



Universidad De Las Ciencias Informáticas.

Facultad # 9.



“Prototipo no funcional del Sistema Gestión de Definiciones para la Oficina Nacional de Recursos Minerales”

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

Autores:

- Iralis Ruiz Torres
- Anny Yisel Torres Martínez

Tutor:

- Ing. Eyllin Hernández Luque

Co-tutores:

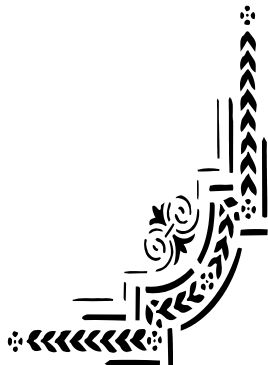
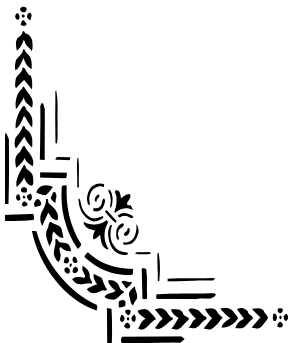
- Ing. Jorge Luis Diéguez Escalona
- Dr. César Rosales

Revisor:

- Dr. José Ortiz Rojas.



Junio del 2008



“El futuro de Cuba tiene que ser necesariamente un futuro de hombres de ciencia, de hombres de pensamiento.”

Fidel Castro



A mi madre a quien amo más que a todo...

A ese hombre especial que puso Dios en mi camino.

A mis tíos, primos, abuelos...

A mis amigos y a mí...

Annita.

A mi familia y a mi marido por haberme apoyado en todo momento para lograr realizar este sueño que era de todos.

Además a mi bebé que está por nacer, por ser mi mayor fuente de inspiración en todo momento.

Inalis.

De Anny:

Tengo tantas personas a las cuales agradecer que no sabría por donde comenzar , y quiero dar gracias a Dios por esto , pues me considero una persona afortunada por contar con un grupo de amigos que ha estado dispuesto a todo por mí , quiero dar gracias a mi compañera de tesis por tenerme tanta paciencia y a mis profes por proporcionarme los conocimientos necesarios para llegar hasta aquí y sobre todo estoy muy agradecida con la Revolución y con Fidel por esta oportunidad que me dio de poder estudiar en una universidad como esta. A nuestra tutora por apoyarnos y ayudarnos para poder llegar a realizar este trabajo, así como al Ing. Armando Ortiz Cabrera, al Msc. Ing. Febe Angel Ciudad Ricardo y al Dr. José Ortiz Rojas por su contribución al mismo.

A mi familia a la cual considero la mejor del mundo, pues si naciera otra vez quisiera una vez mas formar parte de un núcleo familiar tan especial. A mis tíos que tanto apoyo me han dado, en especial a mi tía Idalmis que ha sido como una madre.

A mi abuelita que tanto me ha apoyado y que tanto a pedido a Dios por mí y a la que amo con todas las fuerzas de mi ser, por ser quien es y como es, gracias abuela. A alguien a quien le debo tanto que no podría pagarle ni con otras mil vidas, a la autora intelectual de mi vida, al motor que me impulsa cada día, a esa persona que es mi familia, mi amiga, mi madre, mi todo, mamita gracias por ser la luz que me guía, no se que habría sido de mí sin tu apoyo, Te amo.

Agradezco infinitamente a mi novio por haberme mostrado que la vida tiene otros colores , por sacar lo mejor de mí, por hacerme sentir una persona afortunada sólo por tenerlo a mi lado, al igual que a su familia a la que considero como mía, a Leonel Tomas por confiar en mí cuando ni siquiera yo misma confiaba .

A mi misma por tener la fuerza y la voluntad para superar cada obstáculo que encontré en el camino.

A todos gracias por formar parte de mi vida...

De Irailis:

Hoy quisiera agradecer a tantas personas por ayudarme a realizar este gran sueño que no sabría por donde empezar, pero no puedo dejar de mencionar a:

A todos mis compañeros de aula y amigos, en especial a mi compañera de tesis por su apoyo y por haberme devuelto el aliento en los momentos que lo perdí.

A todos los profesores que durante mi vida de estudiante han aportado su granito de arena para mi desarrollo intelectual.

A nuestra tutora por ayudarnos y defendernos hasta el final.

A toda mi familia y a la familia de mi marido, en especial a dos personas que han hecho de mi vida algo mejor, a las que siempre estaré agradecida por formar parte de mi vida: Mi madre y mi hermana.

A alguien que no sabría de que manera demostrarle lo mucho que le agradezco por todos los momentos maravillosos que le ha dado a mi vida, a ese alguien que sin ser familia a sabido comportarse como tal, a ese que tantas veces a secado mis lágrimas y con sus actos a sacado de mi una sonrisa, a el quiero darle las gracias desde lo mas profundo de mi corazón: Mi marido, Onay te amo.

También quiero agradecer a esta Revolución por su obra maravillosa y a Fidel por crear esta universidad y por darme la oportunidad de estudiar en ella.

A otras muchas personas que de una manera u otra hicieron posible que yo pudiera llegar hasta aquí, dándome apoyo en estos 22 años de vida.

A todos gracias de corazón por confiar en mí...

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Anny Yisel Torres Martinez
Autor

Iralis Ruiz torres
Autor

Ing. Eyllin Hernández Luque
Tutora

OPINIÓN DEL TUTOR

Hoy día, el mundo está en constante cambio y evolución tecnológica; nuestro país se ha incorporado a dicho desarrollo informático, sumándose todas las Entidades y Ministerios, entre los que se encuentra el Ministerio de la Industria Básica (MINBAS) y como oficina adjunta a la misma, la Oficina Nacional de Recursos Minerales (ONRM). Esta es una de las razones que permite expresar que la investigación desarrollada presenta actualidad y es de vital importancia para el mejoramiento de la gestión del conocimiento geológico en Cuba.

Las diplomantes Anny Yisel Torres Martinez e Iralis Ruiz Torres, mostraron cuidado en la realización de la investigación científica, cumplimiento con el objetivo trazado en el trabajo. Las estudiantes mostraron autonomía en la tarea a realizar, logrando documentar y modelar su investigación; trabajando de manera planificada haciendo uso de las tareas investigativas, para centrar y organizar su trabajo.

El análisis bibliográfico es actual y fundamenta correctamente la necesidad del estudio realizado. El diseño de la investigación se ajusta a la problemática que se aborda, existiendo correspondencia entre el problema a desarrollar, los objetivos planteados así como la idea a defender, lo que demuestra dominio del método científico.

El documento realizado posee organización y está basado en métodos investigativos científicamente aprobados. La modelación del análisis y diseño de la aplicación cumple con el objetivo trazado, lo cual brindan beneficios sociales y económicos para la gestión del conocimiento geológico en la Oficina Nacional de Recursos Minerales.

Por todo lo anteriormente expresado, considero que las estudiantes están aptas para ejercer como Ingeniero en Ciencias Informáticas y propongo que se le otorgue al Trabajo de Diploma la calificación de 5 puntos.

Ing. Eylín Hernández Luque

Tutora

Junio del 2008

Fecha

RESUMEN

La Informatización de la Sociedad es el proceso de utilización ordenada y masiva de las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, para satisfacer las necesidades de todas las esferas de la sociedad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos, pero más importante que esto es una informatización inteligente y de esto se trata justamente la utilización de tecnologías catalogadas como gratuitas y libres. La Oficina Nacional de Recursos Minerales no se encuentra exenta de esto por lo que opta por migrar la aplicación que se encarga de la gestión de Definiciones hacia un software libre, justamente en el transcurso del presente documento se definen y desarrollan un conjunto de artefactos que hacen posible que se le pueda dar cumplimiento a esta problemática. Durante el primer capítulo se define todo el marco teórico y se hace alusión a la palabras más significativas, que pueden ser importantes para poder entender mejor el problema en cuestión, así como en el capítulo dos se definen un conjunto de tecnologías que permiten realizar un correcto desarrollo del diseño eficiente de la aplicación que se encarga de la gestión de Definiciones para la PocketPC. El capítulo 3 comprende las características del sistema, realizándose la presentación de la solución propuesta luego de realizarse un minucioso trabajo para garantizar una correcta interpretación del negocio y el sistema en cuestión. En el capítulo 4 es donde se efectúan el análisis y el diseño de dicha solución. El capítulo 5 refleja el estudio de la factibilidad, la planificación, los costos y los beneficios. De esta manera quedan reflejados en el presente documento de manera detallada un conjunto de Definiciones y artefactos que hacen posible dar una eficiente solución al problema planteado.

PALABRAS CLAVES:

- Informatización.
- ONRM.
- Software.
- PocketPC.

Figuras

<i>Figura 1: Esquema general del GeoDato.</i>	5
<i>Figura 2: Vista general de RUP.</i>	15
<i>Figura 3: Arquitectura por Capas.</i>	20
<i>Figura 4: Modelo de Caso de Uso del Negocio.</i>	29
<i>Figura 5: Modelo de Objetos del Negocio.</i>	32
<i>Figura 6: Diagrama de Casos de Usos del Sistema.</i>	34
<i>Figura 7: Diagrama de Clases CU <Crear Definición >.</i>	42
<i>Figura 8: Diagrama de Clases CU <Exportar Definición Creada>.</i>	43
<i>Figura 9: Diagrama de Clases CU <Importar Definición Captada>.</i>	43
<i>Figura 10: Diagrama de Clases CU <Visualizar Datos Existentes>.</i>	43
<i>Figura 11: Diagrama de Clases de Diseño General.</i>	46
<i>Figura 12: Tratamiento de excepciones.</i>	49
<i>Figura 13: Diagrama de Despliegue.</i>	50
<i>Figura 14: Diagrama de Componentes.</i>	51

Tablas

<i>Tabla 1: Descripción del Actor del Negocio.</i>	29
<i>Tabla 2: Descripción del Trabajador del Negocio.</i>	29
<i>Tabla 3: Descripción del CUN<Crear Definiciones para la PocketPC>.</i>	30
<i>Tabla 4: Descripción del CUN<Archivar las planillas de datos geológicos captados>.</i>	31
<i>Tabla 5: Descripción de los Actores del Sistema.</i>	34
<i>Tabla 6: Caso de Uso <Crear Definición>.</i>	36
<i>Tabla 7: Caso de Uso <Exportar Definición Creada>.</i>	37
<i>Tabla 8: Caso de Uso <Importar Definición Captada>.</i>	38
<i>Tabla 9: Caso de Uso <Visualizar Datos Existentes>.</i>	39
<i>Tabla 10: Descripción de los componentes del Diagrama de Despliegue.</i>	50

RESUMEN	IX
Figuras.....	X
Tablas.....	X
INTRODUCCIÓN	1
CAPÍTULO 1	8
Fundamentación Teórica.....	8
1.1 Conceptos que ayudan a entender el problema	8
1.1.1 Definición.....	8
1.1.2 Rasgos.....	8
1.1.3 Tipos de Rasgos.....	9
1.1.4 Perfiles de Trabajo.....	9
1.1.5 Variables Virtuales.....	9
1.1.6 Variables Elementales	9
1.1.7 Puesta de Trabajo.....	9
1.2 Proceso de gestión de las Definiciones para una PocketPC	10
1.2.1 Problema y situación problemática.....	10
1.2.2 Análisis de los resultados de las técnicas aplicadas.....	10
1.2.3 Información sobre las PocketPC.....	10
1.2.4 Existencia de soluciones semejantes a esta investigación	11
1.2.4.1 Concepción General del Sistema.....	11
1.2.4.2 Características.....	11
1.2.4.3 Deficiencias.....	11
1.3 Conclusiones parciales.....	11
CAPÍTULO 2	13
Tendencias y tecnologías a utilizar	13
2.1 Lenguaje Unificado de Modelado (UML)	13
2.2 Metodología de desarrollo de Software	14
2.2.1 Proceso Unificado de Desarrollo (RUP)	14
2.2.2 Programación Extrema (XP)	15
2.2.3 Desarrollo Guiado por Funcionalidades (FDD)	16
2.2.4 Selección de la Metodología a utilizar	17
2.3 Software Libre y su utilización en Cuba	17
2.3.1 Linux	18
2.3.1.1 Distribución Ubuntu	19
2.4 Arquitectura por Capas.....	20

2.5 Aplicación de escritorio como vía alternativa	21
2.6 Fundamentación del Lenguaje que se va a utilizar (Java)	23
2.6.1 Características del Java 5	24
2.7 Herramientas a utilizar	24
2.7.1 Visual Paradigm.....	24
2.7.2 NetBeans	25
2.8 Conclusiones Parciales	26
CAPÍTULO 3	27
<i>Descripción de la solución propuesta</i>	<i>27</i>
3.1 Análisis del problema actual.....	27
3.2 Modelo del Negocio propuesto	28
3.2.1 Reglas generales del Negocio.....	28
3.2.2 Actores del Negocio	28
3.2.3 Trabajador del Negocio	29
3.2.4 Modelos de Casos de Uso del Negocio.....	29
3.3 Descripción de los procesos del negocio propuesto	30
3.4 Modelo de objetos del Negocio	31
3.5 Requisitos del Sistema.....	32
3.5.1 Requisitos funcionales.....	32
3.5.2 Requerimientos no funcionales	33
3.6 Propuesta de Solución	33
3.7 Conclusiones parciales.....	40
CAPÍTULO 4	41
<i>Diseño de la solución propuesta.....</i>	<i>41</i>
4.1 Modelo de Análisis	41
4.1.1 Diagrama de Clases del Análisis	42
4.2 Diagrama de Interacción.....	43
4.2.1 Diagrama de Secuencia	44
4.2.2 Diagrama de Colaboración.....	44
4.3 Diagrama de Clases de Diseño	44
4.4 Estructura del Fichero (XML)	47
4.5 Principios de Diseño	47
4.5.1 Estándares en la interfaz de la aplicación.....	48

4.5.2 Tratamiento de excepciones.....	49
4.7 Diagrama de Despliegue	49
4.8 Diagrama de Componentes	50
4.9 Conclusiones parciales.....	51
CONCLUSIONES GENERALES	XIV
RECOMENDACIONES	XV
Bibliografía	XVI
GLOSARIO DE TERMINOS.....	XIX
ANEXOS	XXII
ANEXO 1: Entrevista realizada a un directivo de la Oficina Nacional de Recursos Minerales.	XXII
ANEXO 2: Diagramas de actividades de los casos de uso del negocio.	XXIII
ANEXO 3: Prototipos de Interfaces de Usuarios de los Casos de Uso críticos.	XXV
ANEXO 4: Diagramas de Interacciones	XXIX
ANEXO 5: Modelos de Diseño.	XXXIII
ANEXO 6: Descripción de las Clases de Análisis.	XXXVII

INTRODUCCIÓN

En el mundo se han desarrollado grandes avances sociales, políticos, tecnológicos, científicos y económicos, gracias al interés del hombre en desarrollarse y conocer a fondo el mundo en que vive. Una de las ramas en la que se ha dedicado a realizar grandes investigaciones es la Geología.

- **Geología** - Ciencia y estudio de la Tierra, su composición, estructura, propiedades físicas, historia y la variedad de procesos que la forman o la afectan. (Española, 2005)
- La **geología** (del griego *geo*, tierra, y *logos*, estudio) es la ciencia que estudia la forma interior del globo terrestre, la materia que lo compone, su mecanismo de formación, los cambios o alteraciones que éstas han experimentado desde su origen, y la colocación que tienen en su actual estado. (2006)
- En otras palabras, la geología es el campo de la ciencia que se interesa por el origen del planeta Tierra, su historia, su forma, la materia que lo configura y los procesos que actúan o han actuado sobre él. Es una de las muchas materias relacionadas como ciencias de la Tierra, o geociencias.

Con el auge y desarrollo de las nuevas tecnologías de la informática y las comunicaciones, se está viviendo un proceso de organización y estandarización de todo tipo de información, por lo que la aplicación de este proceso en los datos espaciales es tarea que acometen cada día más países, a tal punto que la Organización de las Naciones Unidas (ONU) crea una 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.

(Geominera, 2005)

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, entre otras. En este frente se destacan los EEUU, Inglaterra y casi todos los países desarrollados económicamente avanzando hacia otros países como Colombia.

Cuba desde que era colonia de España viene realizando descubrimientos e investigaciones mineras pero de baja calidad, ya que se carecía de personal calificado. Cuando pasa a ser neocolonia de los EE.UU tuvo algunos avances pero sólo en la rama del níquel y no es hasta el triunfo de la Revolución cuando se crea el Servicio Geológico Cubano en 1962, con lo cual se comenzó un intenso programa de investigaciones y levantamientos geológicos en cooperación con el entonces campo socialista, manteniéndose esta actividad hasta su desaparición. Luego en 1995 se crea la Oficina Nacional de Recursos Minerales (ONRM), autoridad minero-petrolera de la República de Cuba que forma parte del Ministerio de Industrias Básicas (MINBAS).

Cuba participa desde el año 2000 en eventos relacionados con la Infraestructura de Datos Espaciales y trabaja para crear su propia Infraestructura donde 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.

Se han realizado muchos intentos de confeccionar bases de datos digitales tales como:

- En los años 1972-1980 se crearon las Bases de datos (BD) con el objetivo de realizar los cálculos de reservas para materias primas níquelíferas en las Empresas René Ramos La tours de Nicaro y Pedro Soto Alba de Moa. El soporte de estas eran en tarjetas y cintas perforadas.
- Desde el año 1982 hasta 1992 se realizaron diferentes Bases de datos para la Industria níquelífera con el sistema Microniq desarrollado por la Empresa Geominera Oriente en computadoras personales.
- En 1988 se emprendió el proyecto para la creación del sistema de manipulación de Bases de Datos Geológicas (Sistema Ramal), financiado por la Unión Geólogo Minera, el cual no llegó a concluirse.
- Entre 1992 y 1997 se trabajó en la Empresa Geominera de Oriente en el desarrollo de un Sistema de Gestión de Bases de Datos Geológicos (GEODAT), el cual fue utilizado en la creación de numerosas BD de dicha empresa.

- A partir de 1991 se crean numerosas BD por las diferentes Empresas de la Unión Geominera, las cuales tienen diferentes formatos en dependencia de la Empresa y los datos que contenían, las que hoy suman casi un centenar que por su diversidad de formatos y contenidos se hacen muy difíciles sus usos.

Al no realizarse estos intentos 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 queda en ejecución trabajos iniciados en el año 2002.

A finales de ese año el Instituto de Geología y Paleontología (IGP) de la Unión Geominera del Ministerio del la Industria Básica (MINBAS) se propuso resolver la situación existente y uniformar y controlar la elaboración de las Bases de datos, por lo que surge el Programa GeoDato (ó Banco de Datos Geológicos), el cual pretende llevar a formato digital la información geológica seleccionada y la nueva que se genere, así como crear un grupo de bases de datos para guardar la información geológica primaria, pretérita y toda la futura. Este programa tiene como objetivo preservar para las futuras generaciones y la economía del país todo el valioso patrimonio de Informes Geológicos soportados en papel que corren el riesgo de perderse por su deterioro y la unificación de todas las bases de datos existentes con el fin de tener un formato único y su interrelación. Posteriormente a solicitud del IGP es creado como parte de las funciones del Banco de Datos Geológicos, GeoDato^{IC}, la cual es la aplicación que permite la creación y mantenimiento de las bases de datos geológicas.

El resultado tangible de esa conquista socialista que es el alto nivel del conocimiento geológico de Cuba para el beneficio de las generaciones presentes y futuras lo constituyen los documentos bien conservados en los archivos técnicos del País, en su mayoría bajo custodia ejemplar de la ONRM, estos fondos tienen por tanto un valor patrimonial indiscutible. El uso de esa información y su conservación ante el desarrollo actual de la informática demanda de su incorporación a soportes digitales. (ORIENTE)

Situación Problémica

Por consiguiente el Ministerio del la Industria Básica (MINBAS) específicamente la ONRM en unión con la Universidad de las Ciencias Informáticas implementan el Programa Nacional de Informatización del Conocimiento Geológico (PNICG), creando el Polo Productivo para su implementación, nombrado GeoInformática. El PNICG está compuesto de ocho módulos: Búsqueda Referativa, Concesionario,

Nomencladores, Balance, Metadatos, SIG WEB, Portal de la ONRM y GeoDato, este a su vez consta de cuatro aplicaciones: Centro de Datos¹, GeoDato^{IC} ², GeoDato^{KIT_ADMON} ³ y Sistema Gestión de Definiciones.

Las Definiciones son el conjunto de variables que se necesitan para la captación de los datos por los geólogos en el campo.

La aplicación para el registro de Definiciones geológicas, basado en Java y con una base de datos relacional desarrollada en PostgreSQL, es la encargada de registrar todas las Definiciones nuevas y almacenar las existentes, esta aplicación también es la encargada de validar la base de datos proveniente del campo para comparar la estructura de esta ya que es una base de datos fácilmente modificable ya que está confeccionada sobre Microsoft Access lo que no brinda ninguna seguridad sobre la integridad de los datos almacenados en ella. La aplicación GeoDato^{IC} basada en Delphi y que posee una base de datos confeccionada en Microsoft Access se encarga actualmente de visualizar y crear Rasgos y por último la aplicación para la comunicación con la PocketPC basada en Java debe ser la encargada de exportar las Definiciones existentes en la base de datos del GeoDato^{IC} con el objetivo de llevar al campo esta en una PocketPC, una vez que se recogen los datos de dichas Definiciones comienza el proceso de importación de las mismas con el objetivo futuro de realizar una validación de las mismas y el sistema encargado de esta validación es RBDARV como se había expuesto anteriormente. Este es a grandes rasgos el trabajo realizado por los tres módulos encargados del registro la visualización y la validación de Definiciones en la ONRM.

A continuación se muestra una figura que visualiza a de manera general lo expuesto anteriormente.

¹ Se encarga de realizar el plan de seguridad de un Centro de datos que posteriormente se construirá.

² La aplicación que permite la creación y mantenimiento de las bases de datos geológicas.

³ Aplicación de GeoDato encargada de la administración de datos.

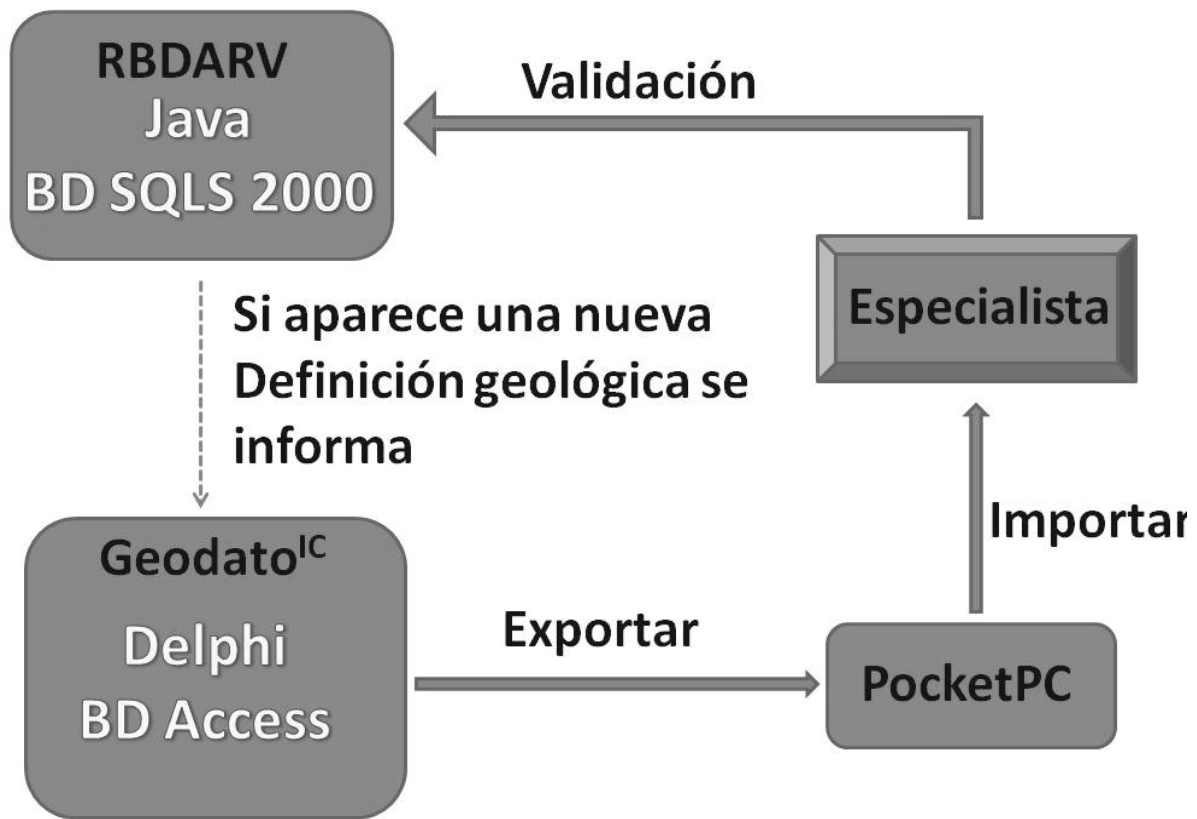


Figura 1: Esquema general del GeoDato.

La ONRM realizó una aplicación nombrada Sistema Gestión de Definiciones que no corre sobre la PocketPC, sino que se encarga de la gestión de las Definiciones, así como de exportarlas hacia la PocketPC e importarlas una vez terminado el trabajo en el campo, cuando el geólogo regrese con esta información. El Sistema gestión de Definiciones actualmente no se está utilizando debido a que sólo tiene implementada la funcionalidad de la creación de las Definiciones, pero no tiene las de exportación e importación, que serían las que darían solución a la problemática que dio lugar a su creación, además de que está desarrollada en software propietario, por lo que se debe migrar para software libre agregarle las funcionalidades que le falta y realizar una buena documentación de la misma.

Por tanto el **problema a resolver** es el siguiente:

¿Cómo perfeccionar el Sistema Gestión de Definiciones para facilitar el trabajo con la PocketPC?

Definiéndose como **objeto de estudio**:

Proceso de gestión de Definiciones.

Siendo el **campo de acción**:

Informatización del Sistema Gestión de Definiciones para facilitar el trabajo con la PocketPC en la ONRM.

Planteándose como **objetivo general**:

Realizar el análisis y el diseño para el Sistema Gestión de Definiciones sobre Software libre, creando nuevas funcionalidades que facilite el trabajo con la PocketPC.

Idea a defender:

Con el análisis y el diseño del Sistema Gestión de Definiciones que facilite el trabajo con la PocketPC basado en nuevas tecnologías, se contribuirá a perfeccionar los procesos de creación, importación y exportación de Definiciones.

Para ello se hace necesario plantear las siguientes tareas:

- Comprobación de la validez del problema a resolver.
- Estudio de la tecnología para desarrollar la aplicación de escritorio.
- Realización de un estudio minucioso de la aplicación existente en Delphi.
- Análisis y diseño del ciclo de vida del software.
- Documentación de la aplicación que facilitará el trabajo de la PocketPC.
- Estudio de la factibilidad, los costos y beneficios del software.

Como **métodos de investigación científica** a utilizar:

Métodos Teóricos:

- Histórico lógico: Posibilita el análisis histórico del proceso de gestión de información.

- Análisis y la síntesis: Se analiza la bibliografía y se realiza la síntesis de la misma.
- Modelación: Se modelan diagramas para la implementación de la aplicación.

Métodos Empíricos:

- Entrevistas: Se realizan entrevistas con el fin de precisar el problema a resolver, así como para la validación de la propuesta que se presenta.
- Análisis de documentos: Se basa en la revisión a documentos utilizados en la investigación.

Resultados esperados:

- El prototipo no funcional de una aplicación en el lenguaje de programación Java para la gestión de Definiciones para la PocketPC.

Estructura de la investigación:

La investigación consta de cinco capítulos. En el capítulo 1 se aborda la fundamentación teórica de la investigación, siendo en el capítulo 2 donde se va a tratar las tendencias y tecnologías actuales para desarrollar la aplicación. El capítulo 3 comprende las características del sistema, realizándose la presentación de la solución propuesta. En el capítulo 4 es donde se efectúa el diseño de dicha solución. Y en el capítulo 5 se maneja el estudio de la factibilidad, la planificación, los costos y los beneficios.

CAPÍTULO 1

Fundamentación Teórica

Introducción:

En este capítulo se aborda la fundamentación teórica del sistema informático existente en el Oficina Nacional de Recursos Minerales (ONRM) para gestionar las Definiciones y la aplicación a construir tiene como meta perfeccionar la gestión de información dentro de la ONRM, teniendo en cuenta el estado actual a nivel internacional y en Cuba, a través de la definición de conceptos, explicando el proceso en si y planteando la situación problemática existente.

1.1 Conceptos que ayudan a entender el problema

1.1.1 Definición

Definición: Es una entidad que está compuesta por uno o varios Rasgos y Perfiles de trabajos con sus respectivas variables.

1.1.2 Rasgos

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, 2005)

1.1.3 Tipos de Rasgos

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, 2005)

1.1.4 Perfiles de Trabajo

Perfiles de Trabajo: Es la representación digital dentro de **GeoDato^{lc}**, de un conjunto de datos en un documento geológico. Contiene 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 ello se hace posible crear 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, 2005)

1.1.5 Variables Virtuales

Variables Virtuales: Son los atributos normalizados de las clases persistentes que definen las entidades diseñadas en los modelos lógicos. (Geominera, 2005)

1.1.6 Variables Elementales

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 en estas. (Geominera, 2005)

1.1.7 Puesta de Trabajo

Puesta de Trabajo: Se denomina al grupo de variables elementales con valor, definidas para caracterizar a la variable virtual y es identificada con una etiqueta. (Geominera, 2005)

1.2 Proceso de gestión de las Definiciones para una PocketPC

1.2.1 Problema y situación problemática

La ONRM ha decidido informatizar todo lo referente al flujo de información dentro de sus instituciones, uno de los avances que introdujeron es utilizar las PocketPC para la captación de datos en el campo por los geólogos, pero esto trajo consigo la necesidad de crear entre otras cosas la aplicación Sistema Gestión de Definiciones que facilita el trabajo con la PocketPC, porque mediante la misma se seleccionan las variables tanto de los Rasgos como de los Perfiles de trabajo que desean exportar a la PocketPC según en el terreno que se va a trabajar.

El Sistema Gestión de Definiciones no cumple con algunas de las funcionalidades que se quieren para llevar a cabo esta operación, se encuentra desarrollada en Delphi y no está documentada, lo que implica que es casi imposible darle soporte técnico de actualizaciones y adecuaciones a la misma; además de que por ser software propietario se crea una dependencia hacia las licencias y es política del Estado Cubano migrar a Software libre, en este caso Java.

1.2.2 Análisis de los resultados de las técnicas aplicadas

En la entrevista realizada a un especialista del MINBAS, se corroboró la necesidad de implementar un nuevo sistema informático que cumpla con todas las funcionalidades que garantice la gestión de las Definiciones basada en software libre, que cumpla con los estándares de la Ingeniería de software, por lo que se facilite el trabajo en el momento de darle soporte técnico de actualizaciones y adecuaciones a la misma. Ver ANEXO 1.

1.2.3 Información sobre las PocketPC

Las PocketPC es un ordenador de bolsillo, también llamada PDA (Personal Digital Assistant). Se trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable, que ejecuta el sistema operativo Windows CE de Microsoft entre otros, el cual le proporciona capacidades similares a las computadoras de escritorio.

Microsoft sacó la línea al mercado en 1998, decidiendo denominarla Palm PC. Debido a una demanda de Palm, el nombre fue cambiado a PocketPC.

1.2.4 Existencia de soluciones semejantes a esta investigación

La única solución semejante a esta investigación es con la que cuentan en la ONRM y a continuación se trata más específicamente sus características y deficiencias.

La aplicación más semejante es la mencionada anteriormente, pero cualquier aplicación de escritorio que tenga funcionalidades como las que aquí se plantean puede ser considerada semejante a esta.

1.2.4.1 Concepción General del Sistema

Es una aplicación de escritorio que permite de una forma sencilla, crea las Definiciones para la PocketPC.

1.2.4.2 Características

- Desarrollada en Delphi.
- Trabaja sobre una Base de Datos de Access.
- Posee una interfaz apropiada.
- No es muy sencilla de usar.
- Sus funcionalidades están divididas según las Definiciones de la BDARV.

1.2.4.3 Deficiencias

Esta aplicación no está siendo usada en la ONRM porque posee ciertas deficiencias que hacen indispensables su sustitución. En primer lugar no tiene todas las funcionalidades que responden a la necesidad de su creación desarrolladas, además de no poseer ningún tipo de documentación, esto hace que su soporte, mantenimiento y actualizaciones sea prácticamente imposible. Está completamente sobre una plataforma propietaria, por lo que acarrea todos los problemas que esto supone, dígase de requerimientos de hardware como de pago de licencias.

1.3 Conclusiones parciales

- El sistema informático Sistema Gestión de Definiciones existente en la ONRM, para facilitar el trabajo con la PocketPC, no es óptimo, debido a la falta de documentación, no cumplir con

todas las funcionalidades necesarias, además la plataforma en la que fue desarrollada no es la más adecuada.

- El estudio teórico de la gestión de Definiciones, facilitó el aprendizaje sobre la información necesaria e imprescindible, que serán necesarios para diseñar el sistema.

2

CAPÍTULO 2

Tendencias y tecnologías a utilizar

Introducción:

Este capítulo aborda los temas convenientes para cumplir con el objetivo planteado, que es definir y justificar las tecnologías a usar durante el desarrollo de la propuesta de solución, teniendo en cuenta que cumplan con las expectativas planteadas como desarrollar el software totalmente sobre plataforma libre, sin perder de vista las particularidades del producto a realizar.

2.1 Lenguaje Unificado de Modelado (UML)

La falta de estandarización en la manera de representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores. (Hernandez Orallo, Enrique;, 2007)

Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño. (Hernandez Orallo, Enrique;, 2007)

UML es además un método formal de modelado.

El modelado del problema es un paso muy importante no sólo para grandes sistemas sino también para algunos pequeños aunque es obvio que mientras mayor sea el sistema más necesario se hace el apoyarse en un formato, entre otras razones porque es la manera que se tiene de garantizar un mayor

entendimiento sobre todo cuando se es capaz de modelar y describir todos los procesos que se precisa realizar.

Alguna de las características que presenta el UML son: (Larman, 2004)

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

UML es independiente del proceso, aunque para utilizarlo óptimamente se debería emplear en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

2.2 Metodología de desarrollo de Software

Como resultado de los avances con los que hoy se cuenta en el campo de la informática, se hace cada día más difícil seleccionar con precisión la metodología más adecuada que posibilite la obtención de los mejores resultados. El objetivo de un proceso de desarrollo es subir la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Independientemente del destino final del SW, hay que producir lo esperado en el tiempo esperado y con el coste esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo, de allí la importancia que amerita hacer una selección acertada.

2.2.1 Proceso Unificado de Desarrollo (RUP)

Un proceso efectivo proporciona normas para el desarrollo eficiente de software (sw) de calidad, es por esto que se emplea en empresas en las cuales el desarrollo de sw es esencial y en otras, así como cuenta con muchísimas ventajas también tiene como todo sus fallas en algunos casos. El proceso unificado del sw está basado en componentes, utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema sw, pero RUP se resume en tres frases claves que hace que sea único: (Pressman, 2002)

- Dirigido por casos de uso
- Basado en la arquitectura
- Iterativo e incremental.

RUP consta de cuatro etapas o fase por donde el software tiene que transitar para que obtenga la calidad requerida, además de está dividido en nueve flujos de trabajo. (Véase Figura 1) (Pressman, 2002)

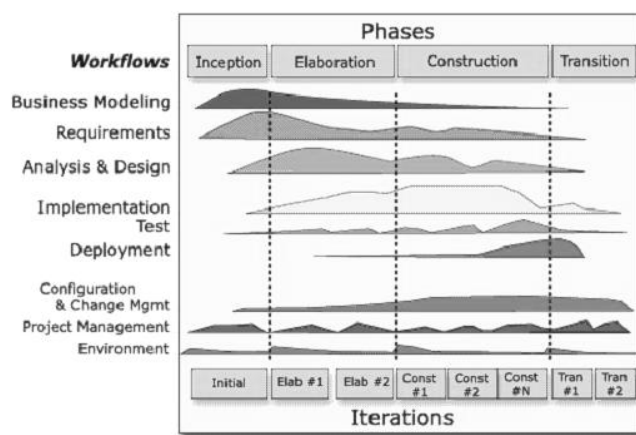


Figura 2: Vista general de RUP.

El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas SW, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferente tamaño de proyectos. (Pressman, 2002)

2.2.2 Programación Extrema (XP)

Una metodología que como todas consta de ventajas y desventajas es la Programación Extrema (XP). De esta señalar varias cosas entre ellas que mientras que el RUP intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. (Molpeceres, 2002)

También es importante resaltar que XP intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este representante debería estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones, de ahí lo de competente. (Molpeceres, 2002)

Este es un paso de suma importancia ya que es el resultado de cada iteración un programa que se transmite al cliente para que lo juzgue. En base a su opinión se definen las siguientes iteraciones del proyecto y si el cliente no está contento se adaptará el plan de releases e iteraciones hasta que el cliente de su aprobación y el software esté a su gusto. (Molpeceres, 2002)

Al contrario que muchos otros métodos, el código pertenece al equipo en completo, no a un programador o pareja, de forma que cada programador puede cambiar cualquier parte del código en cualquier momento si así lo necesita, dejándose en todo caso las mejoras orientadas al rendimiento para el final. (Molpeceres, 2002)

Se sigue un diseño evolutivo con la siguiente premisa: conseguir la funcionalidad deseada de la forma más sencilla posible. (Molpeceres, 2002)

Estas son a grandes rasgos las principales características de XP, son estas las cosas que marcan la diferencia cuando se hace referencia a esta metodología.

2.2.3 Desarrollo Guiado por Funcionalidades (FDD)

FDD se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero es más similar a este último. Está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar. (Molpeceres, 2002)

Un proyecto que sigue FDD se divide en 5 fases: (Molpeceres, 2002)

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.
- Plan de releases en base a las funcionalidades a implementar.

- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.

Las primeras tres fases ocupan gran parte del tiempo en las primeras iteraciones, siendo las dos últimas las que absorben la mayor parte del tiempo según va avanzando el proyecto, limitándose las primeras a un proceso de refinamiento. (Molpeceres, 2002)

2.2.4 Selección de la Metodología a utilizar

Luego de hacer un estudio minucioso de cada una de las metodologías anteriormente planteadas se tomó la decisión de usar RUP, ya que de acuerdo a sus características y ventajas es el que más cumple con los requisitos necesarios para desarrollar un producto con calidad , además las otras dos metodologías estudiadas presentan algunas fallas que pudieran hacer peligrar los objetivos fundamentales del producto y para concluir también señalar que es RUP la más abarcadora de las tres ya que las otras dos se consideran particularidades de esta.

2.3 Software Libre y su utilización en Cuba

En Cuba se ha observado la tendencia a utilizar software libre en varias empresas y dentro de ellas se encuentra la Oficina Nacional de Recursos Minerales (ONRM). Ha aumentado considerablemente la necesidad de migrar del software propietario donde sus creadores se reservan el derecho de vender sus productos de software, al llamado software libre donde los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software de la GNU: (Silva, 2006)

Libertad 1: Usar el programa, con cualquier propósito.

Libertad 2: Estudiar cómo funciona el programa, y adaptarlo a tus necesidades.

Libertad 3: Distribuir copias, con lo que puedes ayudar a los demás.

Libertad 4: Mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Una vez que un programa se orienta bajo licencia GNU GPL15, la libertad es una parte inseparable de él. Según el proyecto GNU dentro de él se encuentra una serie de software que son de prioridad libre, dentro de estos existe un sistema operativo muy estable en su rendimiento que en los últimos

años ha alcanzado una gran demanda y un gran auge de utilización por la comunidad informática; llamado GNU/Linux o Sistema Operativo Linux. Por petición del cliente se hará uso de este sistema operativo, tratando mayormente de hacer uso de herramientas y tecnologías libres. (Silva, 2006)

2.3.1 Linux

Linux es un sistema operativo tipo Unix (también conocido como GNU/Linux) que se distribuye bajo la Licencia Pública General de GNU o GPL, el código fuente tiene que estar siempre accesible y cualquier modificación ó trabajo derivado tiene que tener esta licencia. Su nombre proviene del Núcleo de Linux, desarrollado en 1991 por Linus Torvalds. Es usado ampliamente en servidores y súper-computadores. (Martinez, 1994 - 2008)

Características más importantes de GNU/LINUX: (Martinez, 1994 - 2008)

- **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** Muchos usuarios usando la misma maquina al mismo tiempo.
- **Multipataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como amd64, Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** Soporte para sistemas con más de un procesador está disponible para Intel, AMD y SPARC.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Tiene capacidad para compilar C, C++, Java, Ada, entre otros muchos lenguajes como Python, Perl, PHP y Ruby.
- Tiene varios IDEs disponibles incluyendo, Anjuta, KDevelop, Ultimate++, Code: Blocks, NetBeans IDE y Eclipse.

2.3.1.1 Distribución Ubuntu

Ubuntu es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a computadoras de escritorio aunque también proporciona soporte para servidores. (Libre, 1999 - 2008)

Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth. (Libre, 1999 - 2008)

Entre sus principales características se encuentran:

- Basada en la distribución Debian.
- Disponible en 4 arquitecturas: Intel x86, AMD64, PowerPC, SPARC (para esta última sólo existe la versión servidor).
- Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de las comunidades de Debian y GNOME.
- Las versiones estables se liberan cada 6 meses y se mantienen actualizadas en materia de seguridad hasta 18 meses después de su lanzamiento.
- El entorno de escritorio oficial es Gnome y se sincronizan con sus liberaciones.
- El navegador web oficial es Mozilla Firefox.
- El sistema incluye funciones avanzadas de seguridad y entre sus políticas se encuentra el no activar, de forma predeterminada, procesos latentes al momento de instalarse. Por eso mismo, no hay un firewall predeterminado, ya que no existen servicios que puedan atentar a la seguridad del sistema.
- Para labores administrativas en terminal incluye una herramienta llamada sudo (similar al Mac OS X), con la que se evita el uso del usuario root (administrador).
- Mejora la accesibilidad y la internacionalización, de modo que el software está disponible para tanta gente como sea posible. En la versión 5.04, el UTF-8 es la codificación de caracteres en forma predeterminada.

- No sólo se relaciona con Debian por el uso del mismo formato de paquetes también tiene uniones muy fuertes con esa comunidad, contribuyendo con cualquier cambio directa e inmediatamente, y no sólo anunciándolos. Esto sucede en los tiempos de lanzamiento. Muchos de los desarrolladores de Ubuntu son también responsables de los paquetes importantes dentro de la distribución de Debian.
- Todos los lanzamientos de Ubuntu se proporcionan sin costo alguno. Los CDs de la distribución se envían de forma gratuita a cualquier persona que los solicite mediante el servicio Shipt (la versión 6.10 no se llegó a distribuir de forma gratuita en CD, pero la versión 7.04 sí). También es posible descargar las imágenes ISO de los discos por transferencia directa o bajo la tecnología Bittorrent.
- Ubuntu no cobra honorarios por la suscripción de mejoras de la "Edición Enterprise".

2.4 Arquitectura por Capas

La **programación por capas** es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario. (Zamora, 2008)

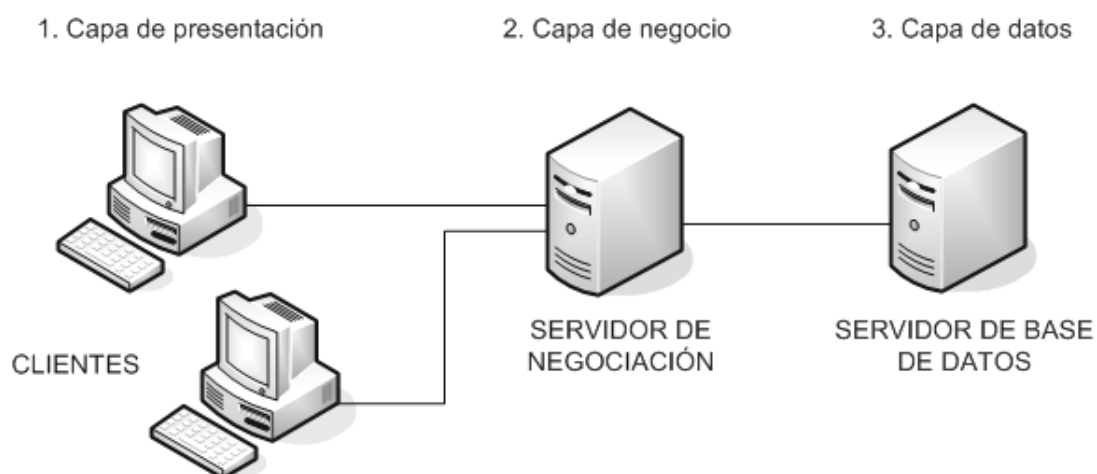


Figura 3: Arquitectura por Capas.

La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. (Zamora, 2008)

Capas o niveles (Zamora, 2008)

1.- Capa de presentación: es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

2.- Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

3.- Capa de datos: es donde residen los datos y es la encargada de acceder a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.5 Aplicación de escritorio como vía alternativa

Hace tiempo, coincidiendo con el inicio del auge de las tecnologías Web, apareció un nuevo modelo de aplicaciones en las cuales el interfaz de usuario se presentaba dentro de páginas Web generadas dentro de algún tipo de servidor Web o de aplicaciones. Este tipo de clientes pasó a llamarse clientes ligeros, haciendo referencia este adjetivo a algunas de las ventajas de este tipo de aplicaciones que hacían muy sencilla su ejecución y despliegue, un ejemplo: no necesitar ningún tipo de instalación en el cliente ya que se ejecutaban en el navegador Web del usuario, no consumir enormes cantidades de espacio en disco, el no incluir apenas lógica de procesamiento lo que hacía que aplicaciones que previamente necesitaban ingentes cantidades de memoria y de CPU se pudiesen ejecutar ahora en máquinas más precarias, entre otras. Evidentemente, si existían clientes ligeros, deberían existir también clientes pesados, y ese fue el adjetivo que le dedicaron los partidarios de este tipo de aplicaciones, a las obsoletas tradicionales aplicaciones de escritorio, cosa que evidentemente no gustó demasiado a la gente que se ganaba la vida desarrollándolas.

Con el paso del tiempo, y por consiguiente con la adopción y el uso de los autoproclamados clientes ligeros, comenzaron a hacerse visibles algunas de sus deficiencias. Estas deficiencias, siempre existieron, como en todas las tecnologías, a pesar de los esfuerzos de los departamentos de marketing por vender servidores de aplicaciones. No disponer de un conjunto tan grande de componentes de interfaz de usuario, no disponer de unas tecnologías de procesamiento realmente potentes, teniendo que limitarse a lenguajes simples de scripting, la problemática de la recepción de datos (pull), o la necesidad de incurrir en continuos accesos al servidor para obtener información, se presentaron como problemas importantes. En ese momento, reaparecen de nuevo los defensores de las aplicaciones de escritorio, que pasan a autoproclamarse clientes ricos (en la actualidad este término ya se ha hecho mucho más genérico, y se tiene que mucha gente considera clientes Flash o DHTML como clientes ricos), quedando relegadas las aplicaciones Web al rango implícito de clientes pobres.

Las aplicaciones de escritorio a pesar de que muchos no las toman en cuenta tienen importantísimas ventajas las cuales, unidas a las necesidades de los clientes han sido fundamentales para que se optase por esta como la vía de solución más viable.

Las principales **ventajas** de una aplicación de escritorio son: (Doral, 2005)

- Navegación e interfaz de usuario **más rápida** que en una **aplicación web**.
- Normalmente **no** es necesaria una conexión a internet ya que funciona localmente o en la intranet de la empresa.
- Es mejor opción cuando es necesario realizar numerosos cálculos o procesos de CPU muy intensivos. En estos casos se utiliza el **lenguaje C++** para generar una aplicación compilada a código nativo dando lugar a un programa que posee un extraordinario rendimiento.
- Fácil acceso a **recursos** locales: disco duro, impresora, memoria de vídeo, sonido, ratón, teclado, entre otros.
- Fácil acceso a **aplicaciones** locales en caso de que sea necesario compartir datos entre aplicaciones.

Las principales **desventajas**: (Doral, 2005)

- Normalmente hace falta utilizar **herramientas de pago**: Windows Server, Microsoft Access, SQL Server, etc. Además este coste depende del número de licencias/ordenadores, lo cuál limita su uso y obliga a estar pendiente para extender el número de licencias. A favor hay que decir que estas herramientas ofrecen un soporte técnico.
- La aplicación **tiene instalación**, es decir, es necesario llevar a cabo un proceso de instalación ordenador por ordenador.
- Idem para **actualizar** la aplicación.
- Cuando sale una nueva versión de Windows puede que haya que actualizar también la aplicación.

2.6 Fundamentación del Lenguaje que se va a utilizar (Java)

El objetivo principal de Java es conseguir un entorno de desarrollo de software que sea independiente de la plataforma de ejecución. Necesita 128 MB de memoria por sesión de usuario. (Villalobos, 2006)

Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje. (Villalobos, 2006)

Java elimina muchas de las características de otros lenguajes como C++, para mantener reducidas las especificaciones del lenguaje y añadir características muy útiles como el garbage collector (reciclador de memoria dinámica). No es necesario preocuparse de liberar memoria, el reciclador se encarga de ello y como es un thread de baja prioridad, cuando entra en acción, permite liberar bloques de memoria muy grandes, lo que reduce la fragmentación de la memoria. (Villalobos, 2006)

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan: (Villalobos, 2006)

- Aritmética de punteros.

- No existen referencias.
- Registros (struct)
- Definición de tipos (typedef)
- Macros (#define)
- Necesidad de liberar memoria (free)

Aunque, en realidad, lo que hace es eliminar las palabras reservadas (struct, typedef), ya que las clases son algo parecido.

Además, el intérprete completo de Java que hay en este momento es muy pequeño, solamente ocupa 215 Kb de RAM.

2.6.1 Características del Java 5

El principal objetivo de esta versión JDK 5 es la facilidad de desarrollo con las nuevas características como tipos de datos genéricos, el loop for-each, autoboxing/unboxing, enums, varargs, static imports y anotaciones. (Villalobos, 2006)

El JDK 5 está diseñado para hacer que los programas se estructuren de manera más ordenada, más corta y fortificando la seguridad. Además, proporciona un soporte lingüístico para idiomas comunes. (Villalobos, 2006)

2.7 Herramientas a utilizar

Luego de hacer un estudio minucioso de algunas herramientas se concluyó que las herramientas más adecuadas por sus características y por cumplir con los requisitos necesarios para obtener una mayor calidad del software son las siguientes.

2.7.1 Visual Paradigm

Visual Paradigm 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. (Manager)

Algunos de los lenguajes que soporta son: (Manager)

- Java.
- C++.
- CORBA.
- IDL.
- PHP.
- esquema de XML.

Son muchas las características que hacen de Visual Paradigm una herramienta en la que se debe pensar si de perfección y calidad se trata, algunas de ellas son las siguientes: (Manager)

- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Modelado colaborativo con CVS y Subversión (nueva característica).
- Interoperabilidad con modelos UML2 (meta modelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta.
- Ingeniería inversa - Código a modelo, código a diagrama.

2.7.2 NetBeans

El NetBeans IDE es un entorno de desarrollo .Una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (2007)

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. (2007)

2.8 Conclusiones Parciales

A modo de conclusión de este capítulo, decir que se ha logrado realizar una acertada selección de las tecnologías más adecuadas para el desarrollo de la propuesta de solución, por lo que se espera que de este modo se le de solución de la mejor manera posible, a los problemas planteados.

- El lenguaje de programación Java es el que cumple con las necesidades planteadas.
- La metodología RUP representa la guía más óptima para el desarrollo de la aplicación propuesta.
- El sistema operativo Linux es el más adecuado según los objetivos.

CAPÍTULO 3**Descripción de la solución propuesta****Introducción:**

En el presente capítulo se tiene como objetivo fundamental describir la propuesta de solución para ello se describen una serie de datos que proporcionan una mejor comprensión del mismo. Primeramente se modela el negocio propuesto, se identifican los actores, trabajadores, los Casos de Usos en cuestión y las descripciones textuales de los mismos, además se enumeran los requerimientos funcionales y no funcionales con los que debe cumplir el sistema a desarrollar y se describe la solución propuesta en términos de casos de uso de sistema. Esperándose que de esta manera se logre tener una mejor concepción, a grandes rasgos, de lo que es el sistema.

3.1 Análisis del problema actual

En la actualidad para el proceso de captación de minerales en el campo, los geólogos de la Oficina Nacional de Recursos Minerales (ONRM) utilizan como herramienta fundamental el PocketPC pero debido a la poca capacidad de memoria con la que cuenta esta, se hace difícil almacenar el total de Definiciones referente a lo que se va a utilizar en el terreno. Por lo que se dificulta la captura de información geológica en un área determinada así como su posterior almacenamiento o hacer diferentes operaciones de gestión de esta información. Por lo que se recurre a crear entre otras cosas el diseño de una aplicación que se encargue de la gestión de las Definiciones, esto consiste en ver cuales son las variables tanto de los Rasgos como de los Perfiles de trabajo que se desean exportar a la PocketPC según en el terreno que se va a trabajar y crear la definición con las mismas, exportarlas y posteriormente importarlas, debido a que en la ONRM cuentan con una aplicación que cumple con una de las funcionalidades de las que debía cumplir, además presenta grandes problemas pues está

programada en Delphi y no está documentada lo que implica que es casi imposible darle soporte técnico de actualizaciones y adecuaciones a la misma.

3.2 Modelo del Negocio propuesto

Se realiza el modelado porque es la mejor forma de describir el negocio en términos de caso de uso además permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades. Lo cual es considerado un punto fundamental para así comprender los problemas actuales de la organización identificando las mejoras potenciales. Logrando de esta manera que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de dicha organización.

Para una correcta descripción de los procesos del negocio, basado en las técnicas de modelado que propone UML se procede a realizar la identificación de los procesos del negocio de la solución propuesta.

3.2.1 Reglas generales del Negocio

1. Sólo el Administrador tiene acceso a:
 - Realizar cualquier operación en el Sistema Gestión de Definiciones.
2. Sólo será fijada una definición nueva en caso de que haya sido solicitada para su posterior exportación.
3. Toda variable debe contar con un valor por defecto.
4. Una vez fijada una definición no estará permitido realizar a las mismas transformaciones de ningún tipo.
5. Debe contarse con un PocketPC.
6. Tiene que existir las Definiciones de rasgos, tipos de rasgos y perfiles de trabajo en GeoDato^{IC}.

3.2.2 Actores del Negocio

Los actores del negocio representan a personas o sistemas (Software) dentro de este que son las que se benefician de las actividades que están comprendidas dentro de un caso de uso y obtienen un resultado conspicuo de gran valor de los proceso del negocio.

Definición del actor estudiado:

Actor	Descripción
Geólogo	Representa al geólogo que debe capturar la información geológica en el campo, principal benefactor de la solicitud de la planilla de los datos geológicos y del archivo de las planillas.

Tabla 1: Descripción del Actor del Negocio.

3.2.3 Trabajador del Negocio

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

Definición del trabajador estudiado:

Trabajador	Descripción
Administrador	Representa al geólogo que debe entregar y archivar las planillas de los datos geológicos.

Tabla 2: Descripción del Trabajador del Negocio.

3.2.4 Modelos de Casos de Uso del Negocio

Se usa un modelo de casos de uso para detallar de manera visual el proceso de creación de las Definiciones para la PocketPC.



Figura 4: Modelo de Caso de Uso del Negocio.

3.3 Descripción de los procesos del negocio propuesto

Caso de Uso:	Solicitar planilla de datos geológicos	
Actores:	Geólogo.	
Resumen:	El Caso de Uso se inicia cuando el geólogo que tiene que tomar los datos geológicos en el campo pide que se le entregue la planilla que indica los datos que debe captar y el mismo finaliza cuando el administrador le entrega la planilla.	
Precondiciones:	Tienen que existir las planillas de rasgos, tipos de rasgos y perfiles de trabajo en GeoDato ^{IC} y que se cuente con un PocketPC.	
Prioridad		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1) El geólogo solicita al administrador la entrega de la planilla de datos geológicos.	1.1) El administrador pide al geólogo que le comunique que información va a captar.	
2) El geólogo le notifica la información que va a captar.	2.1) El administrador busca la planilla referente a esta información. 2.2) Encuentra la planilla y se la entrega.	
3) El geólogo recibe la planilla.		
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
	2.2) No encuentra la planilla.	
3) El geólogo no recibe la planilla.		
Mejoras		
Poscondiciones		

Tabla 3: Descripción del CUN<Crear Definiciones para la PocketPC>.

Caso de Uso:	Archivar las planillas de datos geológicos captados	
Actores:	Geólogo.	
Resumen:	El Caso de Uso se inicia cuando el geólogo que tiene que tomar los datos geológicos en el campo viene a entregar la planilla con los datos captados captar y el mismo finaliza cuando el administrador le recoge la planilla.	
Precondiciones:		
Prioridad		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1) El geólogo entrega la planilla con los datos captados.	1.1) El administrador toma la planilla y la archiva.	
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
Mejoras		
Poscondiciones		

Tabla 4: Descripción del CUN<Archivar las planillas de datos geológicos captados>.

3.4 Modelo de objetos del Negocio

A través del modelado de los objetos del negocio, se identifican las entidades y los trabajadores que participan en las actividades que se realizan en el mismo.



Figura 5: Modelo de Objetos del Negocio.

3.5 Requisitos del Sistema

Los requisitos son una guía factible para lograr que el sistema cumpla con los objetivos planteados al inicio, sobre todo cuando se tiene en cuenta que los requisitos definen que es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. Ya conocido los conceptos asociados al objeto de estudio del problema se procede a modelar el sistema, el cual debe cumplir con los siguientes requisitos.

3.5.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, por tal motivo a continuación se presenta un conjunto de ellos:

1. Crear Definiciones.
2. Visualizar datos
 - 2.1 Visualizar las Definiciones existentes
 - 2.2 Visualizar los tipos de Rango
 - 2.3 Visualizar los perfiles de trabajo
 - 2.4 Visualizar las variables virtuales
 - 2.5 Visualizar las variables elementales
 - 2.6 Visualizar la Definición solicitada.
3. Modificar datos.

4. Exportar Definiciones creadas.
5. Importar Definiciones captadas.

3.5.2 Requerimientos no funcionales

Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Usabilidad

Los usuarios del sistema pueden hacer uso del mismo sin complicaciones porque ha sido diseñado para un manejo eficaz. Sólo podrá tener acceso a este producto el administrador.

Soporte

Se requiere un servidor de bases de datos con las siguientes características:

Soporte para grandes volúmenes de datos y velocidad de procesamiento.

Tiempo de respuesta rápido.

Restricciones de diseño

Para dar mantenimiento al mismo debe buscarse un profesional que conozca la tecnología. Teniendo en cuenta los ambientes de desarrollo del software que se muestran a continuación:

Lenguaje de Programación Java (J2SE)

Framework Para Java: J2EE (java).

IDE de desarrollo (Entorno de Desarrollo Integrado): NetBeans (Java).

Interfaz

La aplicación debe contar con un interfaz amigable para hacer menos engorroso el trabajo del usuario.

3.6 Propuesta de Solución

Como propuesta de solución al problema planteado se ofrece el diseño de un prototipo no funcional de una aplicación de Escritorio. El cual estará a la disposición de la ONRM que se encargará de realizar cualquier señalamiento referente a algún cambio que desee que se realice antes de su posterior implementación. Para que se logre una solución a la altura de lo que se necesita se procede a realizar un Diagrama de Casos de Uso teniendo en cuenta los requisitos funcionales del sistema así

como los actores y los casos de uso que posibilitarán mediante las facilidades que brinda UML hacer una correcta representación del sistema propuesto.

Actor	Descripción
Administrador	Es el encargado de realizar todas las tareas correspondientes para la creación de las Definiciones solicitadas por el geólogo para la PocketPC así como de exportarlas e importarlas y cualquier otra acción vinculada a este proceso.

Tabla 5: Descripción de los Actores del Sistema.

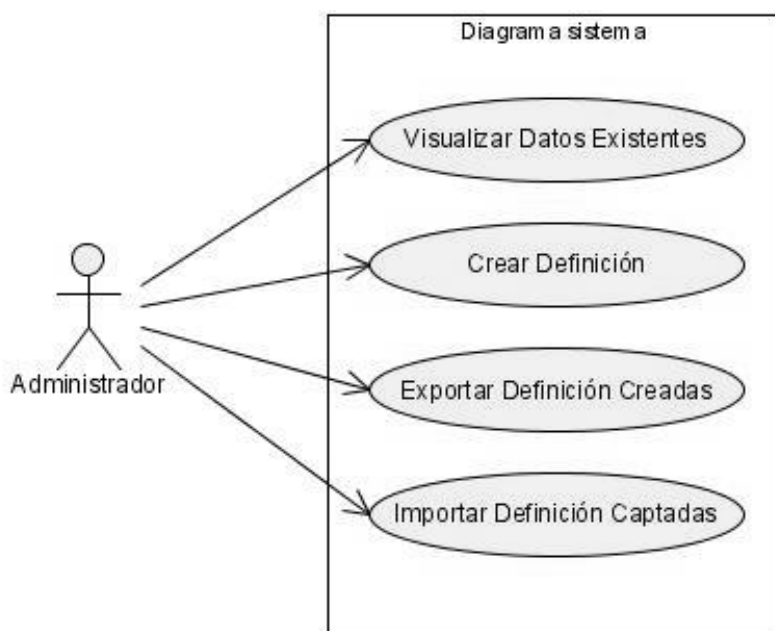


Figura 6: Diagrama de Casos de Usos del Sistema.

Descripción de los Casos de Uso del Sistema.

Caso de Uso:	Crear Definición.
Actores:	Administrador.
Resumen:	El caso de uso inicia cuando se hace la solicitud de una definición para la PocketPC se muestran las variables existentes en la base de datos y partiendo de aquí se seleccionan las que sean necesarias y finalmente se fija.
Precondiciones:	El administrador debe estar previamente autenticado y contar con los permisos de administración.

	Deben existir en la BD las Definiciones de Rango, Tipos de rango y Perfiles de trabajo.
Referencias	R1
Prioridad	Critico.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. El Sistema muestra las variables existentes y controles que permiten la selección de las opciones Rasgos, Tipo de Rasgos y Perfiles de trabajos con sus respectivas variables virtuales.
<ol style="list-style-type: none"> 2. El administrador selecciona las variables necesarias para crear la definición, poniendo "S" en el parámetro Export, así como puede cambiar el nombre de la Etiqueta de salida, debe darle un valor por defecto a todas las variables que no se exporten, así como seleccionar el valor de las puestas de trabajo y las coordenadas. Para seleccionar las variables de los Rasgos y Tipos de rasgos da clic en el tabsheet "Rasgos" y para seleccionar la de los perfiles de trabajo da clic en el tabsheet "Perfil de Trabajo". 	
<ol style="list-style-type: none"> 3. El administrador presiona el tabsheet "Fijar Definición". 	<ol style="list-style-type: none"> 4. El sistema visualiza los datos previamente seleccionados por el administrador para que este verifique si son correctos.
<ol style="list-style-type: none"> 5. El administrador presiona el botón "Fijar". 	<ol style="list-style-type: none"> 6. El sistema fija la definición y muestra un mensaje informando que la definición ha sido fijada satisfactoriamente.
Prototipo de Interfaz	

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El Administrador presiona el botón Cancelar con el propósito de anular la selección realizada.	4. El sistema cancela la operación. Y regresa a la interfaz principal.
5. El administrador selecciona el botón cancelar con el objetivo de cambiar la selección realizada.	6. El sistema regresa a la interfaz anterior.
7. Realiza los cambios deseados y presiona el botón "Fijar Definición".	
<i>Prototipo de Interfaz</i>	
Poscondiciones	

Tabla 6: Caso de Uso <Crear Definición>.

Caso de Uso:	Exportar Definición Creada	
Actores:	Administrador.	
Resumen:	El caso de uso inicia cuando el administrador decide exportar una definición y termina cuando esta es exportada satisfactoriamente a la PocketPC.	
Precondiciones:	Que la PocketPC esté conectada a la computadora.	
Referencias	R4	
Prioridad	Crítico	
Flujo Normal de Eventos		
	Acción del Actor	Respuesta del Negocio
	1. El administrador selecciona la opción exportar haciendo clic en el botón con el mismo nombre.	2. El sistema muestra las Definiciones creadas.
	3. El administrador selecciona la definición que desea exportar.	4. El sistema ofrece la opción de guardar la definición en forma de fichero.
	5. El administrador guarda el fichero en la PocketPC.	
Prototipo de Interfaz		
Flujos Alternos		
	Acción del Actor	Respuesta del Negocio
	2. El administrador no selecciona la definición.	3. El sistema muestra un mensaje informando que debe seleccionar la definición deseada y regresa a la interfaz anterior.
Poscondiciones		

Tabla 7: Caso de Uso <Exportar Definición Creada>.

Caso de Uso:	Importar Definición Captada	
Actores:	Administrador.	
Resumen:	El caso de uso inicia cuando el administrador decide importar las Definiciones desde la PocketPC, y termina cuando esta definición ha sido guardada en la BD.	
Precondiciones:	Que la PocketPC esté conectada a la computadora.	
Referencias	R5, R3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. El administrador selecciona la opción de importar haciendo clic en el botón del mismo nombre.		2. El sistema muestra el directorio para localizar el fichero en la PocketPC
3. El usuario busca el fichero.		
4. Selecciona fichero a cargar.		5. El sistema carga fichero y lo guarda los datos.
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor		Respuesta del Negocio
Prototipo de Interfaz		
Poscondiciones		

Tabla 8: Caso de Uso <Importar Definición Captada>.

Caso de Uso:	Visualizar Datos Existentes	
Actores:	Administrador	
Resumen:	El caso de uso inicia cuando el administrador escoge la opción visualizar datos y finaliza cuando el sistema le muestra los datos de la Base de Datos.	
Precondiciones:	Deben existir en la BD las Definiciones de Rango, Tipos de rango y Perfiles de trabajo.	
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 Negocio	
1. El administrador selecciona la opción visualizar datos en el menú.	2. El sistema le muestra los datos de la BD.	
<i>Prototipo de Interfaz</i>		
Flujos Alternos		
Acción del Actor	Respuesta del Negocio	
<i>Prototipo de Interfaz</i>		
Poscondiciones		

Tabla 9: Caso de Uso <Visualizar Datos Existentes>.

Nota: Para ver las imágenes de los prototipos de interfaz ver el ANEXO 2.

3.7 Conclusiones parciales

Con el desarrollo de este capítulo, se han obtenido resultados sustanciales para la realización de los flujos de trabajo posteriores. Se realizó la descripción detallada de la propuesta de solución a través de la modelación del negocio propuesto; identificando los actores y trabajadores que intervienen, así como los Casos de Uso correspondientes. Además del planteamiento de los requisitos funcionales (RF) y los no funcionales (RNF) de la aplicación. Con el desarrollo de este flujo de trabajo, se han sentado las bases para lograr realizar el análisis y diseño adecuado para la arquitectura de la solución de software propuesta, tema que será abordado en el capítulo siguiente.

4

CAPÍTULO 4

Diseño de la solución propuesta

Introducción:

En este capítulo se desarrollará el análisis y diseño de la propuesta de solución a través de los flujos de Análisis y Diseño, donde se presentarán los diagramas de clases de análisis y de diseño de los casos de uso representados en el capítulo anterior, así como el diagrama de despliegue con la ubicación de los nodos donde va a estar la aplicación y el de componentes para una mejor descripción de la solución propuesta.

4.1 Modelo de Análisis

El modelo de análisis debe ser una representación mínima del sistema que se está modelando, suficiente para capturar la lógica esencial del sistema, sin entrar en temas de rendimiento o construcción. (Rumbauhg, y otros, 2000)

El modelo de análisis se describe a través de un lenguaje desarrollador, proporciona la vista interna del sistema. Este modelo está estructurado por clases y paquetes estereotipados. Ayuda fundamentalmente a los desarrolladores para comprender como ser diseñado e implementado el sistema. Además define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo de casos de uso. Se utiliza como entrada en las actividades de diseño e implementación. Este modelo puede considerarse una primera aproximación al modelo de diseño, aunque es un modelo por si mismo. (Rumbauhg, y otros, 2000)

4.1.1 Diagrama de Clases del Análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

Las clases de análisis representan abstracciones de clases y posiblemente de subsistemas del diseño del sistema. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis que se llama "realizaciones de caso de uso-análisis" (Rumbauhg, y otros, 2000). En el ANEXO 5 se puede visualizar con mayor detenimiento la descripción de las clases del análisis.

A continuación se muestran los diagramas de clases del análisis perteneciente a cada caso de uso modelado en el sistema.

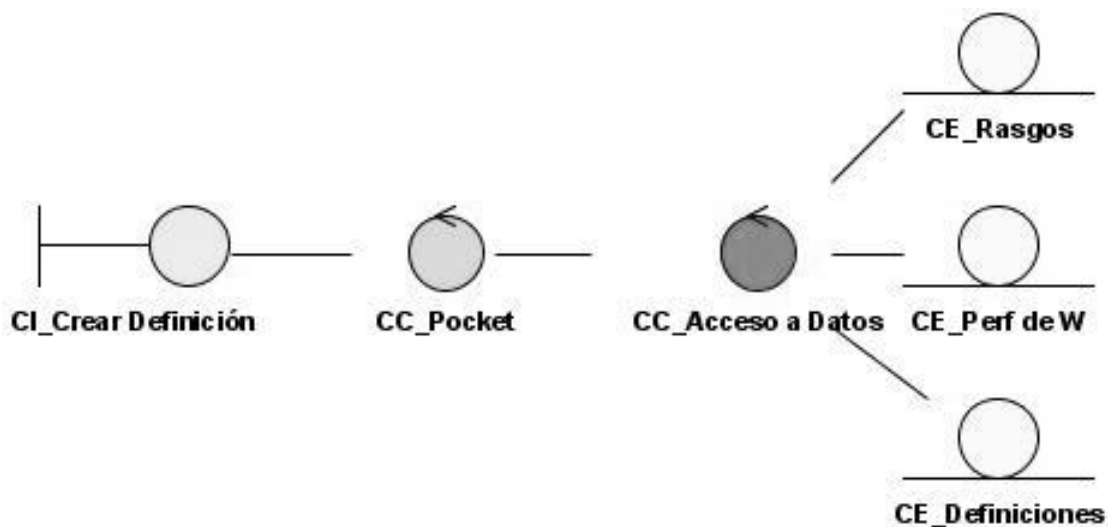


Figura 7: Diagrama de Clases CU <Crear Definición >.

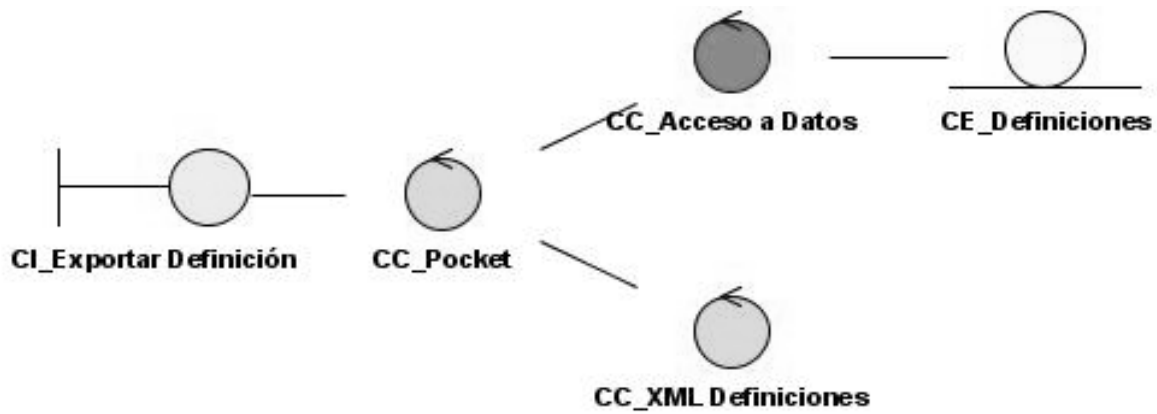


Figura 8: Diagrama de Clases CU <Exportar Definición Creada>.

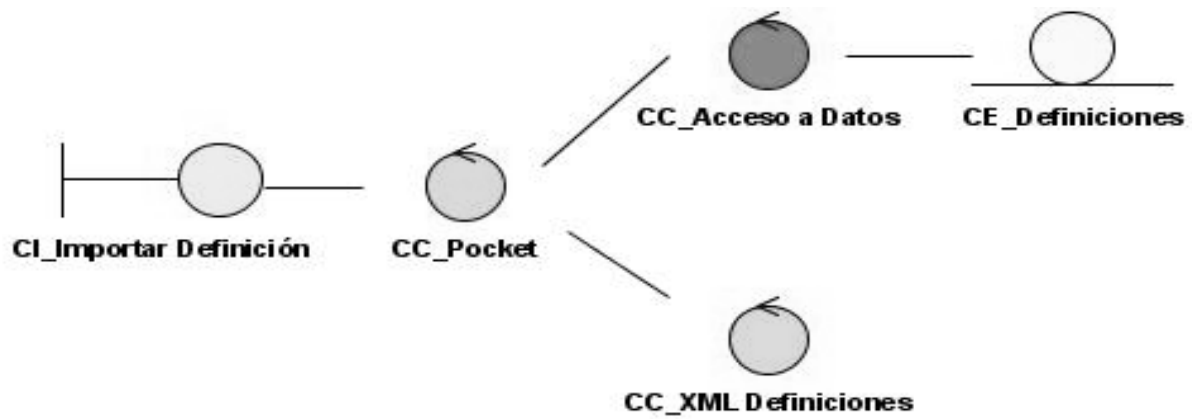


Figura 9: Diagrama de Clases CU <Importar Definición Captada>.

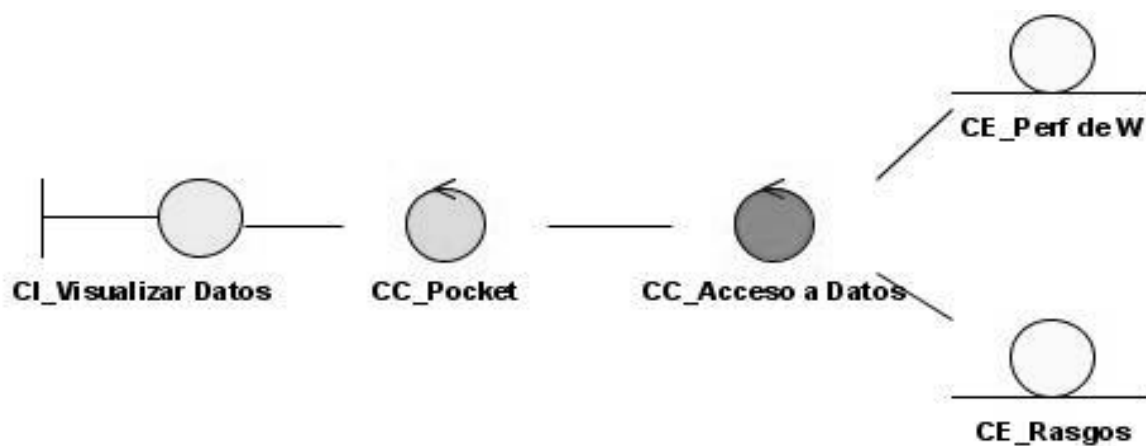


Figura 10: Diagrama de Clases CU <Visualizar Datos Existentes>.

4.2 Diagrama de Interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Ver ANEXO 3.

Los diagramas de interacción están constituidos por dos tipos: los diagramas de Colaboración y los diagramas de secuencia. Ambos expresan información similar, pero en una forma diferente.

4.2.1 Diagrama de Secuencia

El diagrama de Secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y los enlaces entre ellos. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos.

4.2.2 Diagrama de Colaboración

El diagrama de Colaboración representa las interacciones que ocurren entre los objetos que participan en una situación determinada destacando la forma en que las operaciones se producen en el tiempo. Representa una forma de indicar el período durante el que un objeto está desarrollando una acción directamente o a través de un Procedimiento.

4.3 Diagrama de Clases de Diseño

El diagrama de clases de diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

Normalmente contiene la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.

➤ Dependencias.

A diferencia del modelo conceptual, un diagrama de este tipo contiene las definiciones de las entidades del software en vez de conceptos del mundo real.

Para elaborar los diagramas de clases del diseño se tuvo en cuenta el patrón de Arquitectura por capas, por lo que cada diagrama cuenta de las clases interfaces que pertenecen a la capa de presentación, con dos clases controladoras, de las cuales, una controla todas la funcionalidades del negocio y la otra controla toda la gestión de los ficheros XML ,componen la capa de lógica del negocio, y por último la clase controladora de acceso a dato y las entidades de la base de datos forman la capa de Datos.

A continuación se muestra el diagrama general de clase de diseño.

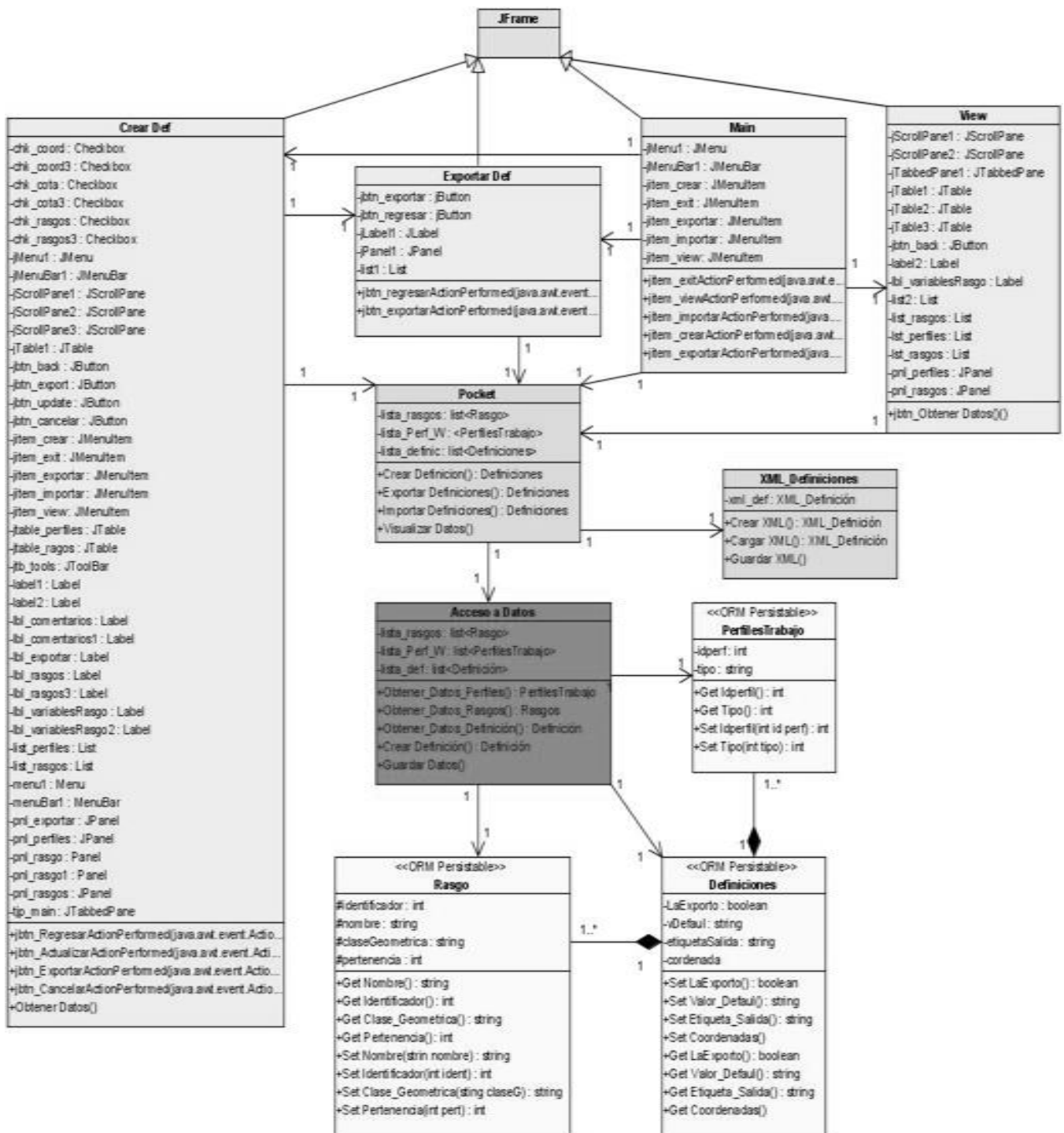


Figura 11: Diagrama de Clases de Diseño General.

4.4 Estructura del Fichero (XML)

El fichero de tipo XML mediante el cual se exportan e importan los datos de valor para el sistema en cuestión tendrá la estructura que se define en la Tabla que se muestra a continuación. Donde el tipo representa si se trata de un Rasgo o un Perfil de Trabajo, Nombre es lo que especifica el Rasgo o el Perfil de trabajo que se este analizando en el momento que se cree la definición, el Id Var es el valor que debe tener esa variable en la BD, las variables virtuales y la Puesta son los atributos que caracterizan la Definición Etiqueta de salida es lo que precisa la Definición en si, la cual admite ser modificada, el campo Valor por Defecto es un identificador que se le ponen a las variables para no perder la organización en el momento de importar estos datos. De esta manera queda conformada la estructura tanto de salida como de entrada del fichero (XML) que se usa para realizar las funcionalidades Importar y Exportar.

<Definiciones>

<Rasgos>

<Rasgo Nombre = “ ”, Id Var = “ ”, Variables Virtuales = “ ”, Puesta = “ ”, Etiqueta de salida = “ ”, Valor por defecto = “ ”>

</Rasgos>

<Perfiles de Trabajo>

< Perfil de Trabajo Nombre= “ ”, Id Var = “ ”, Variables Virtuales = “ ”, Puesta = “ ”, Etiqueta de salida = “ ”, Valor por defecto = “ ” >

</Perfiles de Trabajo>

</Definición>

4.5 Principios de Diseño

Cuando se trata de diseño, independientemente de cuales sean las particularidades del mismo, es necesario que se base en el usuario, y en este caso particularmente se está hablando de trabajadores; que no precisamente tienen que ser informáticos o sea que no requieren de conocimientos avanzados de informática. El diseño no se limita a la apariencia estética, la combinación de colores, o a un logo más o menos acertado. De él depende que la información sea útil, que los servicios se puedan usar, es decir, el diseño convierte a una aplicación en algo atractivo para el usuario.

Existen además principios para garantizar la usabilidad en el diseño, los cuales son tan importantes como garantizar una interfaz bien diseñada acompañado de un buen diseño gráfico, usabilidad y distribución de los contenidos, para garantizar una mayor calidad del producto , a continuación se muestran algunos de estos principios.

- Exija poco esfuerzo físico:
 - Que utilice de manera razonable las fuerzas necesarias para operar.
 - Que minimice las acciones repetitivas.
 - Que minimice el esfuerzo físico continuado.
- Uso equiparable:
 - Que evite segregar o estigmatizar a cualquier usuario.
 - Que el diseño sea atractivo para todos los usuarios.
- Simple e intuitivo:
 - Que elimine la complejidad innecesaria.
 - Que sea consistente con las expectativas e intuición del usuario.
 - Que proporcione avisos eficaces siempre que vaya a realizar una acción relevante sobre el sistema y métodos de respuesta durante y tras la finalización de la tarea.
- Limitar el número de acciones que puede realizar el usuario sobre la aplicación, mostrando sugerencias (opciones) para cada posible acción, evitando así al máximo los errores de usuario.
- Mostrar la mayor cantidad de información acerca de las opciones brindadas en un momento dado, de modo que el usuario siempre sepa cuáles son las operaciones a las que puede acceder y en qué consiste exactamente cada una.

4.5.1 Estándares en la interfaz de la aplicación

El medio mediante el cual el usuario interactúa con el sistema es la interfaz grafica, lo cual implica que esta debe ser lo más amena y amigable posible para lograr que el usuario se sienta identificado con la misma. Para el diseño de la interfaz del sistema se tuvieron en cuenta las características que

posee la aplicación con la que hoy se cuenta en la ONRM además de algunas funcionalidad que fue necesario incluir, se cuenta además con una forma principal que consta de un menú que permite realizar las operaciones que se requieren para dar solución a las peticiones del usuario. Se usan colores claros para no cansar la vista de los usuarios y se reflejan solamente aspectos de importancia para este lo que facilita el trabajo con el sistema además y evita que sea tedioso.

4.5.2 Tratamiento de excepciones

El tratamiento de errores facilita el buen funcionamiento de la aplicación, Una excepción es un evento que ocurre durante la ejecución del programa que interrumpe el flujo normal de las sentencias por lo que cuando se produce un error debido a la entrada o sencillamente la manipulación incorrecta de algún dato por parte del usuario se muestra un mensaje de error que le permita a este percatarse de que la acción que realizo es incorrecta y pueda corregir su error. En el sistema se controlan los errores usando las técnicas de Java así como sus mensajes de alerta.

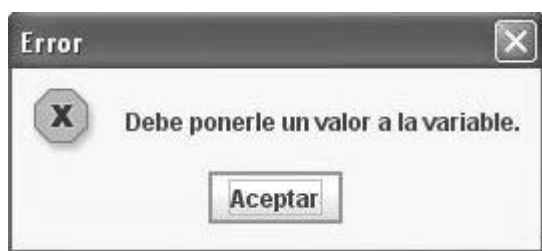


Figura 12: Tratamiento de excepciones.

4.7 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución.

El diagrama cuenta con un nodo que representa la computadora cliente donde estaría instalada la aplicación, otro nodo que simboliza el servidor de la base de datos y el tercero representa a la PocketPC.

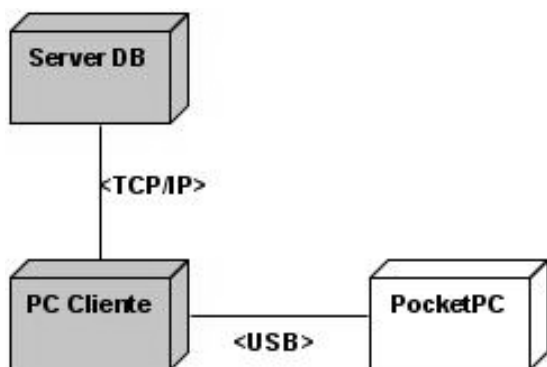


Figura 13: Diagrama de Despliegue.

Descripción de los componentes del Diagrama de Despliegue

Componentes	Descripción
PC Cliente	Ordenador desde donde los usuarios efectúan los servicios.
Server DB	Ordenador donde se encuentra la base de datos con la que trabaja la aplicación.
PocketPC	Pequeño ordenador de bolsillo que es usado para la captación de minerales en el campo.
USB	Protocolo utilizado para la conexión entre las PC Cliente y el PocketPC.
TCP/IP	Protocolo utilizado para la conexión entre las PC Cliente y el Server DB.

Tabla 10: Descripción de los componentes del Diagrama de Despliegue.

4.8 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

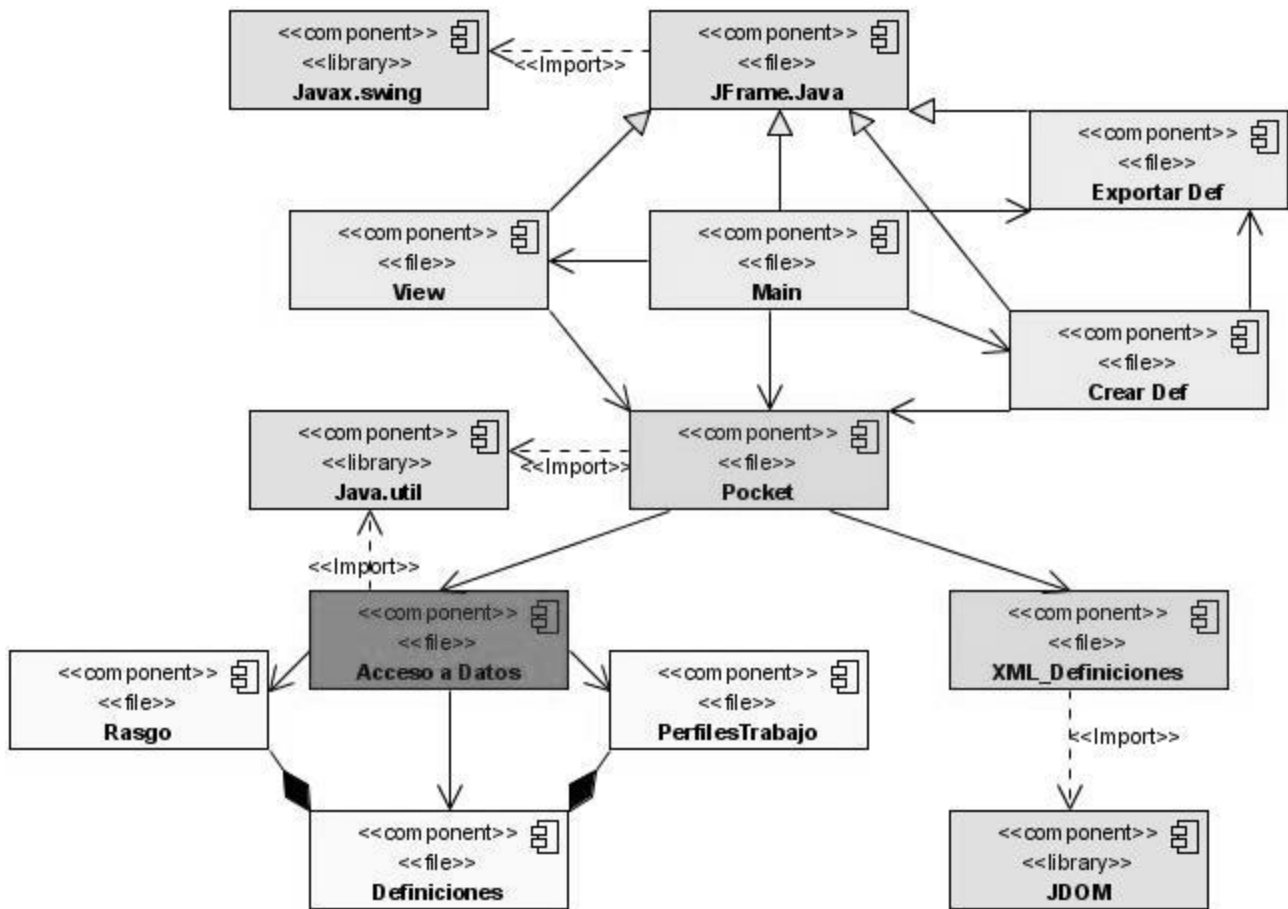


Figura 14: Diagrama de Componentes.

4.9 Conclusiones parciales

Con la presentación de este capítulo culmina la modelación completa de el presente sistema, para ello se hizo necesario el paso a través de los flujos de trabajo que propone RUP, teniendo en cuenta los resultados obtenidos se concluye que cualquier persona con los conocimientos básicos de Ingeniería de Software puede utilizar los diagramas antes expuestos, como referencia para realizar una operación de mantenimiento, mejora o sencillamente como material de estudio, con la implementación de lo expuesto en el presente documento se lograrían mejores resultados, puesto que para probar su eficiencia se hace necesario el producto final. Finalmente se lograron los objetivos planteados al inicio.

CONCLUSIONES GENERALES

Una vez elaborada la investigación, se arriba a las siguientes conclusiones:

- El prototipo no funcional favorece la gestión de Definiciones de la PocketPC para la Oficina Nacional de Recursos Minerales.
- El objetivo elaborado en el diseño de la investigación fue cumplido, comprobándose la idea a defender como respuesta del problema a resolver que originó esta investigación, tributando directamente a la gestión de las Definiciones para la PocketPC.
- El empleo de los métodos teóricos y empíricos facilitó conocer el estado del objeto de estudio.
- La entrevista aplicada en la ONRM, para evaluar el problema, corroboró la necesidad de perfeccionar el proceso de gestión de las Definiciones para la PocketPC.
- Las tareas investigativas facilitaron una adecuada organización de la investigación.

RECOMENDACIONES

Partiendo del interés puesto en el desempeño de este trabajo así como los beneficios que aporta para el personal interesado, se proponen como recomendaciones:

- Que se realice la implementación de la aplicación propuesta.
- Que este material sea tomado como material de consulta por el personal que vaya a desarrollar un sistema similar.
- Que se emplee como material de apoyo para los futuros desarrolladores en el momento de implementar.
- Que sea consultada la referencia bibliográfica para mitigar cualquier duda que pudiera surgir.

Bibliografía

- 2007.** Ubuntu Life. *Importar una base de datos Access con Kexi*. [En línea] 10 de mayo de 2007. <http://ubuntulife.wordpress.com/2007/05/10/importar-una-base-de-datos-access-con-kexi/>.
- Armando Ortiz Cabrera, Yamiskel Pérez Velázquez. 2007.** *Sistema Informativo de partes de producción y de venta de productos del Grupo de la Electrónica del Ministerio de la Informática y las Comunicaciones*. 2007.
- Dieguez Escalona, Jorge Luis. 2006.** *Sistema de Seguridad Centralizada*. 2006.
- DMSOFT Technologies.** db Convert. *DBConvert for Access & PostgreSQL*. [En línea] <http://dbconvert.com/>.
- Doral, Carlos. 2005.** Programas a medidas para empresas. *Programas a medidas para empresas*. [En línea] 2005. [Citado el: 13 de marzo de 2008.] <http://www.haztuprograma.com/tiposAplicaciones.html#win>.
- Englander, Robert.** *Developing Java Beans*.
- Española, Real Academia. 2005.** Real Academia Española. *Real Academia Española*. [En línea] 2005. [Citado el: 1 de febrero de 2008.] <http://www.rae.es>.
- Espinosa Hidalgo, Amado.** *SISTEMA PARA LA ADMINISTRACIÓN UNIFICADA DE USUARIOS*.
Estimación_del_esfuerzo_basada_en_casos_de_usos. [En línea] <http://teleformacion.uci.cu/mod/resource/view.php?id=9007>.
- Flanagan, David.** *Java en pocas palabras*.
- García Molina, Jesus J., Ana Moreira, y Gustavo Rossi. 2004.** Asociación de Técnicos de Informática. *Asociación de Técnicos de Informática*. [En línea] 2004. [Citado el: 26 de marzo de 2008.] <http://www.ati.es/novatica/2004/168/168-4.pdf>.
- Geominera, Empresa. 2005.** *Documentación Técnica de GeoDatoIC (V2.0)*. Santiago de Cuba : Oriente, 2005.
- 2006.** Glosario- Great Sand Dunes National Park and Preserve. *Glosario- Great Sand Dunes National Park and Preserve*. [En línea] 2006. [Citado el: 1 de febrero de 2008.] <http://nps.gov>.
- González, V.H.D., F.M. Lamas, J.N.J. Fabre. 2001.** Curso de Bases de Datos y PostgreSQL. *Curso de Bases de Datos y PostgreSQL*. [En línea] 2001. <http://es.tldp.org/Tutoriales/NOTAS-CURSO- BDD/notas-curso- BD/notas-curso-BD.html>.
- Gracia, Joaquin. 2005.** Desarrollo de Software Orientado a Objetos. [En línea] 7 de mayo de 2005. <http://www.ingenierossoftware.com/analisisydiseno/uml.php>.
- Hernandez Orallo, Enrique;.** 2007. El Lenguaje Unificado de Modelado (UML). *El Lenguaje Unificado de Modelado (UML)*. [En línea] 2007. [Citado el: 4 de marzo de 2008.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

Informática, Grupo Universitario de. 2006. *Eclipse: una herramienta profesional al alcance de todos.* 2006.

Larman, C. 2004. UML y Patrones. [aut. libro] Introducción al análisis y diseño orientado a objetos. *UML y Patrones.* . Habana : Félix Varela, 2004.

León Pavón, Eduardo. 2007. Blog de Eduardo León. *Visual Paradigm, una herramienta de lo más útil.* [En línea] 2 de abril de 2007. <http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.

Libre, Grupo de Usuarios de Software. 1999 - 2008. www.somoslibres.org. *www.somoslibres.org*. [En línea] SurNoticias.com, 1999 - 2008. [Citado el: 28 de febrero de 2008.] <http://www.somoslibres.org/modules.php?name=News&file=article&sid=381>.

Luque, Eyllin Hernández. 2007. *Aplicación Web para la gestión de información de instrumentos meteorológicos del Instituto de Meteorología.* 2007.

Malonna, Javier. 2002. Libertonia . *Libertonia* . [En línea] 2002. [Citado el: 16 de abril de 2008.] <http://libertonia.escomposlinux.org/story/2005/9/4/133646/3993>.

Manager, Free Download. Free Download Manager. *Free Download Manager.* [En línea] [Citado el: 23 de marzo de 2008.] [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p).

Martinez, Rafael. 1994 - 2008. El rincón de Linux para hispanohablantes. *El rincón de Linux para hispanohablantes.* [En línea] 1994 - 2008. http://www.linux-es.org/sobre_linux.

Molpeceres, Alberto. 2002. JavaHispano. *JavaHispano.* [En línea] 2002. [Citado el: 10 de marzo de 2008.] <http://www.javahispano.org/articles.article.action?id=76>.

Mora, Roberto. 2004. Adictos al Trabajo. *Adictos al Trabajo.* [En línea] 2004. <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm>.

2007. Netbeans. *Netbeans.* [En línea] 2007. [Citado el: 23 de marzo de 2008.] http://www.netbeans.org/index_es.html.

ORIENTE, EMPRESA GEOMINERA. Geominera Oriente. [En línea] <http://www.geominera.co.cu/>.

Pressman, Roger S. 2002. *Ingeniería del software. Un enfoque práctico, 5ta ed.* 2002.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2000. *El Lenguaje Unificado de Modelado. Manual de referencia.* Madrid : s.n., 2000.

SIDAR. 2007. Fundación Sidar - Acceso Universal, Seminario SIDAR. [En línea] 2007. <http://www.sidar.org/recur/desdi/usable/dudt.php>.

Silva, Emilio. 2006. ¡GNU No es Unix! – Software Libre, Sociedad Libre. *¡GNU No es Unix! – Software Libre, Sociedad Libre*. [En línea] 2006. [Citado el: 1 de abril de 2008.] <http://gnu.cofman.dk/home.es.html>.

UCI. Conferencia_Gestion_Proyectos_Estudiantes. [En línea]
<http://teleformacion.uci.cu/mod/resource/view.php?id=9007>.

Villalobos. 2006. Java. *Java*. [En línea] 2006. [Citado el: 1 de abril de 2008.]
http://java.ciberaula.com/articulo/nuevo_java_5.

Zamora, Universidad de. 2008. Extensión universitaria de Zamora. *Extensión universitaria de Zamora*. [En línea] 2008. [Citado el: 13 de febrero de 2008.] <http://www.utpl.edu.ec/blog/zamora/2008/02/13/arquitectura-por-capas/>.

GLOSARIO DE TERMINOS

MINBAS: Ministerio De Las Industrias Básicas.

ONRM: Oficina Nacional De Recursos Minerales.

IGP: Instituto de Geología y Paleontología.

PNICG: Proyecto Nacional de Informatización del Conocimiento Geológico.

GeoDato: Sistema que permitiera la carga de los datos primarios de los informes que contienen los archivos del servicio geológico nacional a una base digital.

GeoDato^{IC}: Es el software que permite la crear y mantener de las Bases de datos Geológicas

Pocket PC: Este es un pequeño ordenador de mano del tamaño de un gran teléfono móvil. También es llamado un Asistente Digital Personal (PDA).

Software: Equipamiento lógico o soporte lógico de un computador digital, comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware).

C++: Es un lenguaje híbrido, que se puede compilar y resulta más sencillo de aprender para los programadores que ya conocen C. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Las principales características son abstracción (encapsulación), el soporte para programación orientada a objetos (polimorfismo) y el soporte de plantillas o programación genérica (templates). Es un lenguaje que abarca tres paradigmas de la programación: La programación estructurada, la programación genérica y la programación orientada a objetos.

Java: Lenguaje de programación orientado a objetos con el que se puede realizar cualquier tipo de programa, es un lenguaje muy extendido, es un lenguaje independiente de la plataforma, es compilado en un bytecode que es interpretado desarrollado por la compañía Sun Microsystems a principios de los 90.

C#: Lenguaje de programación orientado a objetos, evolución del lenguaje C++, desarrollado por Microsoft.

XML: Extensible Markup Language (Lenguaje extensible de etiquetas) Es un meta-lenguaje que permite definir lenguajes de marcado adecuado a usos determinados. Se propone como lenguaje de bajo nivel (a nivel de aplicación, no de programación) para intercambio de información estructurada entre diferentes plataformas.

GNU: Conjunto de programas desarrollados por miembros de la Fundación por el Software Libre, son de uso gratuito (FSF- Free Software Foundation).

GNU/LINUX: Es un sistema operativo, es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Es multitarea, multiusuario, multiplataforma y multiprocesador.

GPL: Es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software Libre (General Public License).

Debian: Asociación o comunidad conformada por desarrolladores y usuarios que pretenden crear y mantener un Sistema operativo GNU basado en software libre precompilado y empaquetado en un formato sencillo en múltiples arquitecturas y en varios núcleos.

GNOME: Es un entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Windows. Forma parte oficial del proyecto GNU.

Intel x86: Es la denominación genérica dada a ciertos microprocesadores, sus compatibles y a la arquitectura básica de estos procesadores, por la terminación de sus nombres: 8086, 80286, 80386 y 80486. Los sucesores del 80486 pasarán a ser llamados por nombres no numéricos, bajo la denominación Pentium, sin embargo todavía se los llama procesadores de la familia x86.

AMD64: Inicialmente conocida como x86-64, es una arquitectura basada en la extensión del conjunto de instrucciones x86 para manejar direcciones de 64 bits realizada por AMD.

PowerPC: Usualmente abreviada PPC es el nombre original de la arquitectura de computadoras de tipo RISC, fue desarrollada por IBM, Motorola y Apple. Los procesadores de esta familia son producidos por IBM y Freescale Semiconductor que es la división de semiconductores y microprocesadores de Motorola, siendo utilizados principalmente en ordenadores o computadores Macintosh de Apple Computer

SPARC: (Scalable Processor ARChitecture) es una arquitectura RISC big-endian. Es decir, una arquitectura con un conjunto reducido de instrucciones.

Firewall: Es un cortafuegos (informática), un elemento utilizado en redes de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas

Sudo: (de las siglas en inglés de superuser -o substitute user- do) es una utilidad de los sistemas operativos tipo Unix, como Linux, BSD, o Mac OS X, que permite a los usuarios ejecutar programas con los privilegios de seguridad de otro usuario (normalmente el usuario root) de manera segura.

Root: Es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario). Root es también llamado super-usuario.

UTF-8: (8-bit Unicode Transformation Format) es una norma de transmisión de longitud variable para caracteres codificados utilizando Unicode, creada por Rob Pike y Ken Thompson.

PostgreSQL: Servidor de Base de Datos relacional libre, liberado bajo la licencia BSD, es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

COCOMO: Modelo Constructivo de Coste, permite realizar estimaciones y planificaciones de proyectos de sistemas de información.

ANEXOS

ANEXO 1: Entrevista realizada a un directivo de la Oficina Nacional de Recursos Minerales.

Estimado directivo, estudiantes de 5to año de la Universidad de las Ciencias Informática realizan una investigación referente a Implementar una aplicación de escritorio, para controlar los procesos de creación de las Definiciones para la PocketPC. Su experiencia y criterios serán de gran ayuda para la investigación.

Datos Personales: _____

Años de experiencia: _____

1. Actualmente ¿Cómo gestionan los procesos de creación de las Definiciones para la PocketPC?
2. ¿Qué información es la que se va a gestionar?
3. ¿Qué son las Definiciones?
4. ¿Qué son los rasgos, los perfiles de trabajo, las variables virtuales y las elementales, que relación hay entre ellos?
5. ¿Existe una persona específica que cumple con esta labor?
6. Una vez creada y almacenada la Definición las personas que tenga acceso a ella. ¿Pueden modificarlas?
7. ¿La aplicación con la que cuentan actualmente cumple con todas las funcionalidades necesarias?
8. ¿Esta aplicación se encuentra en uso?
9. ¿De que manera se realiza actualmente este proceso?

Existe alguna otra persona, que pueda brindar más información para enriquecer los conocimientos referentes a lo que se necesita saber para poder desarrollar esta aplicación.

ANEXO 2: Diagramas de actividades de los casos de uso del negocio.

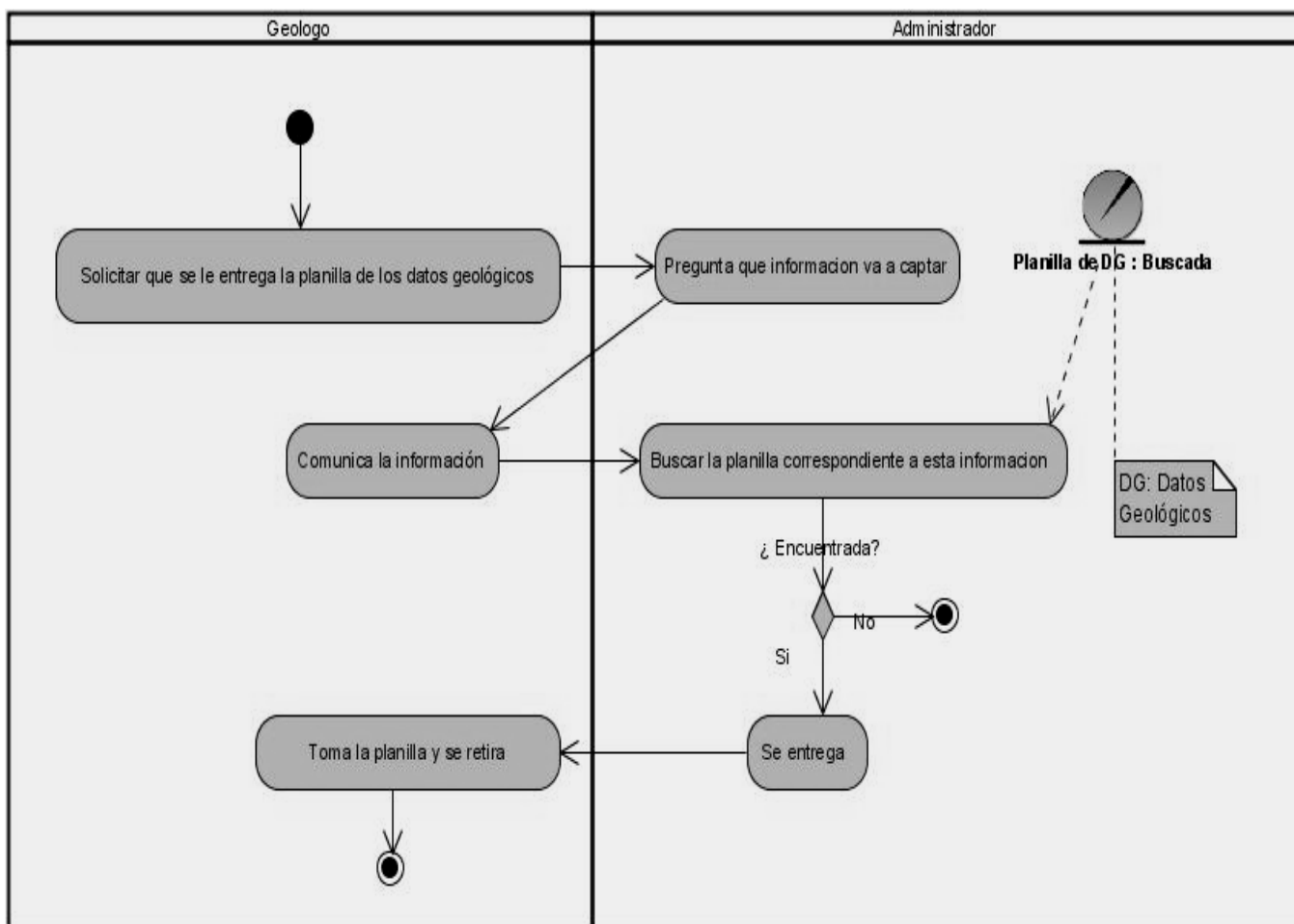


Figura 1: Diagrama de actividades CU<Solicitar planilla de datos geológicos>.

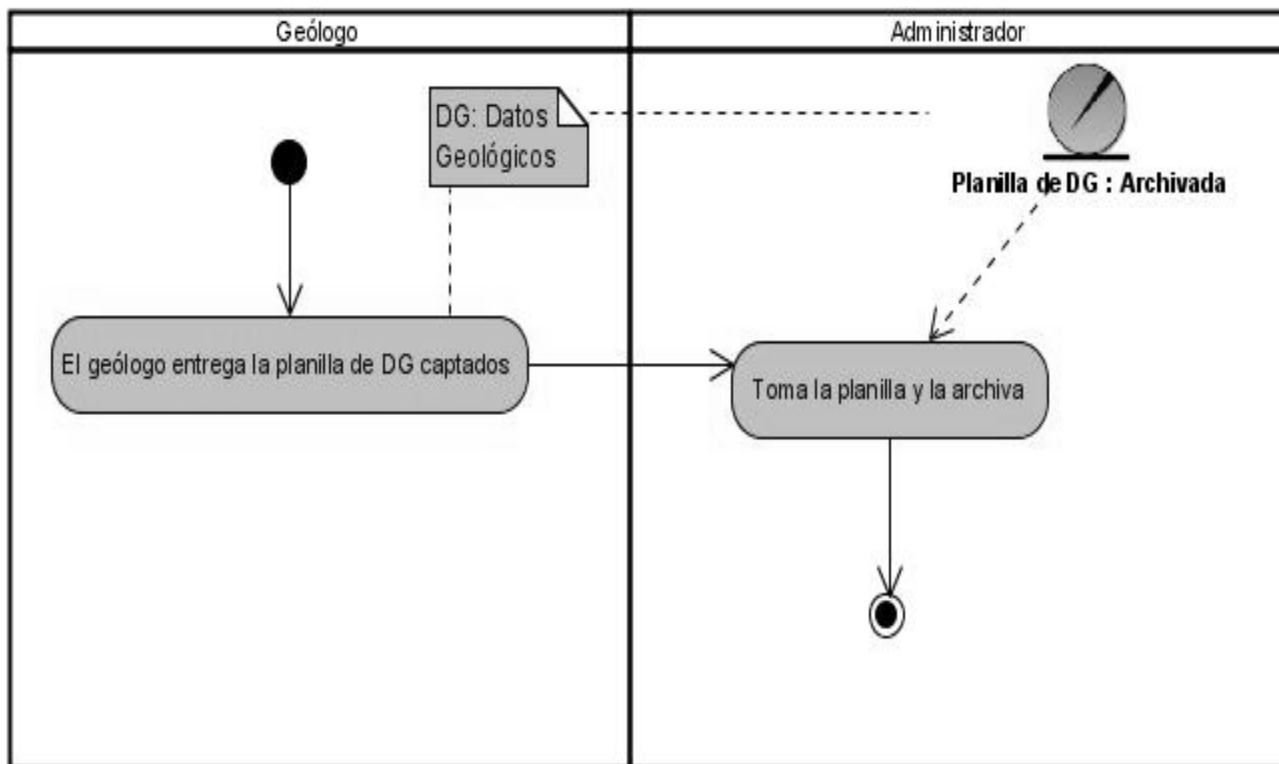


Figura 2: Diagrama de actividades CU<archivar planilla de datos geológicos captados>.

ANEXO 3: Prototipos de Interfaces de Usuarios de los Casos de Uso críticos.

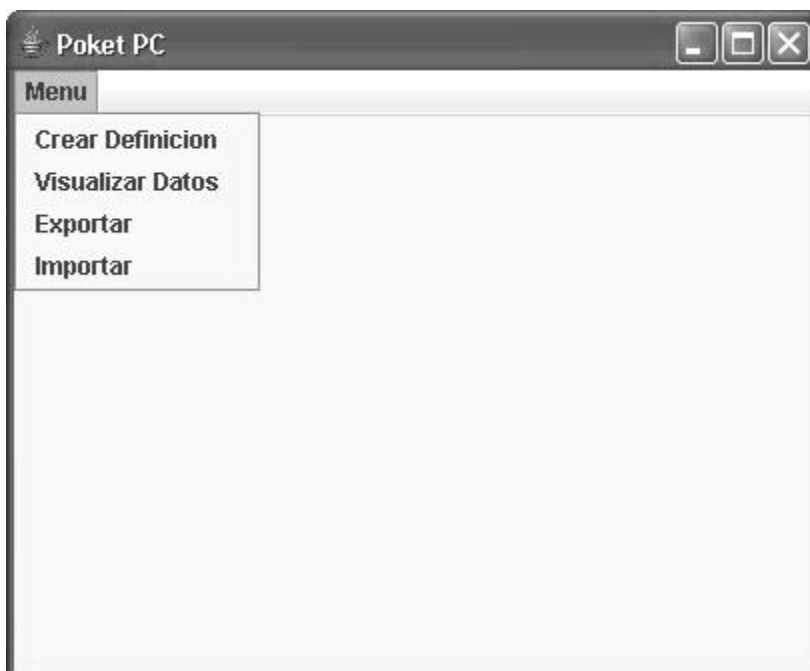


Figura 3: Interfaz Principal.

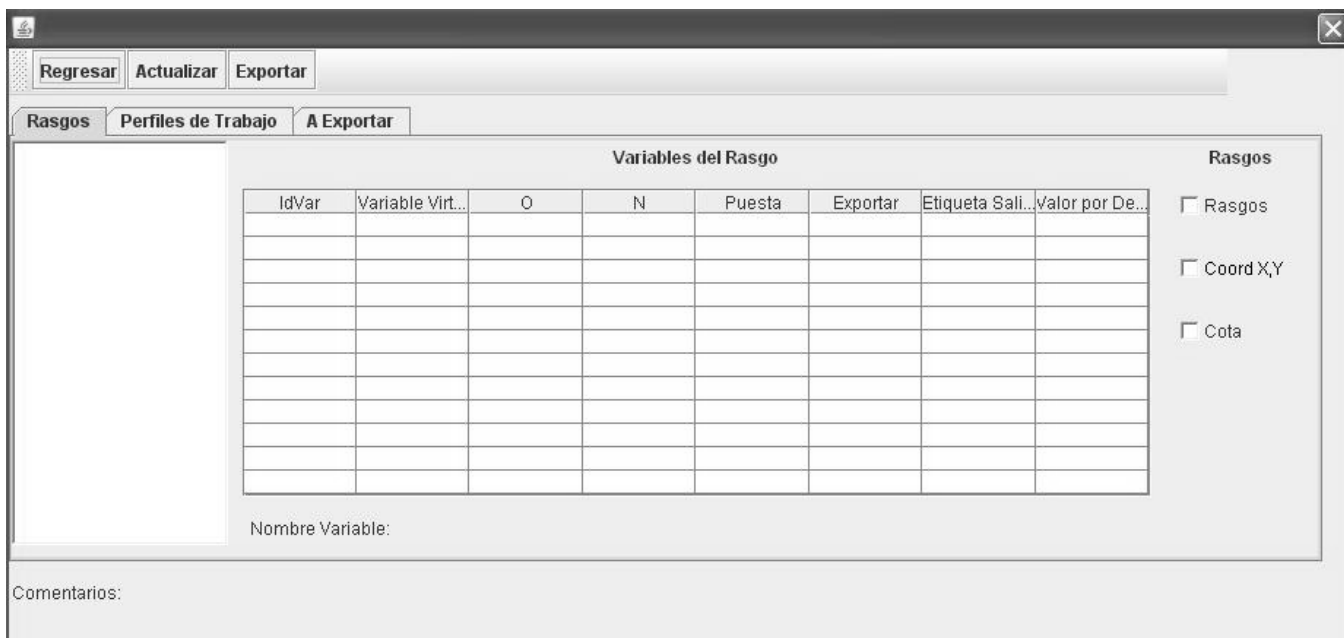


Figura 4: Interfaz del CU <Crear Definición>.

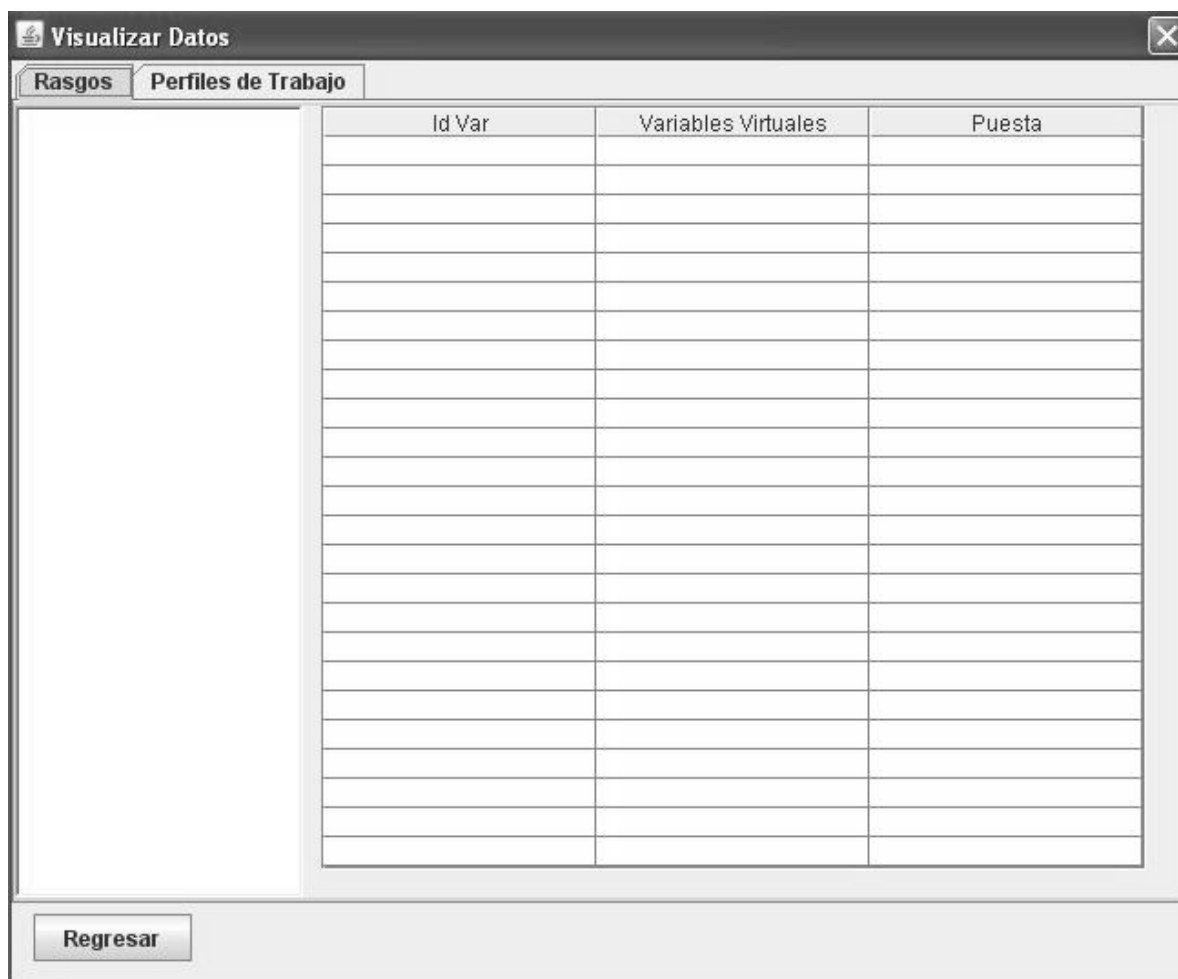


Figura 5: Interfaz del CU <Visualizar Datos Existentes>.



Figura 6: Interfaz del CU <Exportar Definición Creada>.

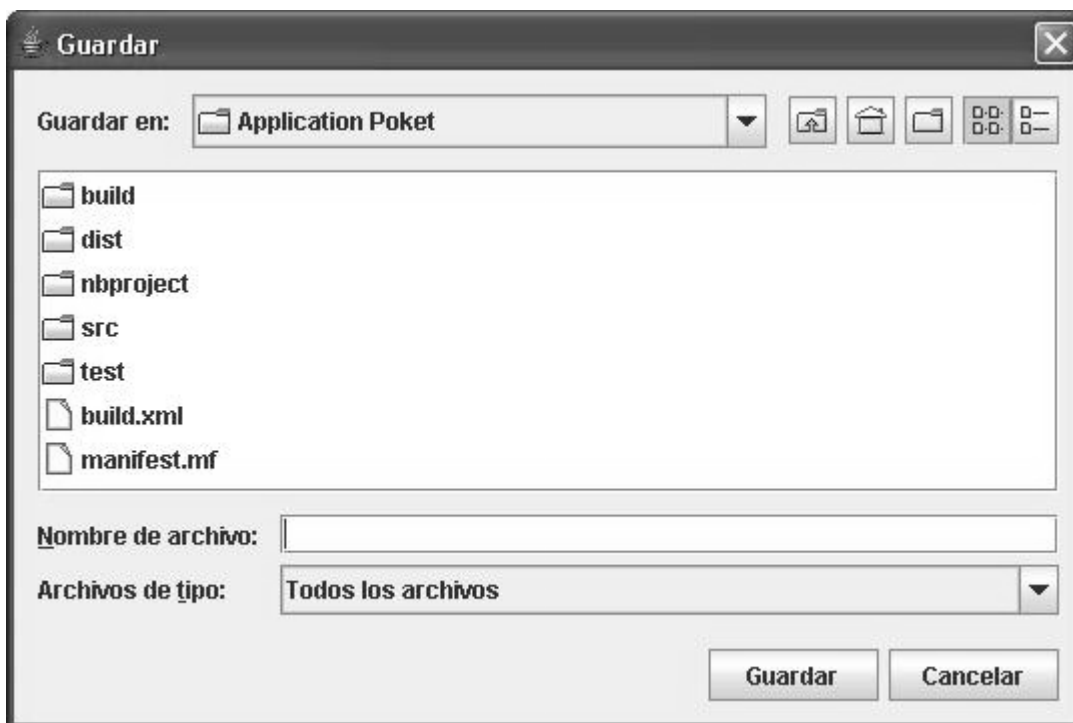


Figura 7: Interfaz del CU <Exportar Definición Creada>.

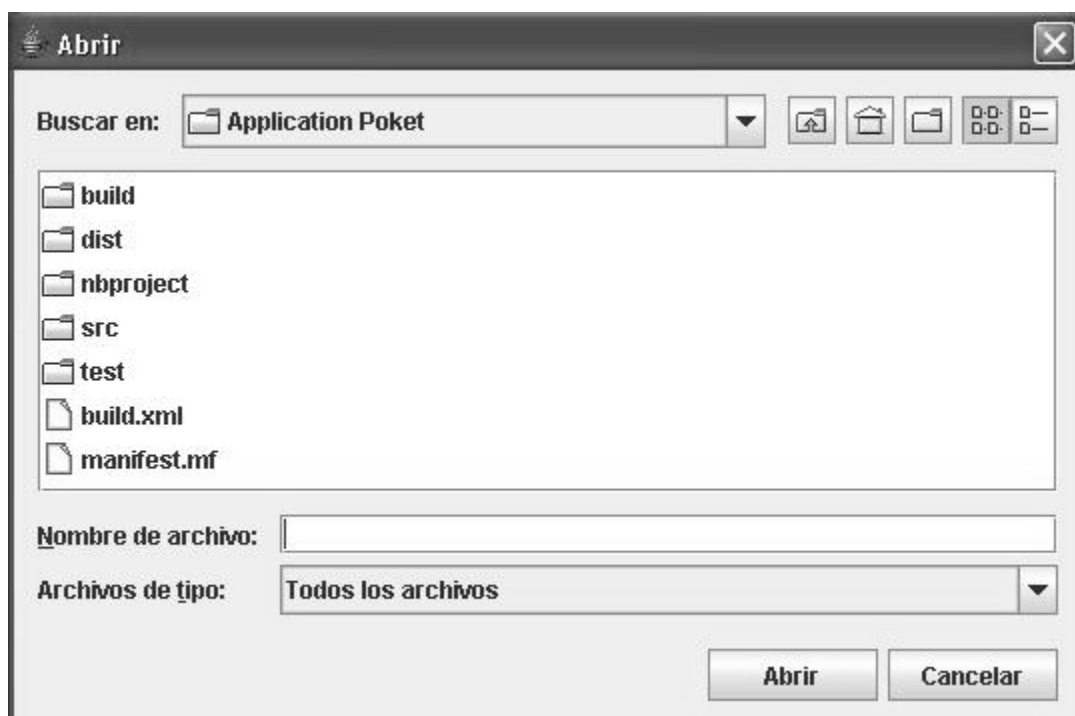


Figura 8: Interfaz del CU < Importar Definición Captada >.

ANEXO 4: Diagramas de Interacciones

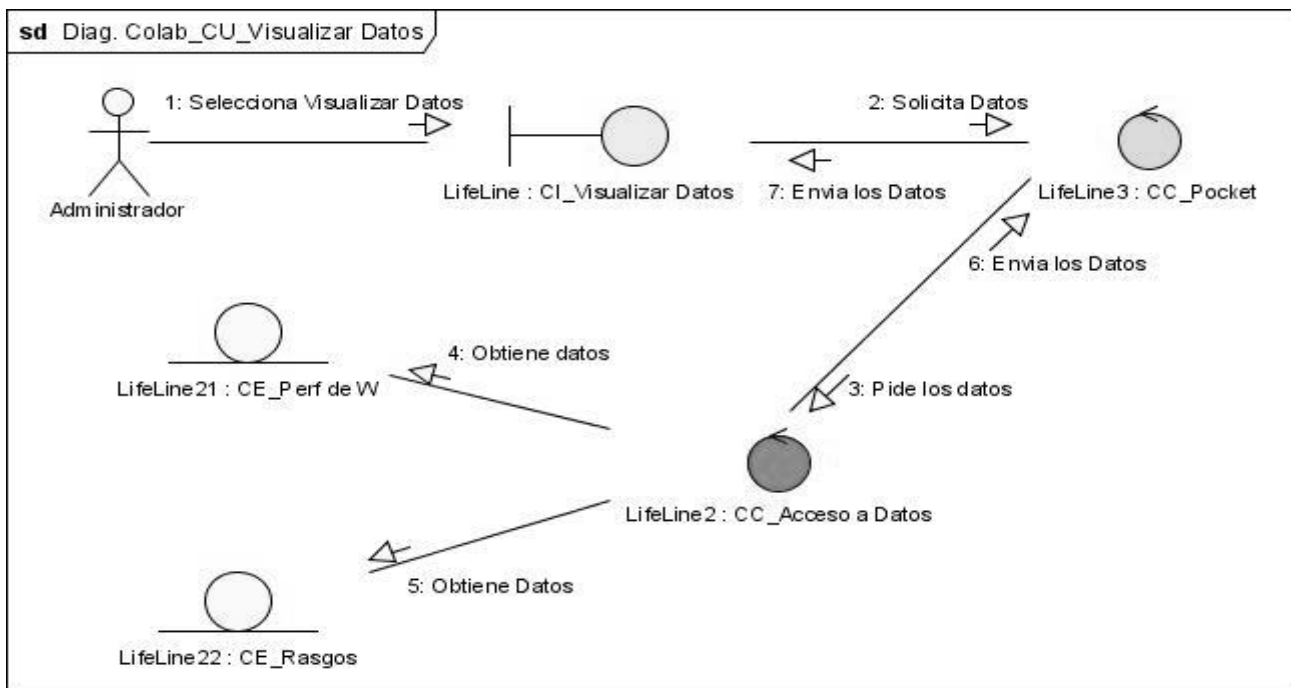


Figura 9: Diagrama de Colaboración CU <Visualizar Datos Existentes>.

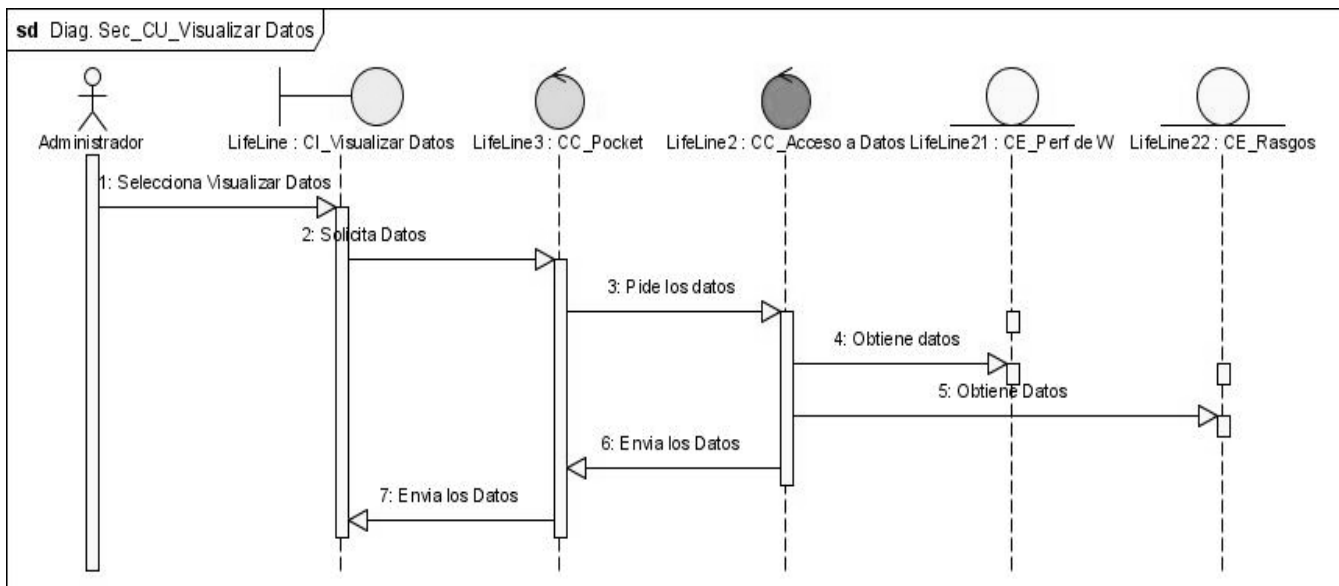


Figura 10: Diagrama de Secuencia CU <Visualizar Datos Existentes>.

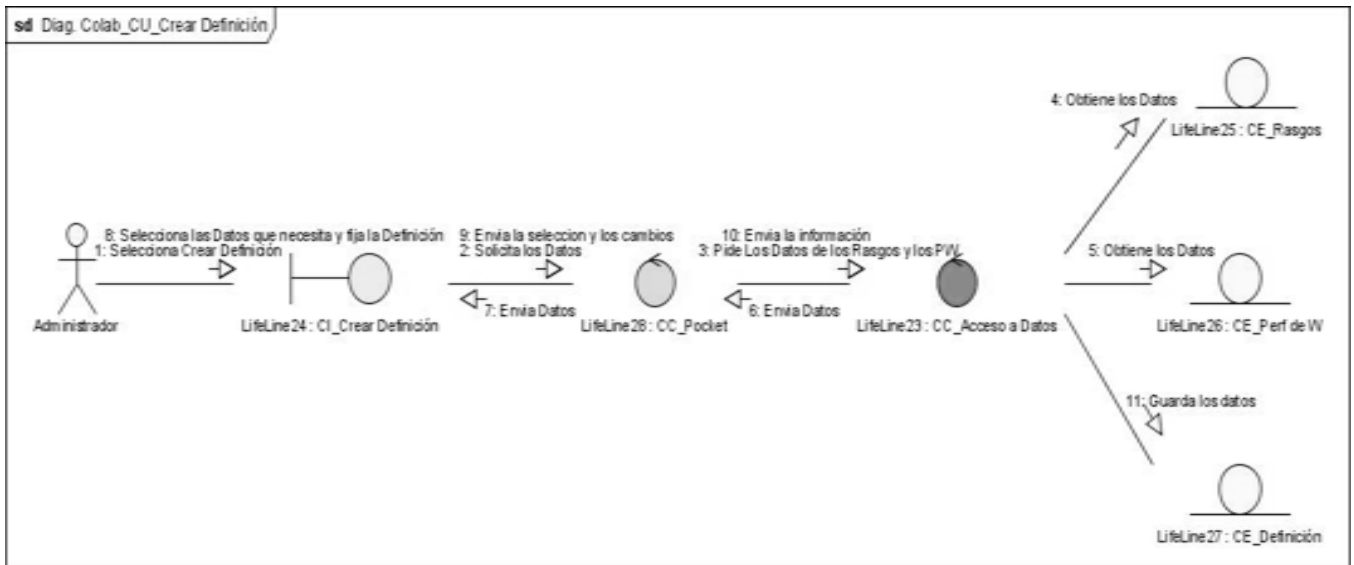


Figura 11: Diagrama de Colaboración CU <Crear Definición>.

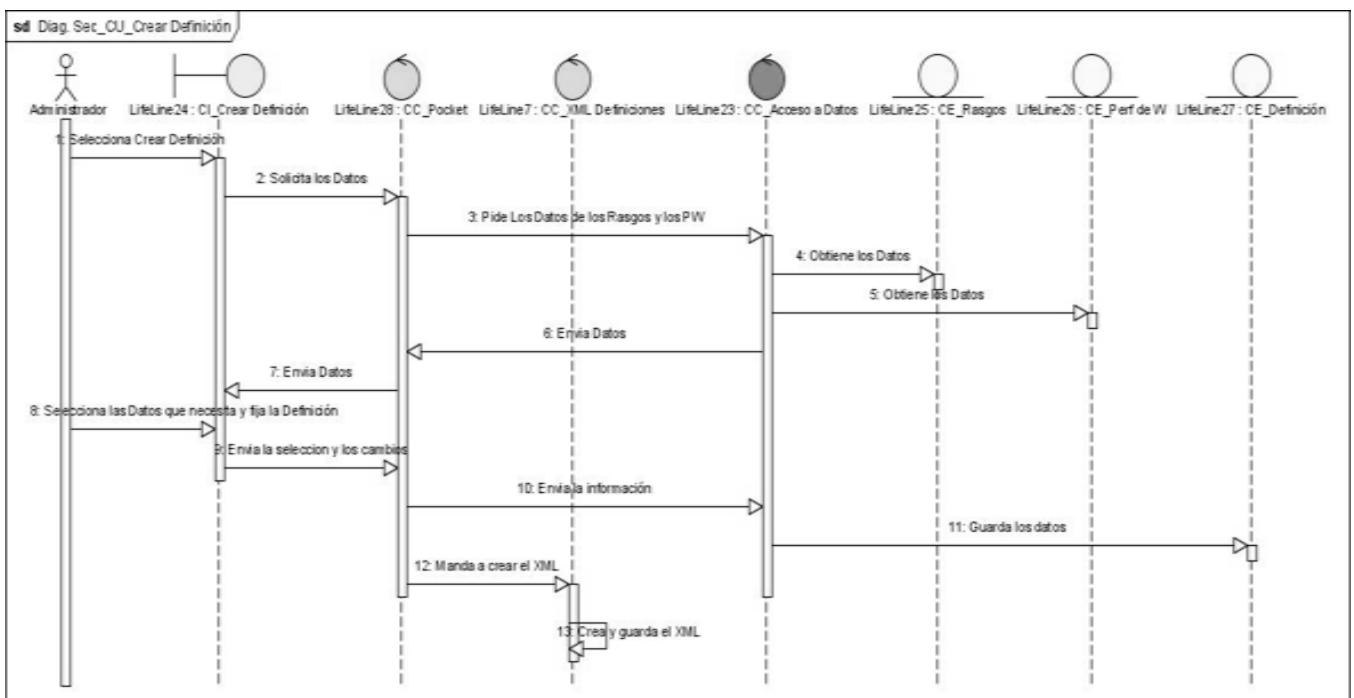


Figura 12: Diagrama de Secuencia CU <Crear Definición>.

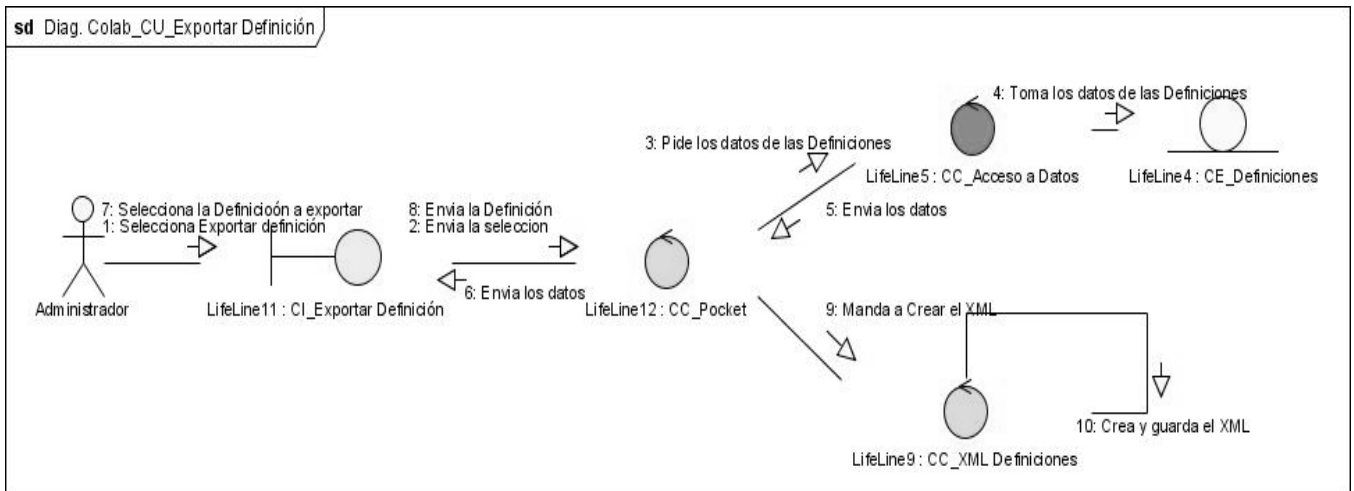


Figura 13: Diagrama de Colaboración CU <Exportar Definición Creada>.

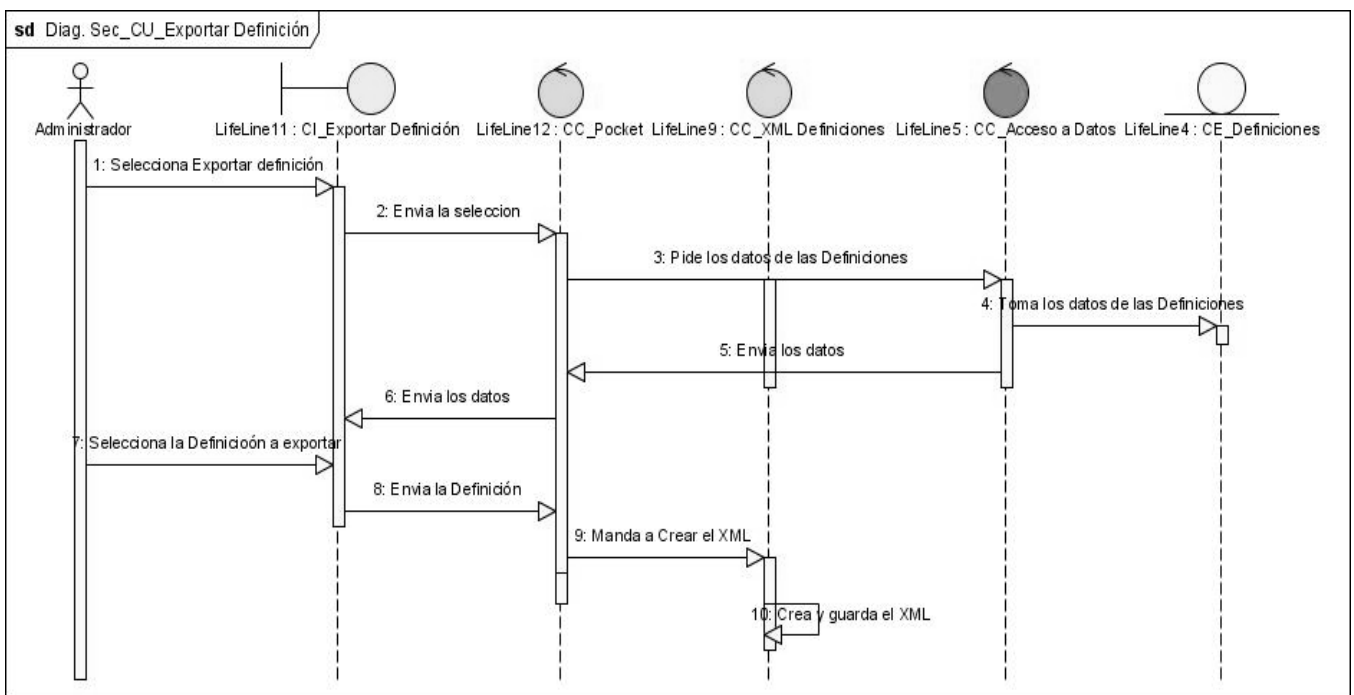


Figura 14: Diagrama de Secuencia CU <Exportar Definición Creada>.

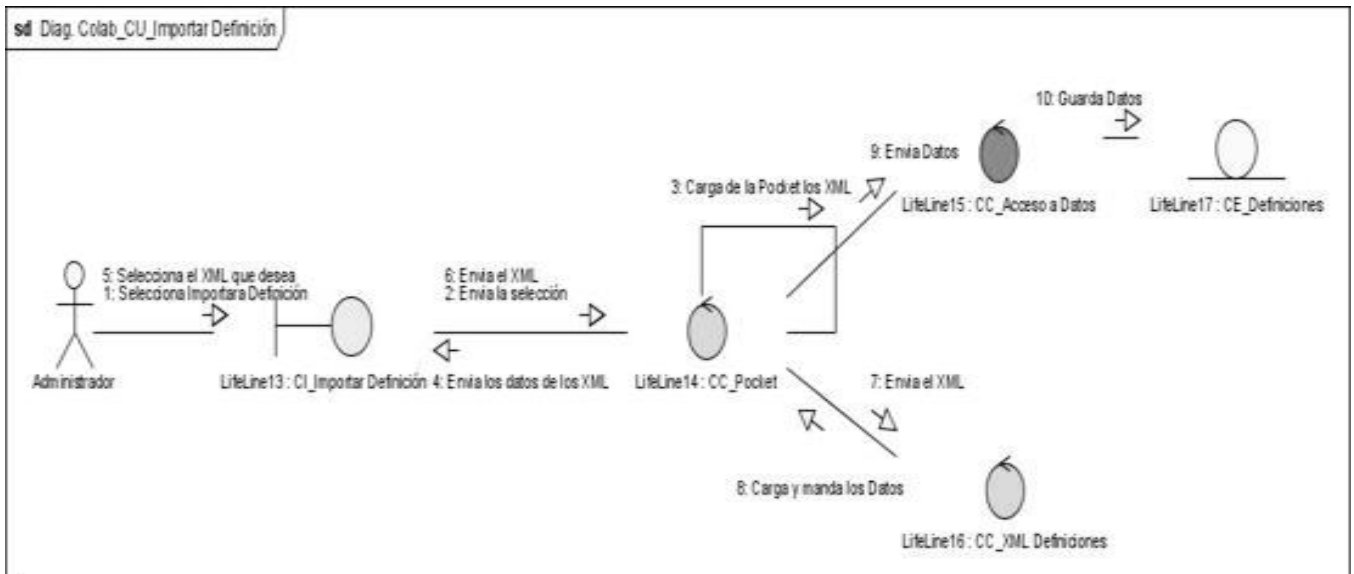


Figura 15: Diagrama de Colaboración CU <Importar Definición Captada>.

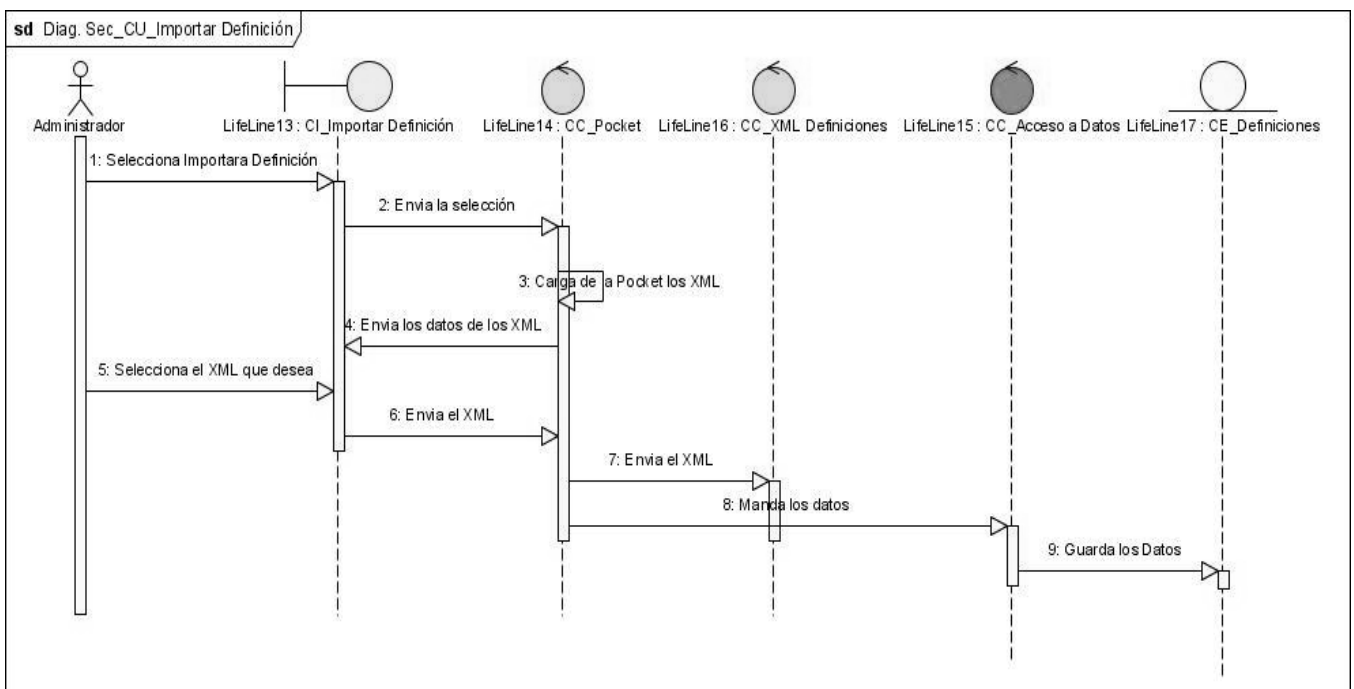


Figura 16: Diagrama de Secuencia CU < Importar Definición Captada>.

ANEXO 5: Modelos de Diseño.

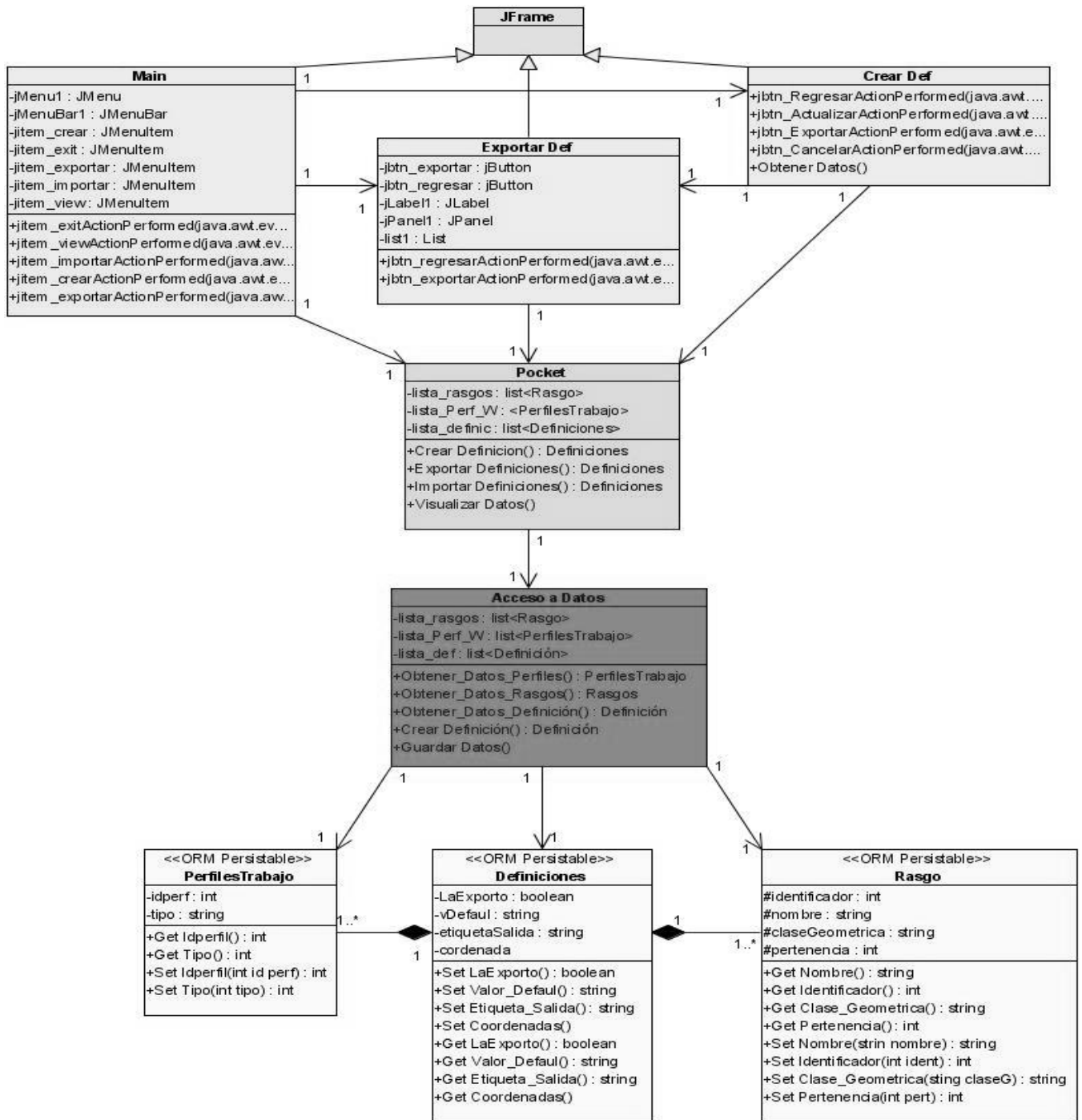


Figura 17: Diagrama de Clases del Diseño CU <Crear Definición>.

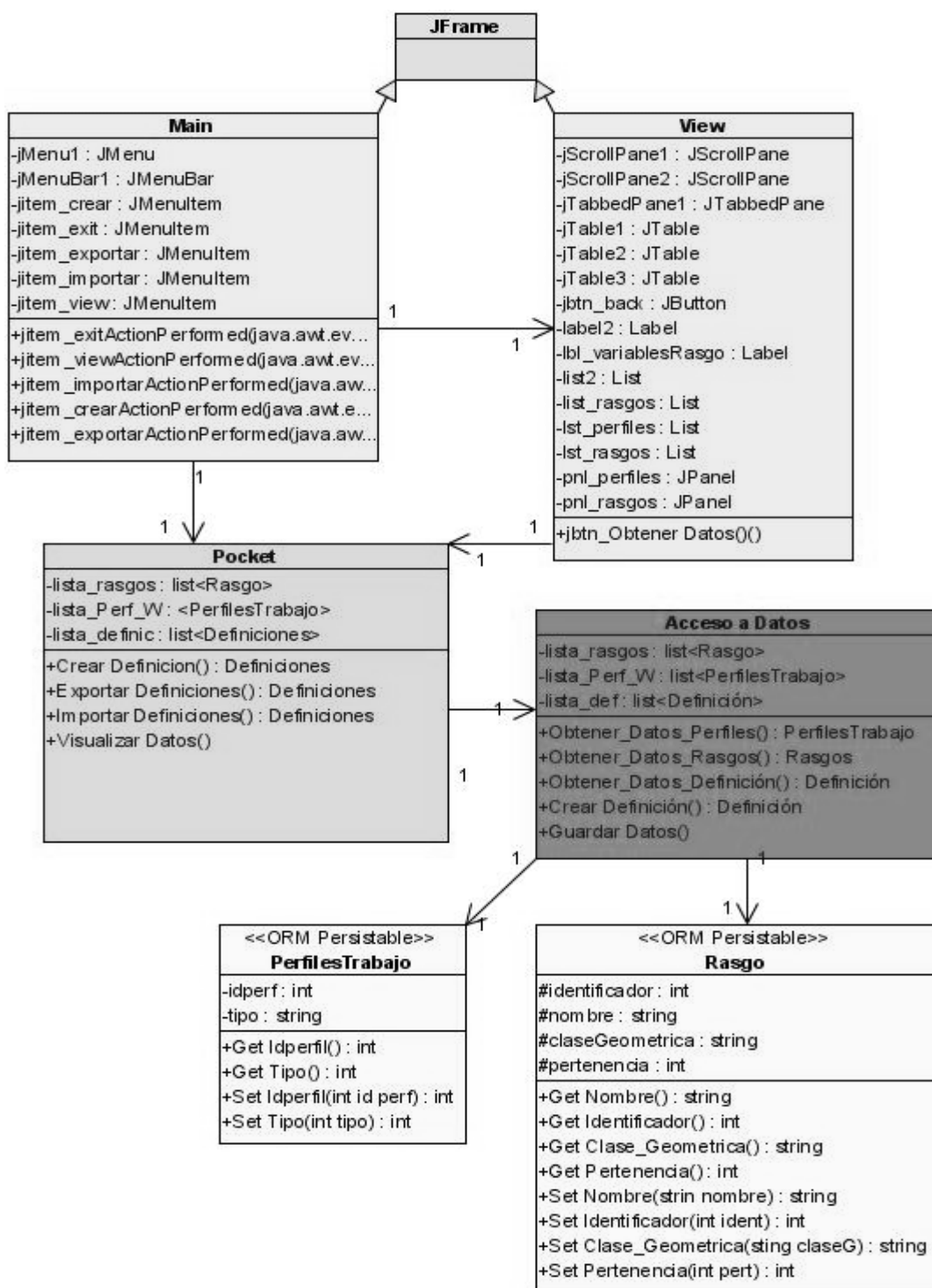


Figura 18: Diagrama de Clases del Diseño CU <Visualizar Datos Existentes>.

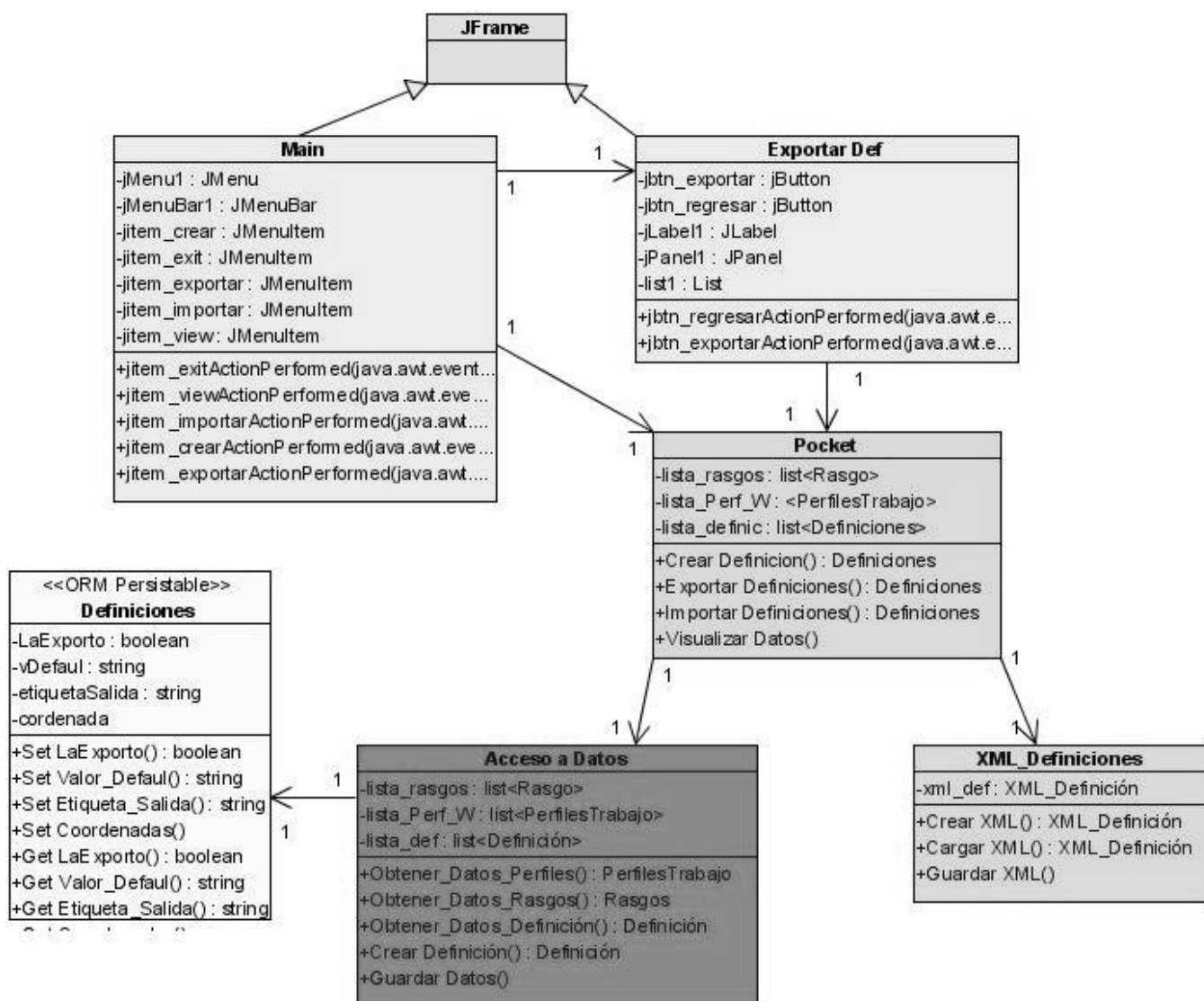


Figura 19: Diagrama de Clases del Diseño CU < Exportar Definición Creada >.

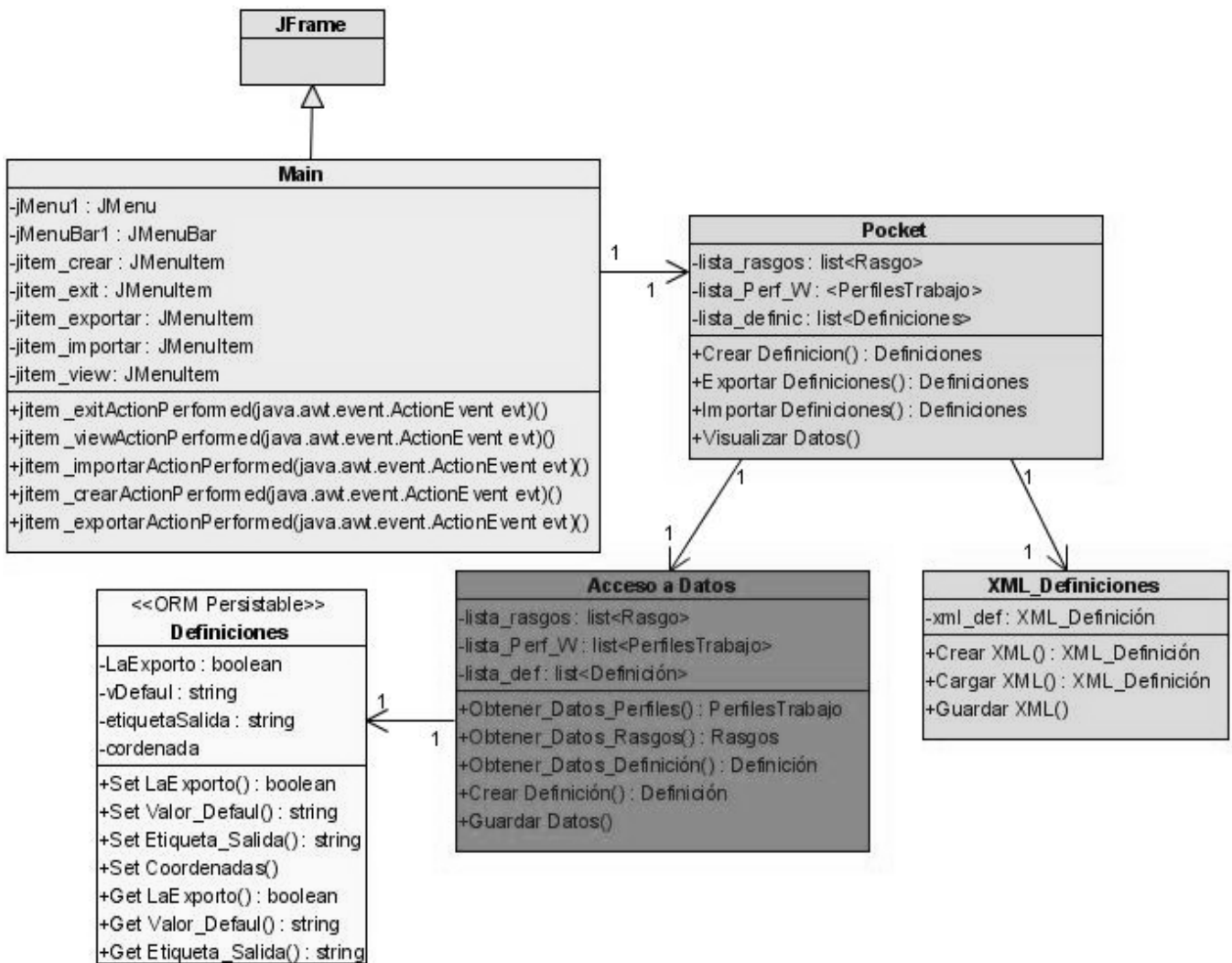


Figura 20: Diagrama de Clases del Diseño CU < Importar Definición Captada >.

ANEXO 6: Descripción de las Clases de Análisis.

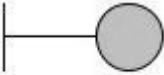


Estereotipo	Nombre	Descripción
 <p>Interfaz</p>	Clase Interfaz	Se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción a menudo implica recibir información y peticiones de los usuarios y los sistemas externos. Modelan las partes del sistema que dependen de sus actores, lo cual implica que clarifican y reúnen los requisitos en los límites del sistema.
 <p>Controladora</p>	Clase Controladora	Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.
 <p>Entidad</p>	Clase Entidad	Se utilizan para modelar información que posee una larga vida y que es a menudo persistente. Las clases entidad modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

Tabla 1: Descripción de las Clases de Análisis.