas por “_”, para los atributos se utilizaron las mismas normas que para las variables; igualmente para los métodos aunque con estos se usan las mayúsculas. Las declaraciones compuestas (if, while, etc.) no serán anidadas a una profundidad mayor de cinco instrucciones. Para

comentar el código se utilizarán, en el caso de una línea “//”, y para comentarios en bloques “/* */”.

4.3 Diseño de la base de datos.

4.3.1 Diagrama de clases persistentes.

Este diagrama es una especie de modelo lógico de la base de datos pero contiene las clases persistentes, llámese clases persistentes a las tablas que permanecen en la base de datos, borrándose solamente mediante una acción explícita del usuario. En la Figura 4.4 se muestran las clases persistentes del sistema.

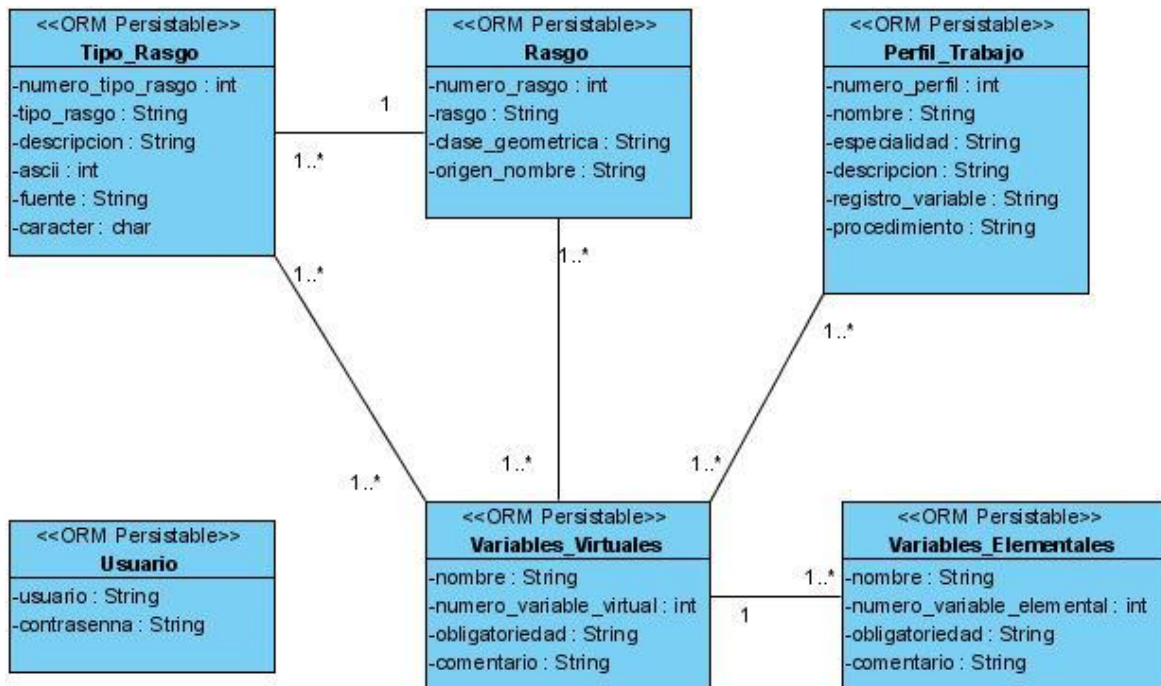


Figura 4.4 – Diagrama de Clases Persistentes.

A partir de este diagrama se obtiene la información necesaria para construir el modelo de datos del sistema.

4.3.2 Modelo de datos.

El modelo de datos no es más que la representación física final de la base de datos obtenida a partir de las clases persistentes como se observa en la Figura 4.5. En el se pueden ver las clases persistentes como tablas de una Base de Datos, sus relaciones y las nuevas tablas creadas a partir de estas últimas.

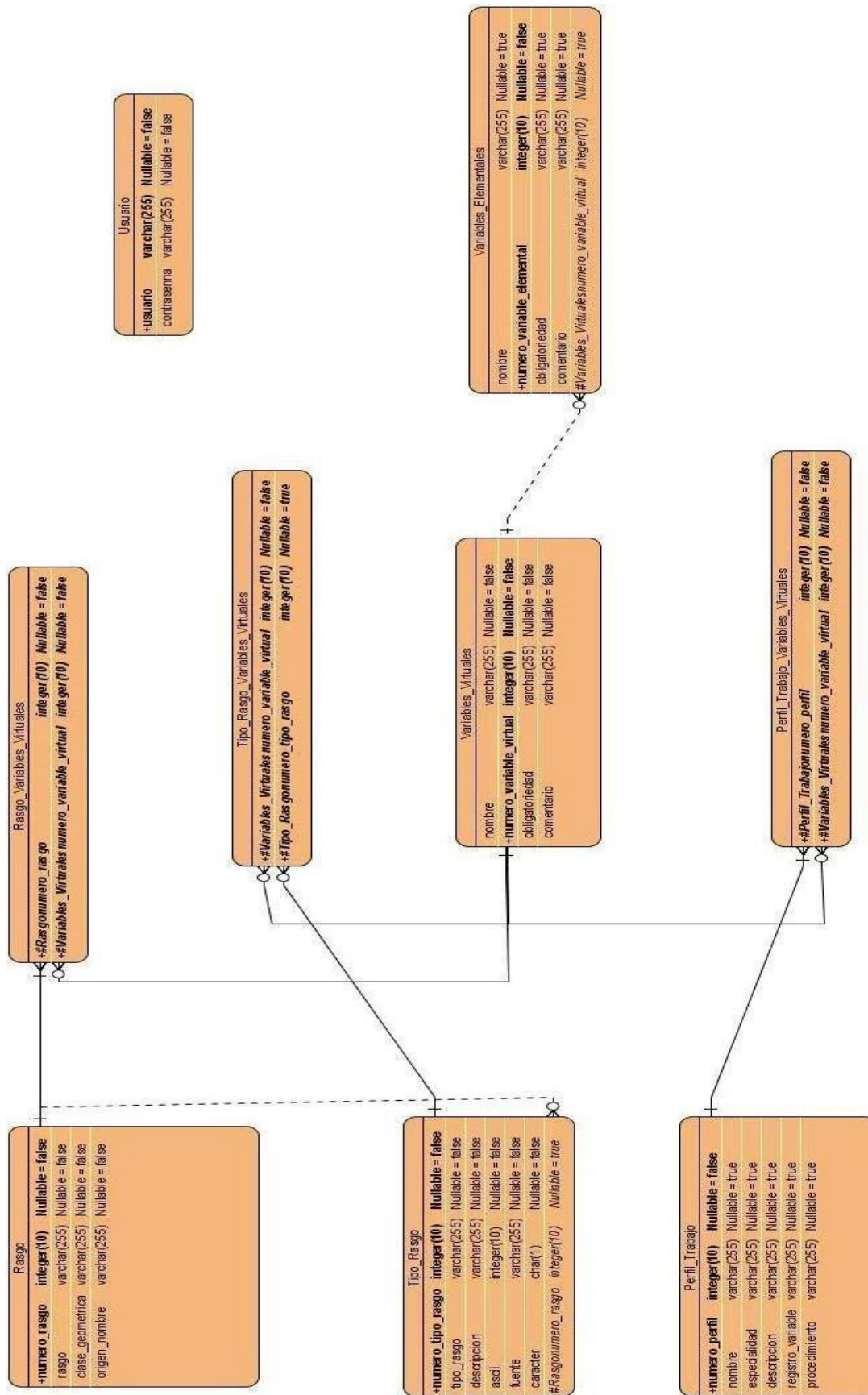


Figura 4.5 – Modelo de Datos.

4.4 Modelo de implementación.

4.4.1 Diagrama de despliegue.

El diagrama de despliegue es el encargado de mostrar los nodos físicos que componen al sistema. Como muestra la Figura 4.6 el sistema esta formado por la PC cliente la cual esta situada en la oficina del Especialista del Centro de Datos y el servidor de Bases de Datos que se encuentra en el Centro de Datos de la ONRM.

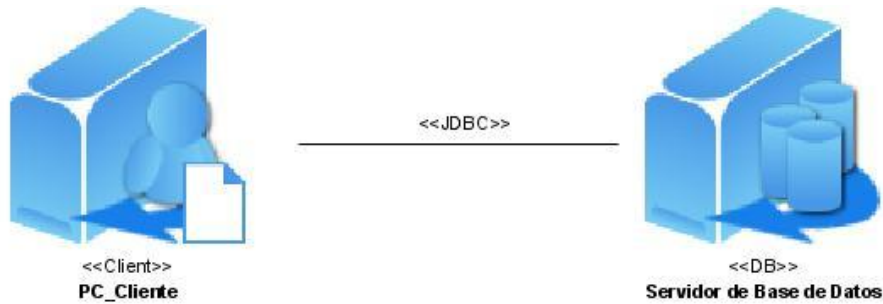


Figura 4.6 – Diagrama de Despliegue.

4.4.2 Diagrama de Componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Es un diagrama que muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones.

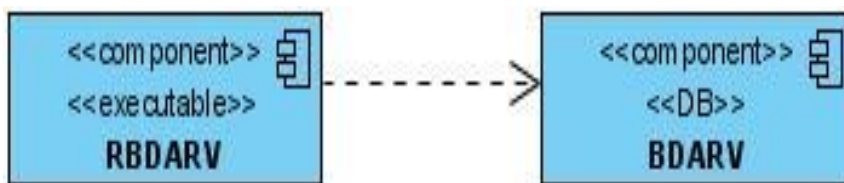


Figura 4.7 – Diagrama de Componentes.

Conclusiones del Capítulo

Una vez realizadas cinco iteraciones a través de los flujos de trabajo que propone RUP, se ha logrado modelar el presente sistema, cualquier programador o estudiante con los mí

nimos conocimientos de ingeniería de software puede utilizar los diagramas antes expuestos como referencia para realizar una operación de mantenimiento, mejora o simplemente como material de estudio, ahora solo queda implementar todo lo expuesto en el transcurso del documento para obtener el producto final.

Capítulo V

Estudio de Factibilidad

El estudio de factibilidad es un paso importante en la realización de un proyecto, pues brinda al equipo de trabajo información inicial relacionada con el costo del producto, tiempo estimado de desarrollo, cantidad de personas que intervienen, entre otros.

En este capítulo se abordarán aspectos relacionados con la estimación de esfuerzos (costes) de desarrollo del sistema. Por último se realizará una valoración de sostenibilidad del producto atendiendo a la dimensión socio-cultural, económica, ambiental y tecnológica.

5.1 Estimación del esfuerzo

El método de estimación mediante el Análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de estos factores.

5.1.1 Cálculo de Puntos de Caso de Uso sin Ajustar.

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente fórmula:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

5.1.2 Factor de Peso de los Actores sin ajustar (UAW).

Para conocer este valor se hacen cálculos mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo Actor	Descripción	Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	1	3

5.1.3 Factor de Peso de los Casos de Uso sin ajustar (UUCW).

El valor se obtiene mediante un análisis de la cantidad de Casos de Uso existentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Tipo de CU	Descripción	Peso	Cantidad	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	2	10
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	1	15

Entonces: $UUCP = UAW + UUCW = 3 + 25 = 28$

5.1.4 Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

5.1.5 Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	1	1
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0,5	3	1,5
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	1	1
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	3	3

TCF = 0,895

5.1.6 Factor de ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo.

$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	2	1
E3	Experiencia en la orientación a objetivos.	1	5	5
E4	Capacidad del analista líder.	0,5	3	1,5
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	2	4
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	1	-1

EF = 0,83

Entonces: UCP = UUCP x TCF x EF = 28 x 0,895 x 0,83= 20,8

De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$E = UCP \times CF$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Para calcular CF

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

$Total_{EF} = Cant_{EF < 3} (entre E1 - E6) + Cant_{EF > 3} (entre E7, E8)$

$$\text{Total}_{EF} = 2 + 0 = 2$$

$$CF = 20 \text{ horas-hombre}$$

$$E = UCP \times CF = 20,8 \times 20 = 416 \text{ horas-hombre}$$

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los Casos de Uso. Para estimar la duración total del proyecto, se distribuye el esfuerzo entre las diferentes actividades del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	182,3095
Diseño	20	364,619
Implementación	40	729,238
Pruebas	15	273,46425
Sobrecarga (otras actividades)	15	273,46425
Total	100	1823,095

Teniendo en cuenta que el tiempo de trabajo planificado para un mes es de 227 horas, el tiempo de desarrollo se estima en aproximadamente 10 meses, como participan 2 personas, en aproximadamente 5 meses se estima que finalice el desarrollo del proyecto.

Para la realización de este proyecto se requirió de un extenso período de investigación, durante el cual se realizó una amplia revisión bibliográfica, debido a que fue necesario realizar un estudio de los estándares conforme a los cuales se implementaría el sistema.

5.2 Beneficios tangibles e intangibles

El desarrollo del software que se propone puede proporcionar una serie de beneficios en el proceso de registrar los datos provenientes del campo en la ONRM.

Además, el uso efectivo de esta herramienta tendría una gran importancia debido a que proporciona una serie de mecanismos que flexibiliza, en términos de tiempo y espacio, la interacción entre los implicados en el proceso.

5.3 Análisis de costos y beneficios

El desarrollo de un producto informático siempre tiene un costo. Este puede estar justificado por los beneficios tanto tangibles como intangibles que origina el mismo.

Además, la tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no es necesario incurrir en gastos en el pago de licencias de uso.

También contribuye al almacenamiento de dichos datos geológicos en una base de datos que garantiza el óptimo estado de los datos.

Por ello se plantea que es factible el desarrollo del mismo.

5.4 Valoración de sostenibilidad según la dimensión socio-cultural, económica, ambiental y tecnológica.

La explotación de una aplicación para el registro de Definiciones en la ONRM, tiene una influencia positiva desde el punto de vista económico, debido a que proporciona un método para mantener seguros los datos geológicos provenientes del campo, ofreciendo una nueva alternativa y un sistema de amplia disponibilidad para la capacitación de datos.

Además, en el momento en que la aplicación se ponga en práctica por la ONRM, esta aplicación puede contribuir positivamente en el proceso de registro de Definiciones.

Por otra parte, en la realización de toda investigación se debe tener presente el impacto medioambiental del trabajo y de las soluciones propuestas, visto éste como el conjunto de alternativas positivas y/o negativas que sufren los elementos del entorno producto del funcionamiento y desarrollo.

Tomando como punto de partida este concepto, se puede plantear que la realización de este trabajo no impacta negativamente el entorno, exceptuando la pequeña e indirecta influencia que pueda ejercer el consumo de electricidad y recursos empleados en la investigación.

La interfaz está diseñada de forma que facilita la navegabilidad de la aplicación, con uso de colores claros y contrastados. Además de no usarse recursos que contribuyan a desviar la atención de la actividad principal.

En cuanto a los beneficios económicos que este trabajo traerá consigo es necesario plantear que resulta impreciso y en extremo engorroso arribar a un estimado fidedigno. Cabe destacar que contribuye al ahorro de una enorme cantidad de recursos materiales y financieros que se gastarían si se tuvieran que pagar todas las licencias ya que la antigua aplicación se ejecutaba sobre plataforma Delphi y la base de datos sobre SQL Server 2000. Es recomendable realizar previamente una identificación de las verdaderas necesidades que presentan los profesionales que recibirán los mismos, y de la demanda existente de un conocimiento específico.

En cuanto al mantenimiento del producto, debido al uso de estándares se facilita la interoperabilidad y la reusabilidad de los recursos, lo cual beneficia directamente la migración de datos y la implementación de nuevas funcionalidades.

Finalmente, se considera que la solución propuesta contribuye al desarrollo, con un impacto positivo en las distintas esferas involucradas.

Conclusiones del Capítulo

En este capítulo se analizó la factibilidad de realización del sistema, se calculó el costo de producción del mismo, el tiempo que se estimó en que debía estar listo, el esfuerzo que debía realizar el equipo de desarrollo y la cantidad de personas necesarias para la realización del sistema, permitiendo confirmar la factibilidad de la construcción del sistema propuesto; y por último se concluyó que el sistema influye positivamente en el desarrollo sostenible.

Conclusiones

Con la finalización de este trabajo se puede concluir que:

Que el sistema concebido al término de los flujos de análisis y diseño resuelve la problemática planteada ya que cuenta con la documentación suficiente, en su concepción se han utilizado mecanismos que lo dotarán de una calidad óptima y su uso será único a nivel nacional.

Ya es posible implementar el Sistema para el Registro e Importación de Definiciones y Bases de Datos Geológicas. Quedando así cumplidos los objetivos trazados para la realización del mismo.

A través del estudio que se llevo a cabo se encontró la necesidad de separar la Base de Datos Relacional de la Base de Datos del Centro de Datos lo cual permitirá llevar a cabo un mayor control sobre la estructura de las Bases de Datos entrantes al centro, y ahorrara múltiples operaciones que se realizaban innecesariamente, garantizando así la optimización del sistema.

Todos los artefactos generados brindarán, una vez desarrollado el sistema, la posibilidad de realizar operaciones de mejora o manteniendo al mismo; cumpliendo así con una de las exigencias mayores hechas por la ONRM.

Con la incorporación de la importación de Bases de Datos Geológicas el sistema concebido se convertirá en la herramienta fundamental para la captación de datos geológicos, ya que evitará la incorporación de datos erróneos, innecesarios o de menos al Centro de Datos de la entidad en cuestión.

El sistema ha sido concebido para que funcione solo en la Oficina Nacional de Recursos Minerales, pero la Base de Datos Relacional puede ser adaptada a cualquier entorno ya que posee gran generalidad y puede ser utilizada para la creación dinámica de bases de datos de todo tipo de información sea o no ésta geológica.

Recomendaciones

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- Utilizar todos los artefactos generados en este trabajo para la implementación del sistema concebido.
- Implementar un sistema que cree las Bases de Datos provenientes del campo en formato de PostgreSQL para facilitar su importación hacia el Centro de Datos.
- Analizar la posibilidad de que en un futuro las Definiciones Geológicas de la tecnología BDRAV puedan ser modificadas una vez registradas en la Base de Datos e integrar dicha funcionalidad al sistema.
- Implementar en futuras versiones del sistema un Servicio Web (Web Service) que permita saber de forma on-line (en línea) las nuevas Definiciones Geológicas disponibles en la Base de Datos Relacional para que sean usadas por otros sistemas.
- Utilizar la tecnología BDRAV en proyectos similares ya que la estructura brindada por la misma permite gran adaptabilidad a cualquier problemática.

Glosario de Términos

ONRM: Oficina Nacional de Recursos Minerales.

PICG: Proyecto de Informatización del Conocimiento Geológico.

UCI: Universidad de las Ciencias Informáticas.

ONU: Organización de Naciones Unidas.

ISO: Organización internacional para la Normalización.

IGP: Instituto de Geología y paleontología.

SIGEOL: Sistema Informático geológico.

BDRAV: Base de Datos Relacional con Atributos y Registros Variables.

GeoDataIC: Programa Informático dentro de la ONRM.

DBConvert: Sistema Informático para la Importación de Bases de Datos.

Kexi: Sistema Informático para la Importación de Bases de Datos.

ECMA: la Asociación de Constructores de Computadoras Europea.

RUP: Proceso Unificado de Rational (Metodología de Desarrollo de Software).

Referencias Bibliográficas

DMSOft Technologies. (s.f.). *db Convert*. Recuperado el 1 de 2008, de DBConvert for Access & PostgreSQL : <http://dbconvert.com/>

España, S. C. (2008). *Business Consulting Services*. Obtenido de http://www.sowre.es/wps/portal!/ut/p/kcxml/04_Sj9SPykssy0xPLMnMz0vM0Y_QjzKLN4g3dAoCSYGYxqb6kWhCjgiRIH1vfV-P_NxU_QD9gtzQiHJHR0UAism46A!!/delta/base64xml/L0IDU0IKQ1RPN29na21BISEv b0VvUUFBSVFnakZJQUFRaENFSVFqR0VBLzRKRmIDbzBlaDFpY29uUVZHaGQtc0LRIS83XzBfMU9KLzE

Geominera Oriente. *Documentación Técnica GeoDatoICV2.0*.

Gracia, J. (7 de mayo de 2005). *Desarrollo de Software Orientado a Objetos*. Obtenido de <http://www.ingenierosoftware.com/analisisydiseno/uml.php>

Gutierrez, I. d. (s.f.). *El Mundo Informático*. Obtenido de <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>

Kexi, E. (16 de Octubre de 2006). *The kexi project*. Obtenido de <http://kexi-project.org/wiki/wikiview/index.php@Welcome.html>

León Pavón, E. (2 de abril de 2007). *Blog de Eduardo León*. Obtenido de Visual Paradigm, una herramienta de lo más útil: <http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>

Ministerio del Poder Popular para Ciencia y Tecnología. (s.f.). Obtenido de <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>

Molina, I. E. (2008). *Soluciones Linux*. Obtenido de <http://www.solucioneslinux.com.mx/index.php?option=content&task=view&id=3&Itemid=25>

Sitio oficial de Sun Microsystems. (30 de enero de 2008). Obtenido de http://blogs.sun.com/SalimAfiune/entry/que_es_java

Tecnología, D. d. (s.f.). *Colegio San José SS.CC*. Obtenido de http://www.colegiosanjosesscc.org/ArchivosColegiosSanJose/SanJose/Archivos/documentos%20p%C3%A1gina/Copister%C3%ADa/Diccionario%20_C,D_.pdf

Tecnología, D. d. *Informática Aplicada*. Colegio San José SS.CC.

Tiobe Software. (s.f.). Obtenido de <http://www.tiobe.com/index.php/content/company/Home.html>

Zero, P. C.

Bibliografía

Dieguez Escalona, J. L. (2006). *Sistema de Seguridad Centralizada*.

Englander, R. *Developing Java Beans*.

Espinosa Hidalgo, A. (2004). *SISTEMA PARA LA ADMINISTRACIÓN UNIFICADA DE USUARIOS*.

Flanagan, D. *Java en pocas palabras*.

Larman, C. *UML Y PATRONES. INTRODUCCION AL ANALISIS Y DISEÑO ORIENTADO OBJETOS*.

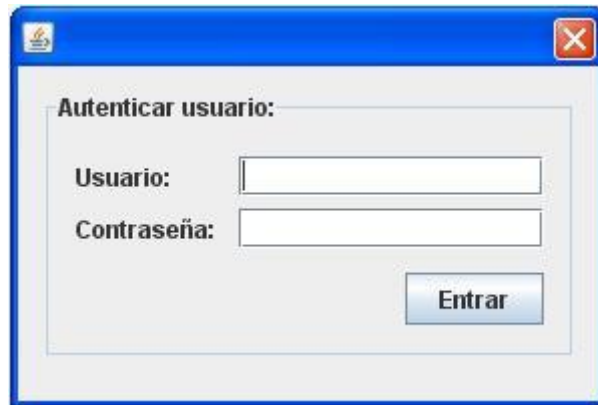
Pressman, R. S. (2005). *Ingeniería de Software un Enfoque Práctico*.

Rumbaugh, J., Jacobson, I., & Booch, G. *El Lenguaje Unificado de Modelado. Manual de referencia*.

Geominera Oriente. *Documentación Técnica GeoDatoICV2.0*.

Anexos

Anexo 1



A screenshot of a Windows-style dialog box titled "Autenticar usuario:". It contains two text input fields: "Usuario:" and "Contraseña:". Below the fields is a button labeled "Entrar". The dialog box has a blue title bar with a close button (X) in the top right corner.

Interface de Autenticación de Usuario.

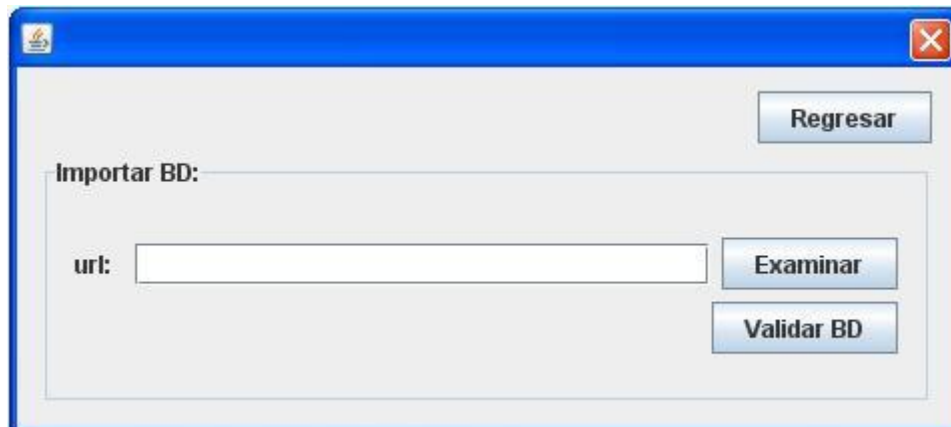
Anexo 2



A screenshot of the main application window. It has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. Below the title bar is a menu bar with three items: "Inicio", "Registrar", and "Ayuda". The main content area is currently empty.

Interface Principal de la Aplicación.

Anexo 3



A screenshot of a dialog box titled "Importar BD:". It features a "Regresar" button in the top right corner. Below the title is a text input field labeled "url:". To the right of the input field are two buttons: "Examinar" and "Validar BD".

Interface de Importación de Base de datos Geológica.

Anexo 4

Regresar

Rasgo

numero:

rasgo:

clase geometrica:

origen nombre:

Variables virtuales:

Crear WV

variables: **Adicionar**

Item 1
Item 2
Item 3
Item 4
Item 5

Aceptar

Interface de Registro de Rasgo.

Anexo 5

Tipo de Rasgo:

numero:

tipo de rasgo:

descripcion:

ascii:

fuente:

caracter:

Rasgo **Variables Virtuales**

Rasgos:

Crear R

rasgos:

Aceptar

Registrar

Interface de Registro de Tipo de Rasgo.

Anexo 6

Regresar

Perfil de Trabajo:

ID: numero:

nombre:

especialidad:

descripcion:

registro variable:

procedimiento:

variables virtuales

Crear VV

variables virtuales: Adicionar

Item 1
Item 2
Item 3
Item 4
Item 5

Aceptar

Interface de Registro de Perfil de Trabajo.

Anexo 7

variables virtuales:

nombre: jTextField1

numero: jTextField2

obligatoriedad: Item 1

comentario:

variables elementales:

Crear VE

nombre de la variable elemental:

Variable elemental 1

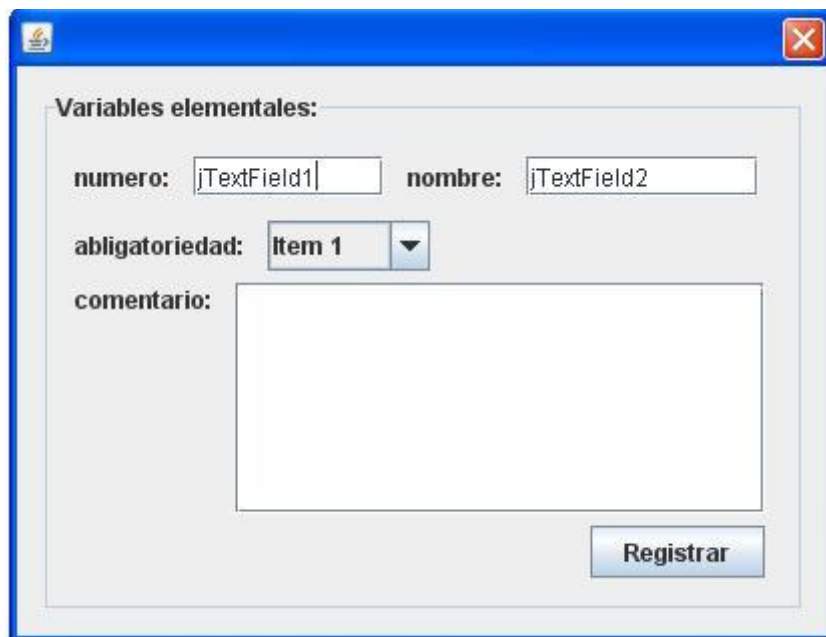
Adicionar

Variable elemental 1
Variable elemental 2

Registrar

Interface de Registro de Variable Virtual.

Anexo 8



The image shows a software window titled "Variables elementales:" with a blue border and standard window controls (minimize, maximize, close). Inside the window, there are four input fields: "numero:" with a text box labeled "jTextField1", "nombre:" with a text box labeled "jTextField2", "obligatoriedad:" with a dropdown menu showing "Item 1", and "comentario:" with a large empty text area. A "Registrar" button is located at the bottom right of the form area.

Interface de Registro de Variable Elemental.