

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título: Aplicación de Escritorio para la Identificación de
Minerales en Secciones Pulidas

Autor: Roger René Ramírez González
Tutor(es): Dra. C Yiezenia Rosario Ferrer
Ing. Oscar Reyes Pérez
Consultante: Ing. Iván Barea Pérez

Moa, 2014
"Año 56 de la Revolución"

Declaración de Autoría

Declaración de Autoría

Declaro que soy el único autor de la “Aplicación de Escritorio para la Identificación de Minerales en Secciones Pulidas” y autorizo al Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, a que haga uso del mismo como estime conveniente.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2014.

Roger René Ramírez González

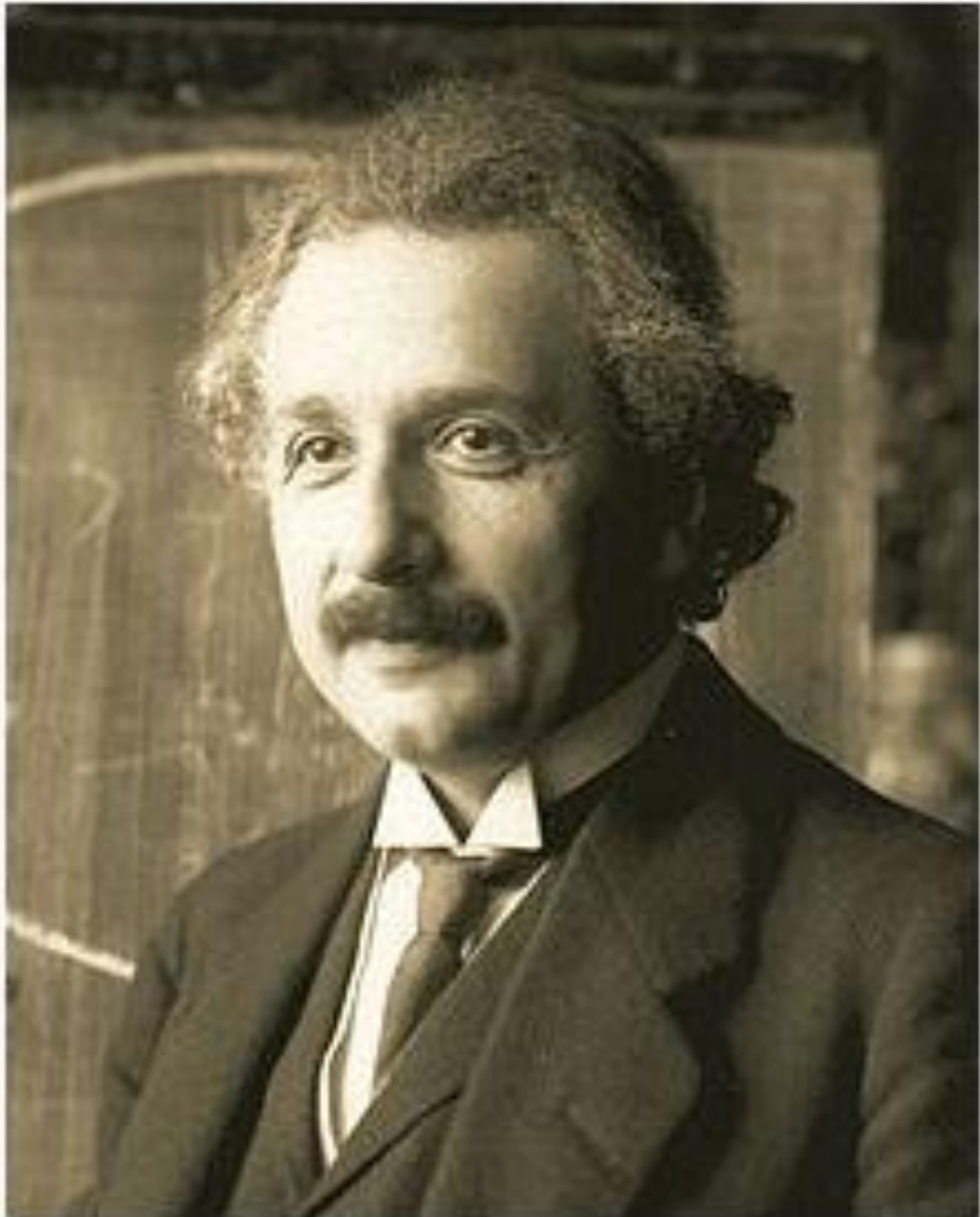
Firma del Autor

Dra. C Yiezenia Rosario Ferrer

Firma de la Tutora

Ing. Oscar Reyes Pérez

Firma del Tutor



La alegría de mirar y comprender es el don más bello de la naturaleza.

Albert Einstein.

Dedicatoria

*Este trabajo va dedicado en especial a mis padres: **Roger Ramírez García** y **Susana Guadalupe González Hernández.***

*A mis abuelos paternos: **Rigoberto** y **Estrella**, maternos: **René** y **Ana Lidia.***

A mis familiares y amigos que me han apoyado de manera incondicional toda la vida.

Roger René.

Agradecimientos

Doy gracias:

*A Dios y a mis padres: **Roger Ramírez García y Susana Guadalupe González Hernández** por crear la persona que hoy soy.*

A mis demás familiares desde oriente hasta occidente y un poquito más por tener esperanza en mí y ayudarme en todo lo que han podido.

*A todos mis compañeros del aula, en especial a **José Luis, Iván, Carlos, Arnel, Roilandy (roco), Yuris, Rafael (lolo), Juan Carlos, Luis Reanne, Eugenio y Marlon.***

*A mis amigos incondicionales **Osvaldito, Idalberto, Karelia, Yosbani, Yisenia y familia** y **a los demás**, que aunque no estén sus nombres escritos saben que le estoy eternamente agradecido.*

*A mis tutores **Yiezenia Rosario Ferrer y Oscar Reyes Pérez**, a mi consultante **Iván Barea Pérez** y todos los demás profesores que me impartieron clases.*

Roger René.

Resumen

En el Laboratorio de Microscopía de Menas del Instituto Superior Minero Metalúrgico de Moa, dentro de los procesos que en él se realizan, se encuentra el de Identificación de Minerales en Secciones Pulidas. Este se realiza de forma manual generando gran pérdida de tiempo tanto a especialistas como a estudiantes de la carrera de geología, por lo que surgió la necesidad de crear un sistema informatizado que permita agilizar dicho proceso.

Con el desarrollo de esta investigación se propone la informatización de la identificación de minerales en secciones pulidas, a partir de la creación de una aplicación de escritorio. Para trazar el desarrollo de la misma se usó la Metodología XP, como lenguaje de programación Java con el IDE NetBeans 1.7.3 y MySQL para el almacenamiento de los datos.

La aplicación permitirá entre otras cosas: agilizar el proceso de identificación de minerales, la identificación de tipos de rocas haciendo uso del diagrama de Streckeisen QAP para rocas Plutonitas y Vulcanitas y generar un reporte con los datos de los minerales existentes en la base de datos.

Abstract

In the Laboratory of Microscopía of Menas of the Higher Institute of Mining and Metallurgies of Moa, within the processes that in him are carried out, meet the Identification of Minerals in Polished Sections. This, are carried out of manual form generating great loss of so much time to specialists like to students of the geology, for this reason the necessity has surged of creating an informatics system that permits to swift this process.

With the development of this investigation proposes the digitalization of the identification of minerals in polished sections, starting from the creation of a desktop application. In order to trace the development of the same, it was used the XP Methodology, how language code, Java with the IDE NetBeans 1. 7. 3 and MySQL for the storage of the facts.

The application will permit between other things: swift the process of identification of minerals, the identification of types of rocks making use of the diagram of Streckeisen QAP for Plutonic rock and Volcanic and generate a report with the facts of the existent minerals in the data base.

Índice

Introducción.....	1
Capítulo1: Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos fundamentales.	5
1.3. Antecedentes.....	8
1.4. Tendencias y Tecnologías actuales.....	8
1.4.1. Software Libre.	8
1.4.2. Lenguaje de Programación.	9
1.4.3. Sistemas Gestores de Bases de Datos.	12
1.5. Metodologías de Desarrollo de Sistemas Informáticos.....	15
1.6. Patrones Arquitectónicos.....	20
1.7. Herramientas a Utilizar.	24
1.8. Conclusiones del Capítulo.	26
Capítulo 2- Planeación y Diseño	27
2.1. Introducción.	27
2.2. Personal relacionado con la aplicación.....	27
2.3. Funcionalidades.....	27
2.3.1. Funcionalidades del Sistema.....	28
2.3.2. Características del Sistema.....	29
2.4. Historias de Usuarios (HU).	30
2.5. Planificación de Entregas.	31
2.6. Estimación de Esfuerzo por Historia de Usuario.....	32
2.6.1. Planificación de Iteraciones.....	32
2.6.2. Plan de duración de Iteraciones.	34
2.7. Tarjetas de Clase-Responsabilidad-Colaboración (CRC).....	34
2.8. Modelo de datos.	36
2.9. Conclusión del Capítulo.....	37
Capítulo 3- Desarrollo y Pruebas	38
3.1. Introducción.	38

3.2. Tareas por Historias de Usuario.	38
3.3. Principales interfaces de la aplicación.	41
3.4. Pruebas.	41
3.4.1 Pruebas de Aceptación.	42
3.3. Conclusiones del capítulo.	43
Capítulo 4- Estudio de Factibilidad.....	44
4.1. Introducción.	44
4.2. Efectos Económicos.	44
4.3. Beneficios y Costos intangibles en el proyecto:	46
4.4. Fichas de Costo:	46
4.5. Conclusiones del capítulo.	49
Conclusiones Generales	50
Recomendaciones.....	51
Referencias Bibliográficas	52
Bibliografía.....	55
Glosario de Términos	57
Anexo 1: <i>Historias de Usuarios</i>	60
Anexo 2: <i>Tarjetas CRC</i>.....	64
Anexo 3: <i>Tarjetas de Ingeniería</i>.	68
Anexo 4: <i>Interfaces</i>	79
Anexo 5: <i>Pruebas de aceptación</i>	83

Índice de Tablas

Tabla 1.1: Diferencias entre las Metodologías Ágiles y las Tradicionales.	16
Tabla 2.1: Personal relacionado con la aplicación.	27
Tabla 2.2: Lista de Funcionalidades del Sistema.	28
Tabla 2.3: Lista de Características del Sistema.	29
Tabla 2.4: Planilla de Historia de Usuario.	30
Tabla 2.5: HU Gestionar usuarios.	31
Tabla 2.14: Estimación de esfuerzo por HU.	32
Tabla 2.15: Duración de iteraciones.	34
Tabla 2.16: Plantilla de tarjeta CRC.	35
Tabla 2.17: Tarjeta CRC: Usuarios.	35
Tabla 3.1: Distribución de tareas por historias de usuario.	38
Tabla 3.2: Historias de usuario abordadas en la primera iteración.	39
Tabla 3.3: Historias de usuario abordadas en la segunda iteración.	39
Tabla 3.4: Historias de usuario abordadas en la tercera iteración.	40
Tabla 3.5: Historias de usuario abordadas en la cuarta iteración.	40
Tabla 3.6: Historias de usuario abordadas en la quinta iteración.	40
Tabla 3.7: Tarea de ingeniería. 1: Insertar Usuario.	40
Tabla 3.30: Planilla de prueba de aceptación.	42
Tabla 3.31: Prueba de aceptación HU 1: Gestionar usuarios.	43
Tabla 4.1: Costo en Moneda Libremente Convertible.	46
Tabla 4.2: Costo en Moneda Nacional.	47
Tabla 2.6: HU Gestionar mineral.	60
Tabla 2.7: HU Listar minerales por tipo de dureza.	60
Tabla 2.8: HU Identificar minerales.	61
Tabla 2.9: HU Visualizar los datos de un mineral.	61
Tabla 2.10: HU Buscar datos de un mineral por nombre.	62
Tabla 2.11: HU Buscar minerales por criterios de búsqueda.	62
Tabla 2.12: HU Generar reporte.	63
Tabla 2.13: HU Identificar tipos de rocas ígneas.	63

Índice de Tabla

Tabla 2.18: Tarjeta CRC: Mineral.....	64
Tabla 2.19: Tarjeta CRC: Imágenes.....	66
Tabla 2.20: Tarjeta CRC: Conexión.	66
Tabla 2.21: Tarjeta CRC: Sistemas.....	67
Tabla 2.22: Tarjeta CRC: Fórmula química.....	67
Tabla 3.8: Tarea de ingeniería No. 2: Listar usuarios.....	68
Tabla 3.9: Tarea de ingeniería No. 3: Modificar usuarios.....	68
Tabla 3.10: Tarea de ingeniería No. 4: Eliminar usuarios.	69
Tabla 3.11: Tarea de ingeniería No. 5: Insertar mineral.	69
Tabla 3.12: Tarea de ingeniería No. 6: Generador de fórmula.	70
Tabla 3.13: Tarea de ingeniería No. 7: Modificar mineral.....	70
Tabla 3.14: Tarea de ingeniería No. 8: Eliminar mineral.	71
Tabla 3.15: Tarea de ingeniería No. 9: Listar por dureza baja.	71
Tabla 3.16: Tarea de ingeniería No. 10: Listar por dureza media.	72
Tabla 3.17: Tarea de ingeniería No. 11: Listar por dureza Alta.	72
Tabla 3.18: Tarea de ingeniería No. 12: Capturar datos y realizar búsqueda.	73
Tabla 3.19: Tarea de ingeniería No. 13: Listar resultados de la identificación.	73
Tabla 3.20: Tarea de ingeniería No. 14: Limpiar datos y resultados.	74
Tabla 3.21: Tarea de ingeniería No. 15: Visualizar datos de un mineral.	74
Tabla 3.22: Tarea de ingeniería No. 16: Capturar nombre y mostrar datos del mineral.	75
Tabla 3.23: Tarea de ingeniería No. 17: Capturar datos y realizar búsqueda.	75
Tabla 3.24: Tarea de ingeniería No. 18: Listar resultado de la búsqueda.	76
Tabla 3.25: Tarea de ingeniería No. 19: Limpiar lista de resultados y datos.....	76
Tabla 3.26: Tarea de ingeniería No. 20: Generar reporte	77
Tabla 3.27: Tarea de ingeniería No. 21: Mostrar diagramas.	77
Tabla 3.28: Tarea de ingeniería No. 22: Graficar y mostrar resultados.....	78
Tabla 3.29: Tarea de ingeniería No. 23: Exportar los datos de minerales.....	78
Tabla 3.32: Prueba de aceptación HU 1: Gestionar usuarios.....	83
Tabla 3.33: Prueba de aceptación HU 2: Gestionar Mineral.	84
Tabla 3.34: Prueba de aceptación HU 2: Gestionar Mineral.	84
Tabla 3.35: Prueba de aceptación HU 2: Gestionar Mineral.	85

Índice de Tabla

Tabla 3.36: Prueba de aceptación HU 2: Gestionar Mineral.	85
Tabla 3.37: Prueba de aceptación HU 3: Listar minerales por tipo de dureza.	86
Tabla 3.38: Prueba de aceptación HU 4: Identificar minarles.....	86
Tabla 3.39: Prueba de aceptación HU 5: Visualizar información de un mineral.	87
Tabla 3.40: Prueba de Aceptación HU 6: Buscar datos de un mineral por nombre.	87
Tabla 3.41: Prueba de aceptación HU 7: Buscar minerales por criterios de búsqueda..	88
Tabla 3.42: Prueba de aceptación HU 8: Generar reporte.	88
Tabla 3.43: Prueba de aceptación HU 9: Identificación de tipo de rocas ígneas.....	89

Índice de Figuras

Figura 1.1: Diagrama QAP para rocas plutonitas.	7
Figura 1.2: Diagrama QAP para rocas vulcanitas.	7
Figura 1.3: Arquitectura - Modelo - Vista - Controlador	21
Figura 1.4: Arquitectura en Capa.	23
Figura 3.1: Interfaz principal.	41
Figura 4.1: Gráfica de comparación de la solución manual y la informatizada.	48
Figura 3.2: Diagrama QAP para rocas plutonitas.	79
Figura 3.3: Confirmación de Administración.	79
Figura 3.4: Gestionar Usuarios.	80
Figura 3.5: Ejemplo de identificación de tipo de roca vulcanita.	80
Figura 3.6: Búsqueda por Criterios.	81
Figura 3.7: Generador de Fórmulas.	81
Figura 3.8: Gestionar Minerales.	82
Figura 3.10: Modificar Mineral.	82



Introducción.

El mundo de hoy se caracteriza por el desarrollo acelerado de las tecnologías. Ningún centro, empresa o negocio podrá tener éxito si no es capaz de adaptarse a los diferentes cambios. En tal sentido se aprovechan los beneficios de esta nueva era, en la que la informática rige cada vez más el proceso de desarrollo y evolución del ser humano. Además, permite automatizar cada uno de los procesos donde el hombre se ve involucrado, simplificándole con esto, disímiles gastos económicos y sociales.

El avance tecnológico en informática ha mejorado y sigue desarrollándose aún más desde su aparición. Ha llegado incluso a que los software informáticos aumenten sus prestaciones, su accesibilidad de forma vertiginosa y que cada día sea mayor la cantidad herramientas informáticas.

Actualmente nuestro país no se encuentra ajeno en el uso de la informática y con ello el desarrollo del producto del software. Este ha tomado mayor auge producto al aumento del software libre que han facilitado la creación de aplicaciones para darle solución a disímiles problemas relacionados con diversas ramas de las ciencias como la Petrografía: rama de la geología que se ocupa del estudio e investigación de las rocas, en especial en cuanto respecta a su aspecto descriptivo, su composición mineralógica y su estructura [1].

Teniendo en cuenta las ventajas de las Tecnología de la Informática y las Comunicaciones (TIC), el Instituto Superior Minero Metalúrgico de Moa (ISMM) principal institución universitaria en el país en la enseñanza de la geología desde su fundación en 1976, se ha propuesto informatizar diversos procesos que se realizan en la rama de la petrografía como es el de Identificación de Minerales en Secciones Pulidas el cual se realiza en el Laboratorio de Microscopía de Menas de dicho Instituto. Este proceso se desarrolla actualmente de forma manual lo que genera gran pérdida de tiempo tanto a los especialistas como a los estudiantes producto a lo engorroso que se vuelve el desarrollo del mismo por la utilización de varios medios (Microscopio, Bibliografía en formato duro, planillas, Micro durómetro, etc.).



Por lo que haciendo un análisis de **la situación problémica** antes expuesta se define como **problema científico**: ¿Cómo agilizar el proceso de identificación de minerales en secciones pulidas en el Instituto Superior Minero Metalúrgico de Moa?

Teniendo en cuenta el problema planteado surge como **objeto de estudio**: la informatización del proceso de identificación de minerales en secciones pulidas. Por lo que se enmarca el **campo de acción**: aplicaciones para la identificación de minerales en secciones pulidas.

Se definió como **objetivo general**: Desarrollar una aplicación de escritorio para la identificación de minerales en secciones pulidas para el Instituto Superior Minero Metalúrgico de Moa y los siguientes **objetivos específicos**:

- ✚ Realizar un análisis teórico y conceptual de los principales elementos que se deben tener en cuenta para el desarrollo del trabajo de diploma.
- ✚ Estudiar las metodologías y herramientas a utilizar para el diseño e implementación de la aplicación.
- ✚ Desarrollar la aplicación para las identificaciones de minerales en secciones pulidas.
- ✚ Realizar el estudio de la factibilidad y la valoración de la sostenibilidad del producto.

Para dar cumplimiento al objetivo general se planteó el **sistema de tareas** siguiente:

- ✚ Estudio del estado del arte de las aplicaciones vinculadas al objeto de estudio.
- ✚ Selección del lenguaje de programación y las herramientas para la implementación de la aplicación
- ✚ Selección de la metodología de desarrollo de software que se utilizará.
- ✚ Realizar la ingeniería de software teniendo en cuenta la metodología seleccionada.
- ✚ Elaborar el diseño de la base de datos para la aplicación.
- ✚ Realizar la implementación de las interfaces y funciones de la aplicación de acuerdo a los requisitos planteados por el cliente.
- ✚ Realización de pruebas a la aplicación desarrollada.



- ✚ Realizar el estudio de factibilidad haciendo uso de la metodología Costo Beneficio.
- ✚ Realizar Manual de Usuario de la aplicación.

El trabajo de diploma se sustenta en la **idea a defender**: Si se crea una aplicación de escritorio para la identificación de minerales en secciones pulidas para el Laboratorio de Microscopía de Menas del ISMM, se facilitará el proceso de identificación de minerales, por lo que se optimizará el tiempo que se invierte para la realización de dicho proceso actuando de forma positiva en el proceso docente educativo.

La investigación para la realización de este trabajo de diploma se realizó basada en los **métodos** que a continuación se describen:

Como **Métodos Teóricos** se utilizaron:

- ✚ Histórico – Lógico: para la búsqueda de antecedentes del software, las herramientas utilizadas y la forma en que se realizaba el proceso de identificación de minerales en secciones pulidas en el (ISMMM).
- ✚ Análisis y Síntesis: permite alcanzar una profundización en el conocimiento del problema en su totalidad, descomponiéndolo en partes para sintetizar su estudio y precisar sus múltiples relaciones y comportamientos.

Como **Métodos Empíricos** se utilizaron:

- ✚ Entrevistas: para determinar los requisitos funcionales de la aplicación que se quiere construir. Se llevaron a cabo varios diálogos con personas expertas en la materia.
- ✚ Análisis de documento: para elaborar los fundamentos teóricos que se relacionan con el campo de acción.

El presente Trabajo de Diploma está conformado por:

Capítulo 1. Fundamentación Teórica: Este capítulo incluye un estudio del estado del arte del tema tratado, a nivel nacional e internacional, además se analizarán los diferentes lenguajes de programación, los sistemas gestores de bases de datos y



metodologías de desarrollo de software, para escoger cuáles se utilizarán para el desarrollo de la aplicación propuesta.

Capítulo 2. Planeación y Diseño Este capítulo se hace uso de la metodología seleccionada para el desarrollo del proyecto abordando sus dos primeras fases. Fueron llenadas las historias de usuarios por el cliente, se desarrollaron las tarjetas CRC y el Modelo de Datos.

Capítulo 3. Desarrollo y Pruebas: En este capítulo se definen las tareas por historias de usuarios, se realizan las tarjetas de ingeniería, las cuales luego son implementadas. También son mostradas algunas de las interfaces principales así como las pruebas de aceptación realizadas con sus resultados.

Capítulo 4. Estudio de Factibilidad: En este capítulo se realiza un estudio de la factibilidad del producto mediante el uso de la metodología Costo Beneficio. Además de un estudio de los esfuerzos requeridos para la realización de la aplicación.



Capítulo 1: Fundamentación Teórica

1

1.1. Introducción

En este capítulo se realiza un proceso investigativo de los aspectos teóricos necesarios para la elaboración y concepción del Trabajo de Diploma, las características generales de los lenguajes de programación y de los sistemas de bases de datos y como propuesta para la futura implementación. Además se verán las diferentes metodologías existentes para el desarrollo de aplicaciones, y de ellas la adecuada a utilizar en este trabajo basando su selección en las ventajas que esta ofrece. También se hace una descripción de las herramientas y tecnologías utilizadas para el análisis y diseño del software.

1.2. Conceptos fundamentales.

La **petrografía** es la rama de la geología que se ocupa del estudio e investigación de las rocas, en especial en cuanto respecta a su aspecto descriptivo, su composición mineralógica y su estructura.

Se complementa así con la **petrología**, disciplina que se centra principalmente en la naturaleza y origen de las rocas. Consiste en el estudio de las propiedades físicas, químicas, mineralógicas, espaciales y cronológicas de las asociaciones rocosas y de los procesos responsables de su formación.

Se encarga de tres tipos de rocas específicamente. La primera y más abundante de todas se basa en el estudio de las rocas **ígneas** que deben su origen al enfriamiento lento del magma en el interior de la Tierra (*rocas ígneas intrusivas*) o a la lava expulsada por los volcanes (*rocas ígneas extrusivas*). El segundo tipo son las rocas **sedimentarias** que se originan por la erosión, desgaste de las rocas por el viento, agua o hielo. El tercer tipo son las rocas **metamórficas** que se forman cuando los tipos anteriores se ven sometidos a elevadas presiones y temperaturas en el interior de la Tierra. [1]



Capítulo 1- Fundamentación Teórica

El **Mineral**: es un sólido homogéneo inorgánico de origen natural, con una composición química definida, que puede ser variable dentro de unos límites, y que presenta una estructura cristalina.

Ser **sólido** implica que los átomos que forman una sustancia prácticamente no pueden moverse con libertad unos respecto a otros. En los líquidos, la movilidad de los átomos o moléculas es mayor, y en los gases, mayor todavía.

La característica **homogénea** hace referencia a que los componentes químicos de que consta un mineral no pueden ser separados por procesos físicos.

El hecho de ser **inorgánico**, permite diferenciar algunos minerales de otras sustancias que presentan la misma composición, y adoptan formas prácticamente idénticas, pero que son originados por seres vivos.

El origen natural diferencia los minerales de aquellas sustancias muy semejantes a ellos, pero producidas por acción de las personas. Según esto, los diamantes sintéticos, no se pueden considerar minerales.

Los minerales presentan **composiciones químicas definidas**. Esto no explica que, en ocasiones, presenten otros elementos químicos en proporciones menores. Las coloraciones variables de muchos cristales de cuarzo se deben a este hecho.

Estructura cristalina: Implica una disposición ordenada de los átomos, es decir, los átomos forman una red tridimensional que sigue un modelo geométrico regular. Los cuerpos que presentan estructura cristalina, se llaman cristalinos. Si presentan una forma más o menos geométrica, en su exterior se llaman cristales. [2]

El **diagrama de Streckeisen** o “QAPF” es actualmente el más utilizado en la identificación de rocas ígneas. Permite en una manera bastante fácil la denominación de rocas plutónicas y volcánicas. Solamente el contenido modal de 4 minerales en una muestra y la textura definen al final el nombre de la roca. Existen pocas excepciones: nombres como “ignimbrita” o piedra pómez no tienen su origen en este diagrama. [3]



Capítulo 1- Fundamentación Teórica

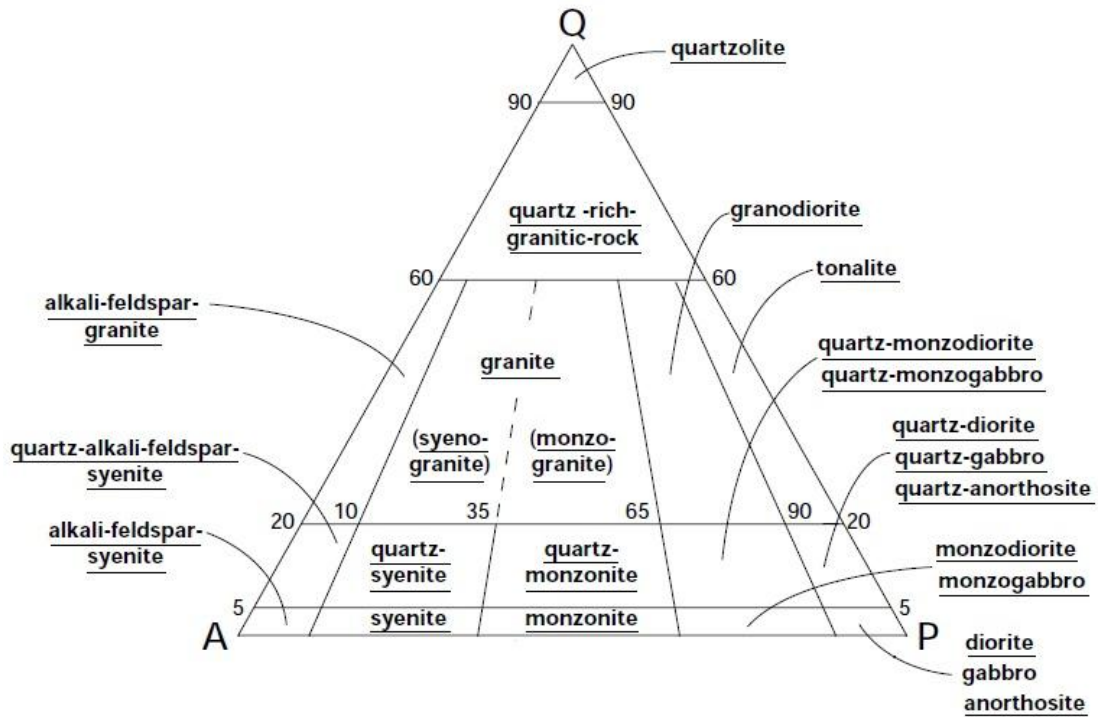


Figura 1.1: Diagrama QAP para rocas plutonitas.

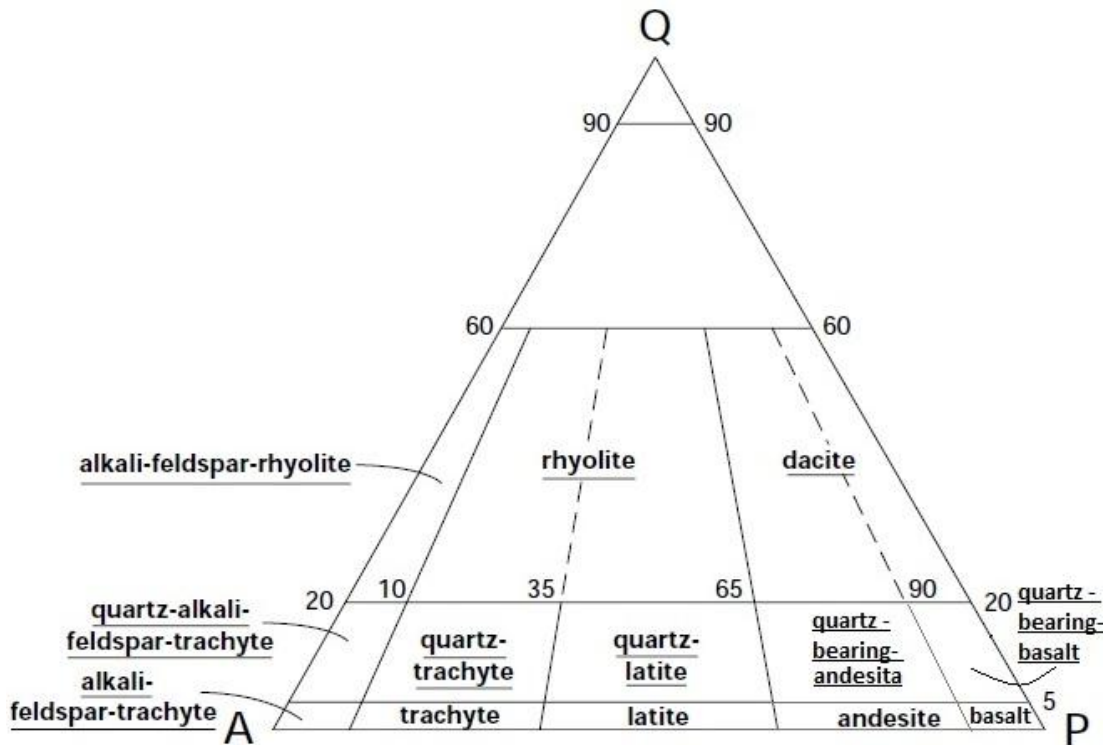


Figura 1.2: Diagrama QAP para rocas vulcanitas.



1.3. Antecedentes.

Hasta el momento de la elaboración de este trabajo de diploma no existe constancia de la existencia de una aplicación que se haya creado para la identificación de minerales en secciones pulidas tanto a nivel nacional como internacional.

Aunque dentro de la búsqueda realizada se encontró una aplicación creada en java conocida como: jVPM - Java Virtual Petrographic Microscope (*Microscopio Petrográfico virtual*) que está especialmente diseñada para ayudar a los mineralogistas en sus esfuerzos de investigación. Les permite analizar secciones delgadas y observar sus propiedades mucho más fácil.

La aplicación se dirige principalmente a los estudiantes de geología, lo que les permite examinar los minerales y las rocas en un entorno fácil de usar sin necesidad de un microscopio. [4]

¿Por qué no utilizarlo?

El software antes mencionado no se ajusta totalmente a las necesidades que se quieren resolver de forma objetiva y factible al problema existente en el laboratorio de microscopía de menas del ISMM. Este software está diseñado para el análisis de minerales en secciones delgadas y no para secciones pulidas, además necesita al menos 14 Mb/s de conexión a Internet, algo con lo que no cuenta la de red del Instituto actualmente.

1.4. Tendencias y Tecnologías actuales.

1.4.1. Software Libre.

Se denomina Software Libre a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software: [5]

- ✚ La libertad de usar el programa, con cualquier propósito.
- ✚ La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades. El acceso al código fuente es una condición previa para esto.



- ✚ La libertad de distribuir copias, con lo que puedes ayudar a tu vecino.
- ✚ La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Las ventajas especialmente económicas que aportan las soluciones libres y las aportaciones de la comunidad de desarrollo han permitido un constante crecimiento del software libre hasta superar en ocasiones al mercado propietario. Estas ventajas hacen que el país siga una política de migración hacia el software libre y como parte de este proceso se decide para el desarrollo de la aplicación la utilización de herramientas y tecnologías pertenecientes al software libre.

1.4.2. Lenguaje de Programación.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación [6].

- ***C++***

Es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. El nombre C++ fue propuesto por Rick Masciatti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases".

Se puede decir que abarca tres paradigmas de la programación:

- ✚ La programación estructurada.
- ✚ La programación genérica.
- ✚ La programación orientada a objetos.



Capítulo 1- Fundamentación Teórica

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- ✚ Posibilidad de redefinir los operadores (sobrecarga de operadores)
- ✚ Identificación de tipos en tiempo de ejecución

Está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo es a su vez uno de los que menos automatismos posee (obliga a hacerlo casi todo manualmente al igual que C) lo que "dificulta" mucho su aprendizaje. [7]

- **Java.**

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc. con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma.

La sintaxis del lenguaje heredó características de C y C++, adoptando una muy similar a la del C++. Actualmente, dentro de los lenguajes populares es uno de los mejores en cuanto a definición, debido a que goza de total independencia del implementador del lenguaje y de sus clases auxiliares. Proporciona los tipos de datos primitivos similares a los de C++, proporciona todas las estructuras contenedoras "clásicas". Tiene cuatro niveles de empaquetamiento: variables y funciones, al igual que los lenguajes anteriores y otros dos propios de él, denominados: clases y paquetes. [8].

Características que ofrece Java:

Simple: Se lo conoce como lenguaje simple porque viene de la misma estructura de C y C++; ya que C++ fue un referente para la creación de java por eso utiliza determinadas características de C++ y se han eliminado otras.

Orientado a objetos: Toda la programación en java en su mayoría está orientada a objeto, ya que al estar agrupados en estructuras encapsuladas es más fácil su manipulación.



Capítulo 1- Fundamentación Teórica

Distribuido: Permite abrir, establecer y aceptar conexiones con los servidores o clientes remotos; facilita la creación de aplicaciones distribuidas ya que proporciona una colección de clases para aplicaciones en red.

Robusto: Es altamente fiable en comparación con C, se han eliminado muchas características con la aritmética de punteros, proporciona numerosas comprobaciones en compilación y en tiempo de ejecución.

Seguro: La seguridad es una característica muy importante en java ya que se han implementado barreras de seguridad en el lenguaje y en el sistema de ejecución de tiempo real.

Multiplataforma: Compatible con los más variados sistemas operativos.

Portable: Por ser indiferente a la arquitectura sobre la cual está trabajando, esto hace que su portabilidad sea muy eficiente, sus programas son iguales en cualquiera de las plataformas, ya que java especifica tamaños básicos, esto se conoce como la máquina virtual de java.

Interpretado y compilado a la vez: Java puede ser compilado e interpretado en tiempo real, ya que cuando se construye el código fuente este se transforma en una especie de código de máquina.

Multihebra o Multihilos: Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que el programa tenga se ejecutarán en tiempo real muchas funciones al mismo tiempo.

Dinámico: El lenguaje java es muy dinámico en la fase de enlazado, sus clases solamente actuarán en medida en que sean requeridas, o necesitadas con esto permitirá que los enlaces se puedan incluir incluso desde fuentes muy variadas o desde la red.

Alto rendimiento: Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código. [9]



1.4.3. Sistemas Gestores de Bases de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) o *DBMA (Data Base Management System)*. Es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. [10]

Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server.

Un SGBD debe permitir:

- ✚ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- ✚ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- ✚ Manipular la base de datos: realizar consultas, actualizarla, generar informes.

- **Las características de un SGBD son:**

Abstracción de la información: Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

Independencia: La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Redundancia mínima: Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.



Capítulo 1- Fundamentación Teórica

Consistencia: En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Seguridad: La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Integridad: Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

Respaldo y recuperación: Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Control de la concurrencia: En la mayoría de entornos lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, o para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

- **PostgreSQL**

Es un sistema gestor de base de datos objeto-relacional, bajo licencia BSD. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del



Capítulo 1- *Fundamentación Teórica*

código fuente en software no libre. Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre. Se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, Mac OS, y Windows. Su documentación está muy bien organizada, pública y libre, y posee comentarios de los propios usuarios. Cuenta con soporte nativo para los lenguajes más populares del medio: Java, PHP, C, C++, Perl y Python, Incluye todas las características de un gestor de datos profesional (disparadores, procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.). [11]

- **MySQL**

Es un sistema de gestión de bases de datos relacional, creado por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es SQL (*Structured Query Language*) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales. [12]

Características principales.

- ✚ El principal objetivo de MySQL es velocidad y robustez.
- ✚ Soporta gran cantidad de tipos de datos para las columnas.
- ✚ Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- ✚ Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.



- + Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- + Flexible sistema de contraseñas y gestión de usuarios, con muy buen nivel de seguridad en los datos.
- + El servidor soporta mensajes de error en distintas lenguas.

Ventajas de MySQL:

- + Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- + Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- + Facilidad de configuración e instalación.
- + Soporta gran variedad de Sistemas Operativos
- + Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- + Conectividad y seguridad.

1.5. Metodologías de Desarrollo de Sistemas Informáticos.

Las Metodologías de Desarrollo de Software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas. Las metodologías pueden ser Tradicionales o Ágiles.

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes. Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características.



Capítulo 1- Fundamentación Teórica

- **Comparación entre Metodologías.**

Tabla 1.1: Diferencias entre las Metodologías Ágiles y las Tradicionales. [13]

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

- **¿Qué Metodología se utilizará?**

Teniendo en cuenta la comparación entre las metodologías anteriormente explicadas, se decide el uso de una Metodología Ágil para el desarrollo de este proyecto, eligiendo la **eXtreme Programming** más conocida por sus siglas (XP). Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y pequeño equipo, consiste en una programación rápida o



Capítulo 1- Fundamentación Teórica

extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

- ***Características de XP.***

- + Los diseñadores y programadores se comunican efectivamente con el cliente y entre ellos mismos.
- + Los diseños del software se mantienen sencillos y libres de complejidad o pretensiones excesivas.
- + Se obtiene retroalimentación de usuarios y clientes desde el primer día.
- + El software es liberado en entregas frecuentes tan pronto como sea posible.
- + Los cambios se implementan rápidamente tal y como fueron sugeridos.
- + Las metas en, tiempos y costos son reajustadas permanentemente en función del avance real obtenido.
- + Empieza en pequeño y añade funcionalidad con retroalimentación continua
- + El manejo del cambio se convierte en parte sustantiva del proceso.
- + El costo del cambio no depende de la fase o etapa.
- + No introduce funcionalidades antes que sean necesarias.
- + El cliente o el usuario se convierte en miembro del equipo. [14]

- ***Derechos del Cliente:***

- + Decidir que se implementa.
- + Saber el estado real y el progreso del proyecto.
- + Añadir, cambiar o quitar requerimientos en cualquier momento.
- + Obtener lo máximo de cada semana de trabajo.
- + Obtener un sistema funcionando cada 3 o 4 meses. [14]

- ***Derechos del Desarrollador:***

- + Decidir cómo se implementan los procesos.
- + Crear el sistema con la mejor calidad posible.
- + Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- + Estimar el esfuerzo para implementar el sistema.
- + Cambiar los requerimientos en base a nuevos descubrimientos. [14]



Capítulo 1- Fundamentación Teórica

Lo fundamental en este tipo de metodología es:

- ✚ La comunicación, entre los usuarios y los desarrolladores.
- ✚ La simplicidad, al desarrollar y codificar los módulos del sistema.
- ✚ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. [15]

Fases de la Metodología (XP):

Fase 1: Planeación

- ✚ Se escriben historias de usuarios, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente de tal forma que creen un test de aceptación para historia de usuarios y permitan hacer una estimación de tiempo de desarrollo del mismo.
- ✚ Se crea un plan de lanzamiento que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto.
- ✚ Se usa como base las historias usuarios, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- ✚ El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuarios a desarrollar y las tareas de desarrollo.
- ✚ Se cambia el proceso cuando sea necesario, para adaptarlo al proyecto.

Fase 2: Diseño.

- ✚ Se eligen los diseños funcionales más simples.
- ✚ Se elige una metáfora del sistema para que el nombrado de clases siga una misma línea, facilitando la reutilización y la comprensión de código.



Capítulo 1- *Fundamentación Teórica*

- ✚ Se escriben tarjetas de clases – responsabilidades - colaboración (CRC) para cada objeto, que permita abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

Fase 3: Codificación.

- ✚ El cliente está siempre disponible, de ser posible cara a cara. La idea es que forme parte del equipo de desarrollo y esté presente en todas las fases de XP. Es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente.
- ✚ El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilidad de comprensión y refactorización del mismo.
- ✚ Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida y que cuando se afronte esta misma se tenga más claro qué objetivos tiene que cumplir lo que se va a codificar.
- ✚ La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento solo habrá una pareja de programadores que integre el código.
- ✚ Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo. De esta manera se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- ✚ Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.

Fase 4: Pruebas.

- ✚ Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.



- ✚ Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo error.
- ✚ Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generalmente a partir de las historias de usuarios elegidas para la iteración y son pruebas de caja negra, en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado.

1.6. Patrones Arquitectónicos.

Los Patrones Arquitectónicos, o Patrones de Arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones.

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software. Muchas arquitecturas diferentes pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa "estrictamente descrita y comúnmente disponible". Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interactuar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón.

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a



problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. [16]

- **Patrón de Arquitectura MVC (Modelo Vista Controlador)**

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón arquitectónico Modelo-Vista-Controlador. Si realizamos un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar la interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. [17]

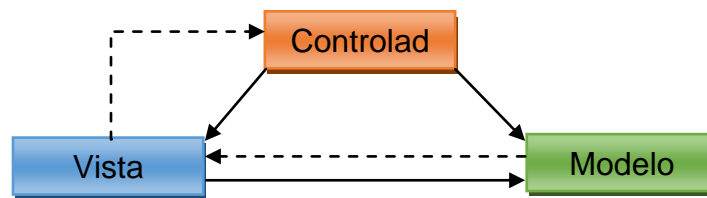


Figura 1.3: Arquitectura - Modelo - Vista - Controlador

Elementos del patrón:

Modelo: Datos y reglas de negocio.

Vista: Muestra la información del modelo al usuario.

Controlador: Gestiona las entradas del usuario.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de pastel, de barras, tabular, etc.

El **modelo** es el responsable de:

- ✚ Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.



Capítulo 1- Fundamentación Teórica

- + Definir las reglas de negocio (la funcionalidad del sistema).
- + Llevar un registro de las vistas y controladores del sistema.

El **controlador** es responsable de:

- + Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- + Contiene reglas de gestión de eventos, del tipo "*Si Evento Z, entonces Acción W*".

Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar. (Una petición al modelo puede ser "*Obtener tiempo de entrega dado una nueva orden*").

Las **vistas** son responsables de:

- + Recibir datos del modelo y mostrarlo al usuario.
- + Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- + Pueden dar el servicio de actualizar, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

- ***Arquitectura en Capas.***

Este tipo de patrón arquitectónico define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores.

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. [18]



Capítulo 1- Fundamentación Teórica

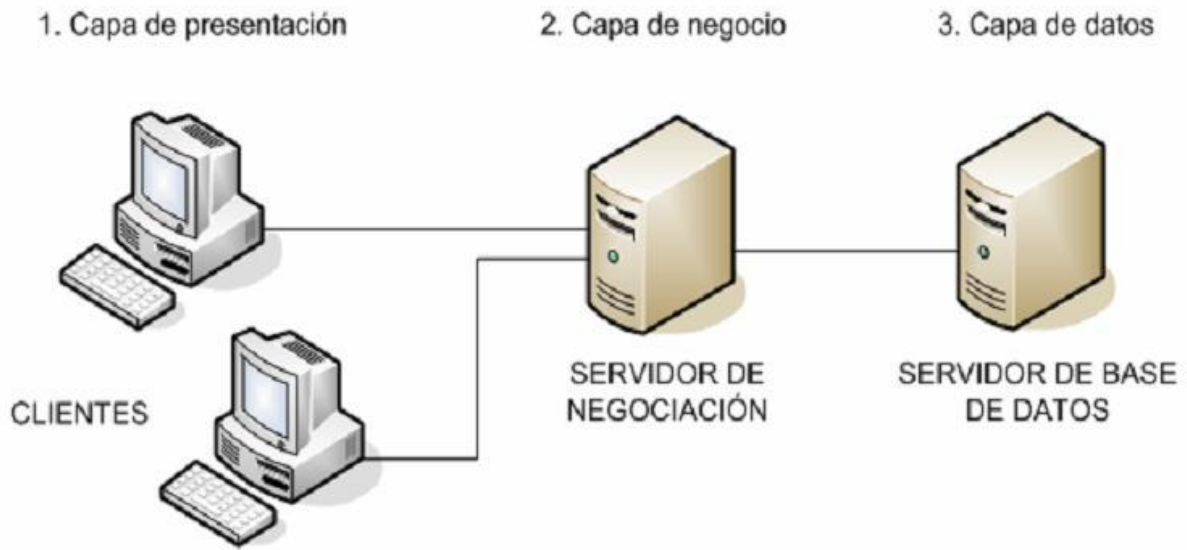


Figura 1.4: Arquitectura en Capa.

Capa de presentación: es la que ve el usuario (también se la denomina "*capa de usuario*"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.

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) porque 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 almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el



Capítulo 1- Fundamentación Teórica

almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Todas estas capas pueden residir en un único ordenador (*no es lo típico*), si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación. Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

1.7. Herramientas a Utilizar.





- **NetBeans**

El IDE NetBeans es un entorno de desarrollo integrado, 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 extenderlo. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. [19]

¿Por qué utilizarlo?

- 1) El desarrollador posee cierto dominio en el uso del IDE.
- 2) Permite la creación de gráficos y dibujos 2D lo cual es necesario para el desarrollo del proyecto.
- 3) Permite ampliarle sus funcionalidades. Ejemplo: Se le agregaran los plugins y librerías de iReport y Jasperreport que a continuación se lista para generar reportes:

Plugin

-  iReport-3.7.0.nbm
-  jasperreports-components-plugin-3.7.0.nbm
-  jasperreports-extensions-plugin-3.7.0.nbm
-  jasperserver-plugin-3.7.0.nbm

Librerías

-  com-jaspersoft-ireport



- + commons-beanutils-1.8.0
- + commons-collections-3.2.1
- + commons-digester-1.7
- + commons-logging-1.1
- + groovy-all-1.5.5
- + iText-2.1.0
- + jasperreports-3.7.0

- ***Embarcadero ER/Studio 8.0***

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos. [20]

¿Por qué utilizarlo?

- + Capacidad fuerte en el diseño lógico.
- + Sincronización bidireccional de los diseños lógico y físico.
- + Construcción automática de Base de Datos.
- + Reingeniería inversa de Base de Datos.

- ***SQL Manager 2005 para MySQL.***

Es una herramienta poderosa para la administración y desarrollo de bases de datos, funciona con cualquier versión de MySQL desde la 3.23 a la 5.02 y soporta las últimas características de MySQL, incluyendo: vistas, procedimientos almacenados, funciones y llaves foráneas. SQL Manager para MySQL permite crear y editar todos los objetos de base de datos MySQL, ejecutar scripts de SQL, importar y exportar datos de bases de datos, gestión de usuarios y sus privilegios y tiene muchas otras características útiles para la eficiente administración de MySQL.



1.8. Conclusiones del Capítulo.

En este capítulo se abordaron conceptos fundamentales asociados a la rama de la geología que se ocupa del estudio e investigación de las rocas. Además se realizó un estudio el cual consistió en describir las potencialidades de los lenguajes de programación, sistemas de base de datos y metodologías de desarrollo escogiendo así los que presentan mayor ventaja con respecto a las características de nuestra aplicación.

Entre lo seleccionado se encuentra JAVA como lenguaje de programación, NetBeans como entorno de desarrollo, XP como metodología para guiar el proceso de obtención del producto. Además para una sencilla administración y navegación de las bases de datos se seleccionó SQL Manager para MySQL.



2.1. Introducción.

En este capítulo se realiza la planeación y diseño en la cual se determina y se detallan a fondo las necesidades del cliente. Se plantean: el personal relacionado con la aplicación, los requisitos funcionales y no funcionales de la aplicación propuesta, los cuales serán objeto de automatización mediante el empleo de historias de usuarios. Para lo cual será necesaria una estimación del esfuerzo en la realización de las mismas y el establecimiento del plan de iteraciones del sistema y su duración, el plan de entrega del proyecto, la creación de las tarjetas Clases-Responsabilidades-Colaboración conocidas por sus siglas (CRC) y el modelo de datos.

2.2. Personal relacionado con la aplicación.

Tabla 2.1: Personal relacionado con la aplicación.

Personal relacionado con la aplicación:	Justificación:
Especialista	Persona que tiene conocimiento en la materia y está encargada de la gestión de la información de la misma.
Desarrollador	Persona responsable de realizar la implementación de sistema.
Usuario	Persona que consulta la información del sistema.
Administrador	Persona responsable de administrar y controlar la aplicación.

2.3. Funcionalidades.

Una vez descrita las personas relacionadas con la aplicación se procede a realizar el análisis de las funcionalidades que debe cumplir la misma, las cuales se listan a continuación.



2.3.1. Funcionalidades del Sistema.

Está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración.

Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto. [21]

Tabla 2.2: Lista de Funcionalidades del Sistema.

Funcionalidades del Sistema.

Gestionar usuarios.

✚ Insertar usuarios.

✚ Listar usuarios.

✚ Modificar usuarios

✚ Eliminar usuarios.

Gestionar Mineral.

✚ Insertar mineral.

✚ Modificar mineral.

✚ Eliminar mineral.

Listar mineral por tipo de dureza.

Identificar minerales.

Visualizar la información de un mineral.

Buscar datos de un mineral por nombre.

Buscar minerales por criterios de búsqueda.

Generar reporte.

Identificar tipos de rocas ígneas.



2.3.2. Características del Sistema.

Tabla 2.3: Lista de Características del Sistema.

Usabilidad

Facilidad de uso por parte de los usuarios: La aplicación debe presentar una interfaz amigable que permita la fácil interacción con la misma y llegar de manera rápida y efectiva a la información buscada, además de posibilitar a los usuarios sin experiencia una rápida adaptación.

Especificación de la terminología utilizada: La aplicación debe contener un lenguaje y términos utilizados por los clientes en la rama abordada, con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.

Seguridad

Integridad: Validación de los datos para evitar estados inconsistentes. Garantizar que la información sea editada solo por quién tiene privilegios. Se debe realizar la confirmación sobre acciones irreversibles como eliminaciones.

Seguridad de la bases de datos: la seguridad de la base de datos está a nivel de usuarios autenticados, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.

Requisitos no funcionales de software

Servidor de base de datos MySQL.

La Máquina Virtual JDK6 o superior.

Requisito para la documentación

Manual de usuario: La aplicación debe contar con un manual de usuario.



2.4. Historias de Usuarios (HU).

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. [18]

Para definir las historias de usuarios se emplean la siguiente planilla:

Tabla 2.4: Planilla de Historia de Usuario.

Historia de Usuario	
Número: (Número de la Historia de Usuario)	Usuario: (Usuario entrevistado para obtener la función requerida a automatizar).
Nombre de la Historia: (Nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente).	
Prioridad en el negocio: (Importancia de la historia para el cliente) (Alta / Media / Baja).	Riesgo en el desarrollo: (Dificultad para el programador) (Alto / Medio / Bajo).
Puntos estimados: 1-3	Iteración asignada: (Iteración a la que corresponde).
Programador responsable: (Nombre del programador).	
Descripción: (Se especifican las operaciones por parte del usuario y las respuestas que dará el sistema).	
Observaciones: (Algunas observaciones de interés).	



Capítulo 2- Planeación y Diseño

Seguidamente se muestra una de las historias de usuarios confeccionadas para el desarrollo de la aplicación:

Tabla 2.5: HU Gestionar usuarios.

Historia de Usuario	
Número: 1	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Gestionar Usuarios	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Roger René Ramírez González.	
Descripción: Solo podrán ser gestionados los usuarios por los administradores. Los que podrán insertar, modificar y eliminar a los usuarios existentes, auxiliándose de la lista de usuarios.	
Observaciones: Confirmado con el cliente.	

Para ver el resto de las Historias de Usuarios ir al: [Anexo 1](#)

2.5. Planificación de Entregas.

En esta fase se establece la prioridad de cada HU, y se realiza una estimación del esfuerzo requerido de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Una entrega debe obtenerse en dos o tres meses. La estimación de la implementación de las HU se establece empleando el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros del equipo de desarrollo, trabajan el tiempo planeado sin ningún tipo de distracción, en este punto de estimación que se utiliza para representar la semana ideal, es de 5 días.

Las historias generalmente tienen un valor de 1 a 3 puntos. Se mantiene un registro de la velocidad de desarrollo establecido por puntos de iteración, basados en la suma de



los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.6. Estimación de Esfuerzo por Historia de Usuario.

Para el desarrollo de la aplicación se realizó una estimación para cada Historia de usuario resultando de la siguiente forma:

Tabla 2.14: Estimación de esfuerzo por HU.

Historia de Usuario	Punto por estimación
1. Gestionar usuarios.	1 semana
2. Gestionar Mineral.	3 semanas
3. Listar minerales por tipo de dureza	1 semana
4. Identificar minerales.	1 semana
5. Visualizar la información de un mineral.	0.1 semana
6. Buscar datos de un mineral por nombre.	0.1 semana
7. Buscar minerales por criterios de búsqueda.	3 semanas
8. Generar reporte	2.4 semanas
9. Identificar tipos de rocas ígneas	3 semanas
Total de semanas	15.1 semanas

2.6.1. Planificación de Iteraciones.

A partir de las historias de usuarios antes expuestas y la estimación del esfuerzo propuesto para la ejecución de las mismas, se realiza la planificación de la etapa de implementación del sistema, apoyándose en el tiempo se ha intentado concentrar las funcionalidades relacionadas en una iteración. En este plan se establece cuántas iteraciones, serán necesarias realizar sobre el sistema para su terminación.

En relación con lo antes tratado se decide realizar cinco iteraciones al sistema, las cuales se describen de forma más detallada a continuación:



Primera Iteración.

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuarios 1 y 2 que se consideraron de mayor importancia para el desarrollo de la aplicación ya que de estas depende el desarrollo de las posteriores.

Segunda Iteración.

Esta tiene como finalidad desarrollar las historias de usuarios 3, 4, 5 y 6. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las historias de usuarios y se tendrá la primera versión de prueba, que contará con una interfaz que incorporan todas las funcionalidades antes vistas, esta se presentará al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Tercera iteración

Se desarrollará la historia de usuario 7, la cual permitirá la identificación de minerales por diferentes criterios de búsquedas.

Cuarta iteración

El desarrollo de la historia de usuario 8, permitirá generar un reporte con la lista de minerales el cual el usuario lo podrá guardar con formato .pdf, o mandar a imprimir.

Quinta iteración

Esta iteración tiene como finalidad desarrollar la HU 9. La misma brinda la funcionalidad de identificar tipos de rocas ígneas mediante el uso del diagrama Streckeisen.

La versión que se obtenga de esta iteración en unión con la anterior, le facilitará al cliente una oportunidad para ver si cumple con los requisitos requeridos. De ser negativo el criterio del cliente, se realizarán los cambios pertinentes solicitados por el mismo. De ser positivo, se dará por concluido el proceso de implementación del proyecto.



2.6.2. Plan de duración de Iteraciones.

Como parte del desarrollo de un proyecto utilizando la metodología XP se crea un plan de duración de iteraciones, el cual tiene como objetivo mostrar el tiempo de duración de cada iteración que se realizará, el cual reflejado de la siguiente forma:

Tabla 2.15: Duración de iteraciones.

Iteración	Historias de Usuarios	Duración total
1ra Iteración	Gestionar Usuarios.	4 Semanas
	Gestionar Mineral.	
2da Iteración	Listar minerales por tipo de dureza.	2.2 semanas
	Identificar minerales.	
	Visualizar la información de un mineral.	
	Buscar datos de un mineral por nombre.	
3ra Iteración	Buscar minerales por criterios de búsqueda.	3 Semanas
4ta Iteración	Generar reporte.	2.4 semanas
5ta Iteración	Identificar tipos de rocas ígneas.	3 semanas

2.7. Tarjetas de Clase-Responsabilidad-Colaboración (CRC).

Las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.

Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.



Tabla 2.16: Plantilla de tarjeta CRC.

(Nombre de la tarjeta CRC)	
Descripción:	
Atributos:	
Nombre	Descripción
(Atributos de la tabla)	(Identificador de la tabla)
Responsabilidades:	
Nombre	Colaborador
(Métodos)	(Relaciones entre tablas)

Muestra de una de las tarjetas CRC llenada para el desarrollo del proyecto:

Tabla 2.17: Tarjeta CRC: Usuarios.

Usuario	
Descripción: almacena los datos de los usuarios.	
Atributos:	
Nombre	Descripción
Id_user	Id_user
user	
pass	
admin	
Responsabilidades:	
Nombre	Colaborador
insertarUser()	usuarios, conexión
ListaUsuarios()	usuarios, conexión
modificarUsuario()	usuarios, conexión
eliminarUsuario()	usuarios, conexión
buscarUsuarioBd()	usuarios, conexión



Para ver el resto de las Tarjetas CRC ir al: [Anexo 2](#)

2.8. Modelo de datos.

Es un lenguaje orientado a describir una Base de Datos. Típicamente un modelo de datos permite describir:

- ✚ Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- ✚ Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- ✚ Operaciones de manipulación de los datos: Operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

A continuación se muestra el modelo de datos con el cual trabaja la aplicación:

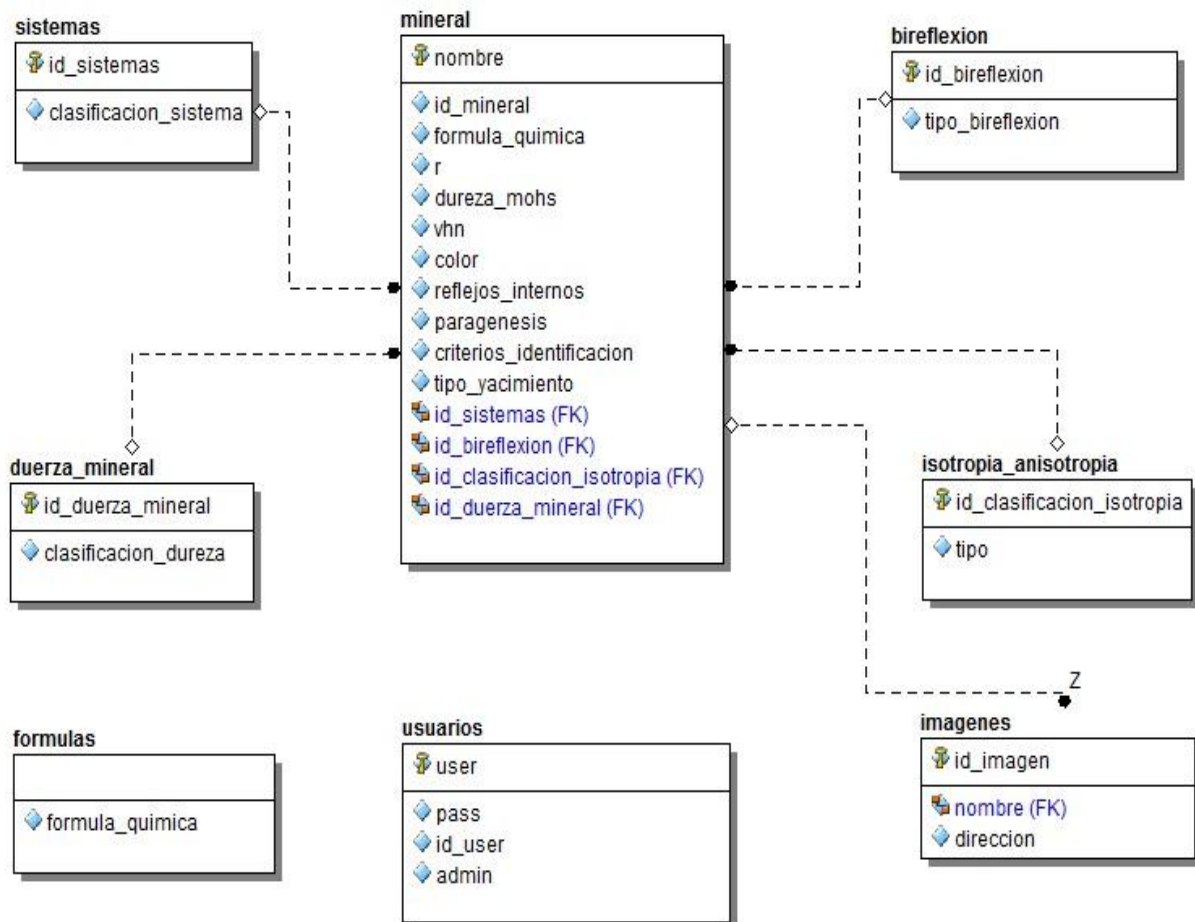


Figura 2.2: Modelo de datos.



2.9. Conclusión del Capítulo.

En este capítulo se abordaron las fases de Planeación y Diseño donde se ha expuesto una de las historias de usuarios elaboradas por el cliente. Se realizó el plan de iteraciones para cada una de estas historias aplicando una estimación de esfuerzo de las mismas. Se presentaron además un ejemplo de las clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC, así como el modelo de datos, terminando de esta forma estas fases para pasar a las siguientes Desarrollo y Pruebas.



3.1. Introducción.

Luego de concluido el capítulo anterior en el que se abordaron las fases de Planeación y Diseño, damos inicio en el presente capítulo conforme a la metodología utilizada a las fases de Desarrollo y Pruebas, en el cual se describen cada una de las tareas confeccionadas para cumplir con el desarrollo de las historias de usuarios definidas. Además se realizan las pruebas de aceptación para comprobar el correcto funcionamiento de la aplicación.

3.2. Tareas por Historias de Usuario.

Siguiendo el desarrollo de la metodología en esta fase las historias de usuario se descomponen en tareas de programación o ingeniería, y a su vez, estas son asignadas al equipo de desarrollo para su implementación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, solo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Las mismas se representan mediante las tarjetas de tareas.

- ***Distribución de tareas por historias de usuario.***

Tabla 3.1: Distribución de tareas por historias de usuario.

Historia de Usuario	Tareas
Gestionar usuarios.	Insertar usuarios. Listar usuarios. Modificar usuarios. Eliminar usuarios.
Gestionar Mineral.	Insertar minerales Generador de formulas Modificar minerales. Eliminar minarles.
Listar mineral por tipo dureza.	Listar por dureza baja. Listar por dureza media.



Capítulo 3- Desarrollo y Pruebas

Identificar minerales.	Listar por dureza alta. Capturar datos y realizar búsqueda. Listar resultados de la identificación. Limpiar datos y resultado de la identificación.
Visualizar la información de un mineral. Buscar datos de un mineral por nombre.	Mostrar los datos de un mineral. Capturar nombre y mostrar datos del mineral.
Buscar minerales por criterios de búsqueda.	Capturar los datos y realizar la búsqueda. Listar resultado de la búsqueda. Limpiar lista de resultados y datos.
Generar reporte	Generar reporte.
Identificación de tipos rocas ígneas.	Mostrar diagramas. Graficar y mostrar resultados. Limpiar datos y gráfica.

- **Historias de usuarios abordadas en la primera iteración.**

Tabla 3.2: Historias de usuario abordadas en la primera iteración.

Historias de usuario	Estimación en (semanas).	
	Estimación inicial.	Estimación real.
Gestionar Usuarios.	1	0.4
Gestionar Mineral.	3	2.4

Tabla 3.3: Historias de usuario abordadas en la segunda iteración.

Historias de usuario	Estimación en (semanas).	
	Estimación inicial.	Estimación real.
Listar mineral por tipo dureza.	1	0.3
Identificar minerales.	1	0.4
Visualizar la información de un mineral.	0.1	0.1
Buscar datos de un mineral por nombre.	0.1	0.1



Capítulo 3- Desarrollo y Pruebas

Tabla 3.4: Historias de usuario abordadas en la tercera iteración.

Historias de usuario	Estimación en (semanas).	
	Estimación inicial.	Estimación real.
Buscar minerales por criterio de búsqueda.	3	2.2

Tabla 3.5: Historias de usuario abordadas en la cuarta iteración.

Historias de usuario	Estimación en (semanas).	
	Estimación inicial.	Estimación real.
Generar reporte	2.4	2.2

Tabla 3.6: Historias de usuario abordadas en la quinta iteración.

Historias de usuario	Estimación en (semanas).	
	Estimación inicial.	Estimación real.
Identificación de tipo de rocas ígneas.	3	3

- **Tares de ingeniería.**

Tabla 3.7: Tarea de ingeniería. 1: Insertar Usuario.

Tarea de ingeniería	
Número de tarea: 1	Número historia: 1
Nombre de tarea: Insertar usuario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 27/01/14.	Fecha de fin: 27/01/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite insertar los nuevos usuarios a la base de datos.	

Para ver el resto de las Tarjetas de ingeniería ir al: [Anexo 3.](#)



3.3. Principales interfaces de la aplicación.

En este capítulo tiene lugar la implementación de la Aplicación a realizar, en el mismo se muestra la imagen de la interfaz principal con la cual interactuará el usuario.

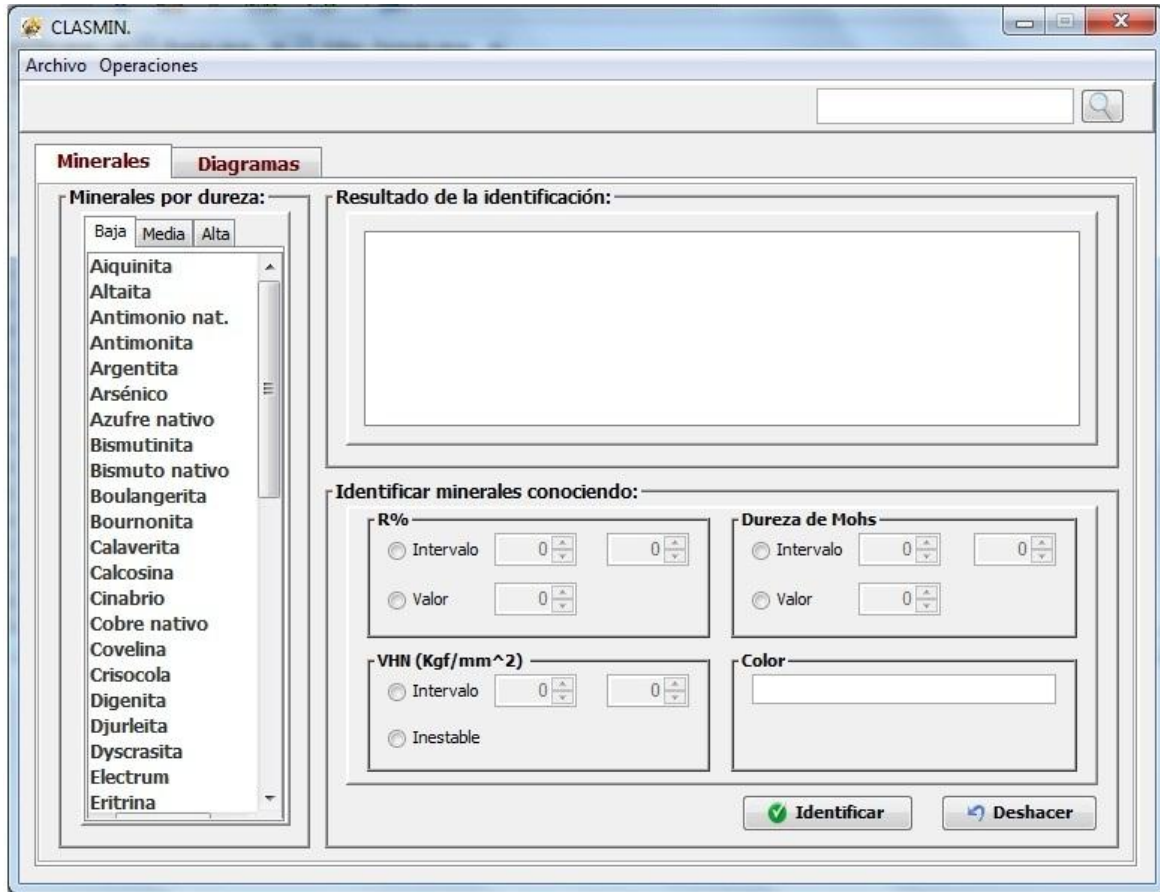


Figura 3.1: Interfaz principal.

Para ver el resto de las interfaces ir al: [Anexo 4.](#)

3.4. Pruebas.

En la Metodología XP las pruebas juegan un papel fundamental, pues esta permite la comprobación continua del código. El desarrollo constante de las pruebas da lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo. En esta metodología hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación. [21]



3.4.1 Pruebas de Aceptación.

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación.

Las pruebas de aceptación son creadas a partir de las historias de usuario. Durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada.

Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración o se considerará que el equipo de desarrollo no realiza ningún progreso.

Cada una de estas pruebas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas.

La garantía de calidad es una parte esencial en el proceso de XP. La realización de este tipo de pruebas y la publicación de los resultados debe ser los más rápidos posibles, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios. [14]

Tabla 3.30: Planilla de prueba de aceptación.

Prueba de Aceptación
HU: Nombre de la HU.
Nombre: Nombre del caso de Prueba.
Descripción: Breve descripción de la prueba.
Condiciones de ejecución: Precondiciones para realizar la prueba.
Entrada/Pasos ejecución: Pasos para probar funcionalidad.
Resultado: Resultado de la prueba.
Evolución de la prueba: Aceptada o Denegada.



Tabla 3.31: Prueba de aceptación HU 1: Gestionar usuarios.

Prueba de Aceptación
HU: 1
Nombre: Prueba para validar que sea administrador.
Descripción: El administrador selecciona en el menú Operaciones la opción de Administrar y luego Gestionar Usuarios.
Condiciones de ejecución: Se deben de introducir los datos que se pidan el formulario. Debe de estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Si los datos son correctos mostrará un formulario en el que podrá gestionar los usuarios. Se muestra un mensaje de error cuando: <ul style="list-style-type: none">✓ No escribe el nombre o contraseña.✓ El nombre o la contraseña no son correctos.
Resultado: Comprueba correctamente si es administrador
Evolución de la prueba: Aceptada.

Para ver el resto de las pruebas de aceptación ir al: [Anexo 5.](#)

3.3. Conclusiones del capítulo.

En este capítulo se llevaron a cabo las fases de Desarrollo y Prueba donde se realizó el desarrollo de las iteraciones a partir de la distribución de tareas por historias de usuarios, se presentan las principales interfaces de la aplicación y se les practicó las pruebas de aceptación a las funcionalidades de mayor importancia.



4.1. Introducción.

En la actualidad para un desarrollo satisfactorio de cualquier proyecto se hace imprescindible el estudio de factibilidad para tener en cuenta una estimación de los costos a incurrir logrando así definir si será factible o no desarrollar el mismo.

Hay muchas formas de calcular el costo, pero para nuestro proyecto se utilizará la metodología Costo - Beneficio, la cual sugiere que la conveniencia de la ejecución de un proyecto se determina por la observación de ciertos factores en conjunto, estos son:

- ✚ El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- ✚ La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo).[23]

4.2. Efectos Económicos.

Los efectos económicos se clasifican en:

- ✚ Efectos directos.
- ✚ Efectos indirectos.
- ✚ Efectos externos.
- ✚ Intangibles.

- **Efectos directos:**

Positivos:

- ✚ Se agilizará el proceso de identificación de minerales en secciones pulidas.
- ✚ La identificación de rocas ígneas se realizará de forma más sencilla.



- ✚ Se evitará el deterioro de las bibliografías existentes utilizadas en la realización del proceso.

Negativos:

Para usar la aplicación es imprescindible el uso de un ordenador por lo que esto trae aparejado gastos por consumo de corriente eléctrica.

- ***Efectos indirectos:***

Debido a que la aplicación no está construida con un objetivo de venta, sus efectos económicos indirectos no son de un valor significativo.

- ***Efectos externos:***

Se obtendrá un producto disponible que les facilitará todo el trabajo a los profesores y estudiantes involucrados en el proceso de identificación de minerales en secciones pulidas de la carrera de geología del ISMM.

- ***Intangibles:***

A la hora de realizar la valoración económica siempre hay elementos apreciables por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.

Con el fin de evaluar con precisión los efectos, deberán considerarse dos situaciones: la situación sin proyecto y la situación con proyecto.

Situación sin proyecto:

El desarrollo del proceso de identificación de minerales en secciones pulidas, al igual que la identificación de rocas ígneas se realiza de forma manual, provocando esto gran pérdida de tiempo a los profesores o estudiantes, gastos de papel y deterioro de las bibliografías utilizadas en dichos procesos.

Situación con proyecto:

Con la realización de la aplicación se agilizó en gran manera el proceso de identificación de minerales en secciones pulidas, la identificación de rocas ígneas y se minimizó el uso de bibliografía en formato duro, así como el uso de papel.



4.3. Beneficios y Costos intangibles en el proyecto:

Costo:

- ✚ Resistencia al cambio, es decir que puede variar el costo.

Beneficios:

- ✚ Mayor rapidez a la hora de realizar el proceso.
- ✚ Facilidad al interpretar los resultados.
- ✚ Los datos se encuentran centralizados y disponibles para su uso en cualquier momento.

4.4. Fichas de Costo:

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una ficha de costo de un producto informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

- **Costo en Moneda Libremente Convertible**

Tabla 4.1: Costo en Moneda Libremente Convertible.

Ficha de Costo		
Costo Moneda Libremente Convertible		Precios(\$)
Costos Directos		
Comprar de equipos de cómputos		0,00
Alquiler de equipos de cómputos		0,00
Compra de licencia de software		0,00
Depreciación de equipos		16,66
Materiales directos		0,00
	Subtotal	16,66
Costos Indirectos		
Formación del personal que elabora el proyecto		0,00
Gastos en llamadas telefónicas		0,00
Gastos para mantenimientos del centro		0,00



Capítulo 4 Estudio de Factibilidad

Know how		0,00
Gastos en representación		0,00
	Subtotal	0,00
Gastos de Distribución y Ventas		
Participación en ferias o exposiciones		0,00
Gastos en transportación		0,00
Compras de materiales de propaganda		0,00
	Subtotal	0,00
	Total	16,66

- **Costo en Moneda Nacional**

Tabla 4.2: Costo en Moneda Nacional.

Ficha de Costo		
Costo Moneda Nacional		Precios(\$)
Costos Directos		
Salario del personal que laborará en el proyecto		100,00
12,5% de total de gastos por salarios se dedican a la seguridad social		0,00
9,09% de salario total, por conceptos de vacaciones a acumular		0,00
Gasto por consumo de energía eléctrica		10,70
Gasto en llamadas telefónicas		0,00
Gastos administrativos		0,00
	Subtotal	110,70
Costos Indirectos		
Know how		0,00
	Subtotal	0,00
	Total	110,70



Capítulo 4 Estudio de Factibilidad

El análisis de costo-beneficio se basa en un principio muy simple:

Compara los beneficios y los costos de un proyecto particular y si los primeros exceden a los segundos entrega un elemento de juicio inicial, que indica su aceptabilidad.

Mientras que el análisis costo-efectividad sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda; si no en productos. Para esta técnica es imprescindible definir una variable directa que haga variar los costos. [18]

Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo, el tiempo en minutos empleado para realizar la identificación de mineral en secciones pulidas y la variable sería, complejidad de las pruebas que se desarrollan durante este proceso.

Valores de la variable (solución manual):

- Identificar mineral (45 min).
- Identificar tipo de roca ígnea (10 min).
- Generar reporte (15 min).

Valores de la variable (solución con la aplicación):

- Identificar mineral (2 min).
- Identificar tipo de roca ígnea (0.30 min).
- Generar reporte (1 min).

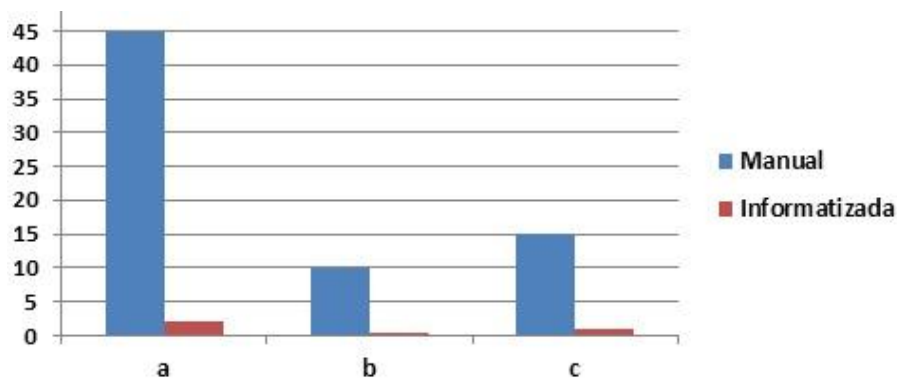


Figura 4.1: Gráfica de comparación de la solución manual y la informatizada.



El gráfico que se presentó anteriormente muestra el comportamiento de las variables, teniendo en cuenta las formas de realización de las actividades que componen el proceso. Observando los resultados reflejados en la gráfica queda demostrada la factibilidad de la aplicación desarrollada.

4.5. Conclusiones del capítulo.

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Beneficio, se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$16.66 CUC y \$110.70 CUP por mes, teniendo en cuenta que el software se desarrolló en un período de 4 meses entonces el costo total es de \$66,64 CUC y \$ 442,80 CUP, demostrándose la factibilidad económica del proyecto.

.



Conclusiones Generales

Durante la realización de este trabajo se dio cumplimiento a los objetivos específicos concebidos para el desarrollo de la investigación, dando así cumplimiento al objetivo general del trabajo, obtener un sistema informático seguro, confiable y de fácil utilización.

De manera general se realizó:

- ✚ Un análisis teórico y conceptual de los principales elementos a tener en cuenta para el desarrollo del trabajo de diploma.
- ✚ Un estudio de las metodologías y herramientas a utilizar para el diseño e implementación de la aplicación.
- ✚ La implementación la aplicación para las identificaciones de minerales en secciones pulidas.
- ✚ El estudio de la factibilidad y la valoración de la sostenibilidad del producto arrojando un resultado satisfactorio.

Luego de los resultados arrojados por las pruebas de aceptación evaluadas por el cliente y el análisis de factibilidad realizado podemos concluir que el trabajo responde como propuesta de solución al problema específico planteado.



Recomendaciones

A modo general los objetivos trazados al inicio de este trabajo han sido logrados, pero al mismo tiempo, a lo largo del proceso de desarrollo, ha quedado claro que la propuesta es sólo la primera fase de un proyecto que puede ser mucho más ambicioso.

Por tanto se hacen las siguientes recomendaciones:

- ✚ Agregarle un módulo para la clasificación de rocas ultramáficas.
- ✚ Agregarle otras funcionalidades que permita ser su uso extensivo a las especialidades de Minería y Metalúrgica.
- ✚ Realizar una nueva versión portable de la aplicación utilizando algún tipo de base de datos embebida.



Referencias Bibliográficas

1. KERR, P. F. *Mineralogía óptica*. Madrid, 1972.
2. Mineral [en línea]: 20 de octubre 2009. [Consultado el 9 de enero del 2014]. Disponible en: http://html.rincondelvago.com/minerales_6.html
3. jVPM. [en línea]: 17 de septiembre del 2013. [Consultado el: 10 de enero 2014]. Disponible en: www.softpedia.es/programa-jVPM-Java-Virtual-Petrigraphic-Microscope-242044.html
4. Diagrama de Streckeisen. [en línea]: 25 de junio del 1997. [Consultado el: 18 de marzo 2014]. Disponible en: <http://www.geovirtual.cl/geologiageneral/ggcap04a.htm>
5. Software Libre. [en línea]: 11 de febrero de 2010. [Consultado el: 9 de enero del 2014]. Disponible en: <http://hispalinux.es/SoftwareLibre>
6. Lenguaje de Programación. [en línea]: 11 de febrero de 2010. [Consultado el: 13 de enero del 2014]. Disponible en: <http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php>
7. SUBIRÓS MUÑOZ, D. R. *Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2009.
8. PIERRA FUENTES, A. *Sistema de Gestión de Información de los Parámetros del Viento*. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2013.
9. Características de Java. 11 de febrero de 2010. [Consultado el: 13 de enero del 2014]. Disponible en: <http://sheyla88.blogspot.es/>



10. Sistema gestor de Bases de Datos o SGBD [en línea]: 2008. [Consultado el: 15 de enero del 2014] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sqbd/>.
11. CUENCA PEÑA, J.R. *Desarrollo de los Módulos Peso Volumétrico y Granulometría en el Sistema de Gestión de Informes de Ensayos*. Trabajo de diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2013.
12. MySQL. [en línea][Consultado el: 15 de enero del 2014]. Disponible en: <http://www.gridmorelos.uaem.mx/~mcruez//cursos/miic/MySQL.pdf>
13. LETELIER, P; PANADÉS, C. M. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia.
14. CORBEA RODRÍGUEZ, M; ORDÓÑEZ PÉREZ, M. *La metodología xp aplicable el desarrollo del software en cuba*. Trabajo de diploma. Universidad de Ciencias Informáticas, 2007
15. MENDOZA SANCHES, M.A. *Metodología de Desarrollo de Software*. Perú, 2004.
16. Patrones arquitectónicos. [en línea]: 9 de agosto del 2012. [Consultado el: 23 de enero del 2014]. Disponible en: <http://ingesoftware2vr.blogspot.com/2012/08/patrones-de-arquitectura.html>
17. JIMENES IGLESIA, E.R. *Sistema automatizado para la Gestión del producto final de los servicios de la UEB Organización Empresarial de la Empresa Empleadora del Níquel*. Trabajo de diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2009.
18. SALAZAR PUPO, R. *Sistema de Gestión de Capacitación ECRIN*. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2013.
19. NetBeans. [en línea]: 7 de marzo del 2012. [Consultado el: 10 de enero del 2014]. Disponible en: <http://informaticaivanozaeta.blogspot.com/2012/03/netbeans.html>



Referencias Bibliográfica

20. ROMMELL LAITANO, P.A. *Clase de Ingeniería de Software Asistido por Computador*. Universidad Católica de Honduras.
21. NARANJO AGUILERAS, J.R. *Herramienta Informática para la Gestión de información de los Trabajos de Diplomas en la carrera de Ingeniería Informática del Instituto Superior Minero Metalúrgico de Moa*. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2013.
22. Requisitos no Funcionales. [en línea]: 7 de julio del 2008. [Consultado el: 20 de enero del 2014]. Disponible en: <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>
23. HERNANDEZ GARCELL, A.S. *Desarrollo del Módulo Peso Específico de los Suelos para el Sistema de Gestión de Informes de Ensayos*. Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, 2013.



Bibliografía

- ✚ Beck, K. *Una explicación de la programación .extrema. Aceptar el cambio*. Addison Wesley, 2000.
- ✚ Calcular depreciación. [en línea]: 16 de marzo del 2008. [Consultado el: 20 de febrero del 2014]. Disponible en: <http://www.economicas-online.com/bienesde5.htm>
- ✚ Creación de reportes con iReport. [en línea]: febrero del 2009. [Consultado el: 3 de marzo del 2014]. Disponible en: www.javatutoriales.com/2009/02/creacion-de-reportes-con-jasper
- ✚ Diagrama de Streckeisen. [en línea]: 28 de agosto del 2009. [Consultado el: 20 de febrero del 2014]. Disponible en: <http://geologiavenezolana.blogspot.com/p/diagramas-y-esquemas.html>
- ✚ GÁLVEZ ROJAS, S; ALCAIDE GARCÍA, M; MORA MATA, M. A. *Java a tope: (Cómo tratar con figuras, imágenes y textos en dos dimensiones)*. Edición electrónica. Universidad de Málaga.
- ✚ GILLISPIE, M. R; STYLE, M. T. *Classification of Igneous Rocks*. British Geological Survey, 1999, vol. 1.
- ✚ HERNÁNDEZ MELÉNDREZ, E. *Cómo escribir una tesis*. Escuela Nacional de Salud Pública, 2006.
- ✚ HURLBUT, D. *Manual de Mineralogía*. 2da ed. Barcelona – Buenos Aires – México: editorial Reverté, S.A.
- ✚ Manual de SQL. [en línea]: 10 de abril del 2011. [Consultado el: 13 de enero del 2014]. Disponible en: <http://www.webexperto.com/manuales>



- ✚ Metodología Costo Benéfico. [en línea]: 10 de abril del 2011. [Consultado el: 13 de enero del 2014]. Disponible en:
<http://www.ongei.gob.pe/publica/metodologias/Lib5006/n00.htm>

- ✚ MySQL. [en línea]. [Consultado el: 10 de enero del 2014]. Disponible en:
<http://dev.mysql.com/>

- ✚ Tarjeta CRC. [en línea]: 18 de septiembre del 2003. [Consultado el: 17 de enero del 2014]. Disponible desde:
<http://jms32.eresmas.net/tacticos/UML/UML04/UML0402.html>

- ✚ Wake, W.C. *Extreme Programming Explored*. Addison-Wesley. 2002.

- ✚ ZUKOWSKI, J. *Programación Java 2*. Editado por Ediciones Anaya Multimedia. Madrid, 2003.

- ✚ ZUKOWSKI, J. *The Definitive Guide to Java Swing*. USA, 2005.



Glosario de Términos

Aplicación: Programa preparado para una utilización específica. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

Costo: En economía el coste o costo es el valor monetario de los consumos de factores que supone el ejercicio de una actividad económica destinada a la producción de un bien o servicio.

Clase: Una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

Cliente: Persona, organización o grupo de personas que solicita la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

GPL: Una licencia creada por la Fundación del Software Libre en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

IDE: Entorno de desarrollo integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Informatización: Aplicar los conjuntos de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.



JDK: Es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

Know how: (del inglés *saber-cómo*) o **Conocimiento Fundamental** es una forma de transferencia de tecnología. Es una expresión anglosajona utilizada en los últimos tiempos en el comercio internacional para denominar los conocimientos preexistentes no siempre académicos, que incluyen: técnicas, información secreta, teorías e incluso datos privados (como clientes o proveedores).

Licencia BSD: Es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Es una licencia de software libre permisiva como la licencia de OpenSSL. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Metodología Ágil: Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías tradicionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Metodologías de desarrollo de software: Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Open-Source: Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

Proyecto: Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

Release: Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.



Roca: Al material compuesto de uno o varios minerales como resultado final de los diferentes procesos geológicos.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Spike: Es un programa muy simple que explora una posible solución al problema. A partir de estos pequeños programas se construye la solución del problema.



Anexo 1: Historias de Usuarios

Tabla 2.6: HU Gestionar mineral.

Historia de Usuario	
Número: 2	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Gestionar minerales.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Roger René Ramírez González.	
Descripción: Solo podrán ser gestionados los minerales por los usuarios autenticados. Los que podrán insertar, modificar y eliminar minerales.	
Observaciones: Confirmado con el cliente.	

Tabla 2.7: HU Listar minerales por tipo de dureza.

Historia de Usuario	
Número: 3	Usuario: Ing. Iván Barea Pérez.
Nombre de la Historia: Listar minerales por tipo de dureza.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Roger René Ramírez González.	
Descripción: Se muestra al usuario una lista de minerales diferenciados por el tipo de dureza existente en la base de datos, la cual permite visualizar la información de un mineral que sea seleccionado. La misma se actualiza si se inserta, modifica o elimina los minerales existentes en la base de datos.	
Observaciones: Confirmado con el cliente.	



Tabla 2.8: HU Identificar minerales.

Historia de Usuario	
Número: 4	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Identificar minerales.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Roger René Ramírez González.	
Descripción: El usuario introduce los valores de (r%, vhn, Mohs) y el color y da clic al botón aceptar, el resultado se mostrará en una lista. De la misma se podrá visualizar los datos de cualquier mineral identificado. Contará con un botón para limpiar los datos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.9: HU Visualizar los datos de un mineral.

Historia de Usuario	
Número: 5	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Visualizar datos de un mineral	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Roger René Ramírez González.	
Descripción: Al ser seleccionado cualquier mineral tanto de la lista de minerales por dureza, lista de minerales identificados o al realizar la búsqueda de mineral por nombre se mostrará un formulario con los datos de dicho mineral.	
Observaciones: Confirmado con el cliente.	



Tabla 2.10: HU Buscar datos de un mineral por nombre.

Historia de Usuario	
Número: 6	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Buscar datos de un mineral por nombre.	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Roger René Ramírez González.	
Descripción: El usuario introduce el nombre de un mineral que desee visualizar sus datos, si existe en la base de datos se mostrará el formulario de visualizar con los datos.	
Observaciones: Confirmado con el cliente.	

Tabla 2.11: HU Buscar minerales por criterios de búsqueda.

Historia de Usuario	
Número: 7	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Buscar minerales por criterios de búsqueda.	
Prioridad en el negocio: media	Riesgo en el desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Roger René Ramírez González.	
Descripción: El usuario selecciona los criterios por lo que desea realizar la búsqueda e introduce los datos y le da clic a botón aceptar. El resultado se muestra en una lista.	
Observaciones: Confirmado con el cliente.	



Tabla 2.12: HU Generar reporte.

Historia de Usuario	
Número: 8	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Generar reporte.	
Prioridad en el negocio: Media	Riesgo en el desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Roger René Ramírez González.	
Descripción: El usuario da clic en el botón exportar para exportar los datos de uno o varios minerales que desee.	
Observaciones: Confirmado con el cliente.	

Tabla 2.13: HU Identificar tipos de rocas ígneas.

Historia de Usuario	
Número: 9	Usuario: Ing. Iván Barea Pérez
Nombre de la Historia: Identificación de tipo de rocas ígneas.	
Prioridad en el negocio: media	Riesgo en el desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 5
Programador responsable: Roger René Ramírez González.	
Descripción: El usuario selecciona la vista de diagrama, selecciona el tipo de diagrama que desea utilizar, le introduce los datos y presiona el botón graficar, en la parte inferior derecha se muestra el resultado (tipo de roca y los valores de Q, A, P).	
Observaciones: Confirmado con el cliente.	



Anexo 2: Tarjetas CRC.

Tabla 2.18: Tarjeta CRC: Mineral.

Mineral	
Descripción: Almacena los datos de los minerales.	
Atributos:	
Nombre	Descripción
Id	Id
nombre	
r	
dureza_mohz	
vhn	
color	
reflejos_internos	
paragenesis	
criterio_identificacion	
tipo_sistema	
bireflexion	
clasificacion_isotropia	
dureza_mineral	
Responsabilidades:	
Nombre	Colaborador
insertar()	mineral, sistema, dureza_mineral, bireflexión, isotropia_anisotropia, imágenes, conexión
modificar()	mineral, sistema, dureza_mineral, bireflexión, isotropia_anisotropia, imágenes, conexión



DropMineral()	mineral, imágenes, conexión
findMineralData	mineral, sistema, dureza_mineral, bireflexión, isotropia_anisotropia, imágenes, conexión
getMinDurezBaja()	mineral, conexión
getMinDurezMedia()	mineral, conexión
getMinDurezAlta()	mineral, conexión
getDatos()	mineral, conexión
Buscar()	mineral, conexión
igualCriterio1()	mineral, conexión
posibleValor1()	mineral, conexión
CriterioUno()	mineral, conexión
igualCriterio1y2()	mineral, conexión
igualCriterio1Dos()	mineral, conexión
igualCriterioUno2()	mineral, conexión
igualCriterioUnoDos()	mineral, conexión
igualCriterio1_2y3()	mineral, conexión
igualCriterioUnoDos3()	mineral, conexión
igualCriterioUno2Tres()	mineral, conexión
igualCriterio1DosTres()	mineral, conexión
igualCriterioUno2y3()	mineral, conexión
igualCriterio1y2Tres()	mineral, conexión
igualCriterio1Dos3()	mineral, conexión
igualCriterio1_2_3()	mineral, conexión
listaNombresMinerales()	mineral, conexión
subNombreMinerales()	mineral, conexión
buscarColor()	mineral, conexión
Colores()	mineral, conexión



Tabla 2.19: Tarjeta CRC: Imágenes

Imágenes	
Descripción: almacena los nombres y la dirección de las imágenes. Siendo el nombre el mismo del mineral al que pertenece la imagen.	
Atributos:	
Nombre	Descripción
id_imagen	id_imagen
dirección	
Responsabilidades:	
Nombre	Colaborador
insertar()	mineral, conexión
Modificar()	mineral, conexión

Tabla 2.20: Tarjeta CRC: Conexión.

Conexión	
Descripción: conecta la aplicación con la base de datos	
Atributos:	
Nombre	Descripción
servidor	
puerto	
nombre_bd	
usuario	
password	
Responsabilidades:	
Nombre	Colaborador
Conectar()	mineral, sistemas, imágenes, usuarios, dureza_mineral, Isotropia_anisotropia, bireflexión, formula.
executeSQL()	



Tabla 2.21: Tarjeta CRC: Sistemas.

Sistema	
Descripción: almacena los tipos de sistema de los minerales.	
Atributos:	
Nombre	Descripción
id_sistema	Id_sistema
clasificación_sistema	
Responsabilidades:	
Nombre	Colaborador
	mineral, conexión

Tabla 2.22: Tarjeta CRC: Fórmula química

Formula_quimica	
Descripción: almacena las formulas en texto plano en la tabla fórmula.	
Atributos:	
Nombre	Descripción
formula_quimica	
Responsabilidades:	
Nombre	Colaborador
insertarFormula()	conexión, formulas
findFormula()	conexión, formula
deleteFormula()	conexión, formula
findFormulaMineral()	mineral, conexión



Anexo 3: Tarjetas de Ingeniería.

Tabla 3.8: Tarea de ingeniería No. 2: Listar usuarios.

Tarea de ingeniería	
Número de tarea: 2	Número historia: 1
Nombre de tarea: Listar usuarios.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 28/01/14.	Fecha de fin: 28/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar la lista con los nombres de los usuarios existentes en la base de datos.	

Tabla 3.9: Tarea de ingeniería No. 3: Modificar usuarios.

Tarea de ingeniería	
Número de tarea: 3	Número historia: 1
Nombre de tarea: Modificar usuarios	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 29/01/14.	Fecha de fin: 29/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite modificar la contraseña y el privilegio de administración de los usuarios existentes en la base de datos.	



Tabla 3.10: Tarea de ingeniería No. 4: Eliminar usuarios.

Tarea de ingeniería	
Número de tarea: 4	Número historia: 1
Nombre de tarea: Eliminar usuarios.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 30/01/14.	Fecha de fin: 30/01/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite eliminar los usuario existentes en la base de datos	

Tabla 3.11: Tarea de ingeniería No. 5: Insertar mineral.

Tarea de ingeniería	
Número de tarea: 5	Número historia: 2
Nombre de tarea: Insertar mineral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 03/02/14.	Fecha de fin: 07/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite insertar los datos de los minerales en la base de datos.	



Tabla 3.12: Tarea de ingeniería No. 6: Generador de fórmula.

Tarea de ingeniería	
Número de tarea: 6	Número historia: 2
Nombre de tarea: Generador de fórmula	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.4
Fecha de inicio: 10/02/14.	Fecha de fin: 14/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite generar las formulas químicas, cuando se inserta un nuevo mineral, o se modifica la fórmula de uno existente.	

Tabla 3.13: Tarea de ingeniería No. 7: Modificar mineral.

Tarea de ingeniería	
Número de tarea: 7	Número historia: 2
Nombre de tarea: Modificar mineral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.4
Fecha de inicio: 17/02/14.	Fecha de fin: 20/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite modificar los datos de los minerales en la base de datos.	



Tabla 3.14: Tarea de ingeniería No. 8: Eliminar mineral.

Tarea de ingeniería	
Número de tarea: 8	Número historia: 2
Nombre de tarea: Eliminar mineral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 21/02/14.	Fecha de fin: 21/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite eliminar los minerales existentes en la base de datos.	

Tabla 3.15: Tarea de ingeniería No. 9: Listar por dureza baja.

Tarea de ingeniería	
Número de tarea: 9	Número historia: 3
Nombre de tarea: Listar por dureza baja.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 24/02/14.	Fecha de fin: 24/01/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar una lista con todos los minerales existentes en la base de datos que el tipo de dureza es baja.	



Tabla 3.16: Tarea de ingeniería No. 10: Listar por dureza media.

Tarea de ingeniería	
Número de tarea: 10	Número historia: 3
Nombre de tarea: Listar por dureza media.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 25/02/14.	Fecha de fin: 25/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar una lista con todos los minerales existentes en la base de datos que el tipo de dureza es media.	

Tabla 3.17: Tarea de ingeniería No. 11: Listar por dureza Alta.

Tarea de ingeniería	
Número de tarea: 11	Número historia: 3
Nombre de tarea: Listar por dureza alta.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 26/02/14.	Fecha de fin: 26/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar una lista con todos los minerales existentes en la base de datos que el tipo de dureza es alta.	



Tabla 3.18: Tarea de ingeniería No. 12: Capturar datos y realizar búsqueda.

Tarea de ingeniería	
Número de tarea: 12	Número historia: 4
Nombre de tarea: Capturar datos y realizar búsqueda.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.2
Fecha de inicio: 27/02/14.	Fecha de fin: 28/02/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite capturar los datos que introduce el usuario los cuales son utilizados para realizar la búsqueda en la base de datos.	

Tabla 3.19: Tarea de ingeniería No. 13: Listar resultados de la identificación.

Tarea de ingeniería	
Número de tarea: 13	Número historia: 4
Nombre de tarea: Listar resultados de la identificación.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 03/03/14.	Fecha de fin: 03/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite que se llene la lista de resultados con los nombres de los minerales que tienen coincidencia con los datos capturados.	



Tabla 3.20: Tarea de ingeniería No. 14: Limpiar datos y resultados.

Tarea de ingeniería	
Número de tarea: 14	Número historia: 4
Nombre de tarea: Limpiar datos y resultados de la identificación.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 04/03/14.	Fecha de fin: 04/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite que se vacíe la lista de resultados de la identificación y se borra los campos de los datos.	

Tabla 3.21: Tarea de ingeniería No. 15: Visualizar datos de un mineral.

Tarea de ingeniería	
Número de tarea: 15	Número historia: 5
Nombre de tarea: Visualizar datos de un mineral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 05/03/14.	Fecha de fin: 05/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar un formulario con los datos del mineral que sea seleccionado.	



Tabla 3.22: Tarea de ingeniería No. 16: Capturar nombre y mostrar datos del mineral.

Tarea de ingeniería	
Número de tarea: 16	Número historia: 6
Nombre de tarea: Capturar nombre y mostrar datos del mineral.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 06/03/14.	Fecha de fin: 06 /03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite realizar una búsqueda por un nombre de un mineral introducido por el usuario y se muestra un formulario con los datos de dicho mineral.	

Tabla 3.23: Tarea de ingeniería No. 17: Capturar datos y realizar búsqueda.

Tarea de ingeniería	
Número de tarea: 17	Número historia: 7
Nombre de tarea: Capturar datos y realizar búsqueda.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha de inicio: 07/03/14.	Fecha de fin: 20/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite capturar los datos introducidos por el usuario para luego realizar la búsqueda.	



Tabla 3.24: Tarea de ingeniería No. 18: Listar resultado de la búsqueda.

Tarea de ingeniería	
Número de tarea: 18	Número historia: 7
Nombre de tarea: Listar resultado de la búsqueda.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 21/03/14.	Fecha de fin: 21/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite llenar la lista de resultado con los nombres de los minerales que coincidan con los datos capturados.	

Tabla 3.25: Tarea de ingeniería No. 19: Limpiar lista de resultados y datos.

Tarea de ingeniería	
Número de tarea: 19	Número historia: 7
Nombre de tarea: Limpiar lista de resultados y datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 24/03/14.	Fecha de fin: 24/03/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite limpiar la lista de resultados y los datos introducidos por el usuario.	



Tabla 3.26: Tarea de ingeniería No. 20: Generar reporte

Tarea de ingeniería	
Número de tarea: 20	Número historia: 8
Nombre de tarea: Exportar los datos de minerales.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2.2
Fecha de inicio: 25/04/14.	Fecha de fin: 28/05/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite generar reportes con los datos de los minerales.	

Tabla 3.27: Tarea de ingeniería No. 21: Mostrar diagramas.

Tarea de ingeniería	
Número de tarea: 21	Número historia: 9
Nombre de tarea: Mostrar diagramas	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.4
Fecha de inicio: 29/05/14.	Fecha de fin: 05/05/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite mostrar los diagramas de Streckeisen para la identificación de tipos de rocas ígneas.	



Tabla 3.28: Tarea de ingeniería No. 22: Graficar y mostrar resultados.

Tarea de ingeniería	
Número de tarea: 22	Número historia: 9
Nombre de tarea: Graficar y mostrar resultados.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2.1
Fecha de inicio: 06/05/14.	Fecha de fin: 03/06/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite graficar tomando los datos entrados por el usuario.	

Tabla 3.29: Tarea de ingeniería No. 23: Exportar los datos de minerales.

Tarea de ingeniería	
Número de tarea: 23	Número historia: 9
Nombre de tarea: Limpiar la gráfica, los datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.1
Fecha de inicio: 27/05/14.	Fecha de fin: 27/05/14.
Programador responsable: Roger René Ramírez González.	
Descripción: Esta tarea permite generar reportes con los datos de los minerales.	

Anexo 4: Interfaces

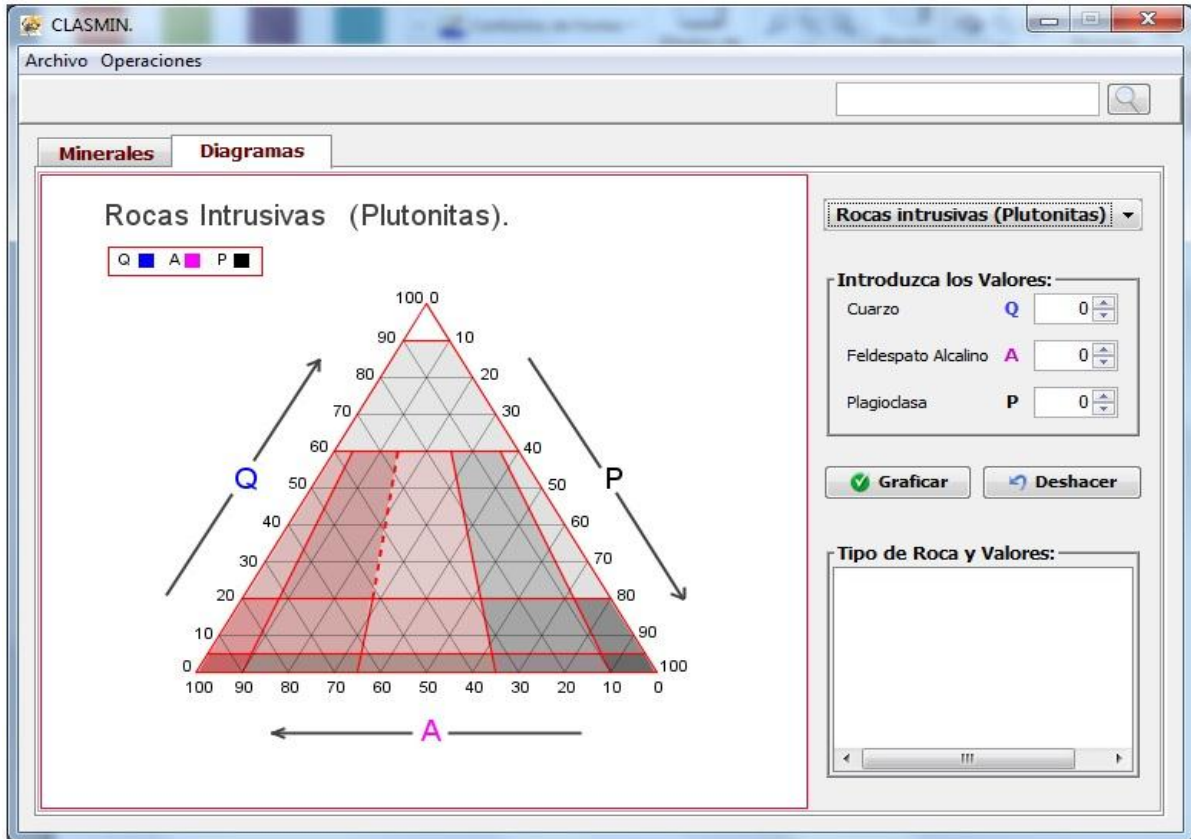


Figura 3.2: Diagrama QAP para rocas plutonitas.

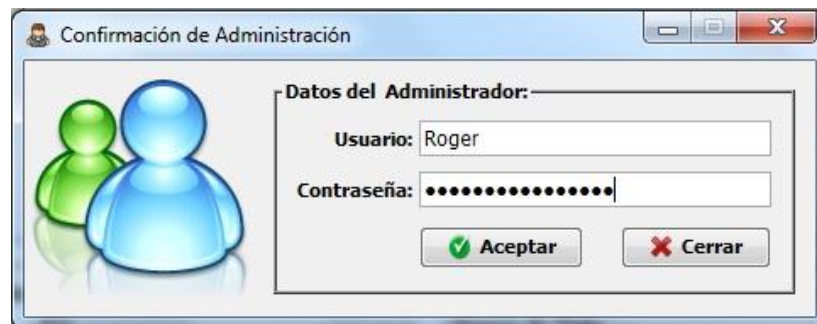


Figura 3.3: Confirmación de Administración.

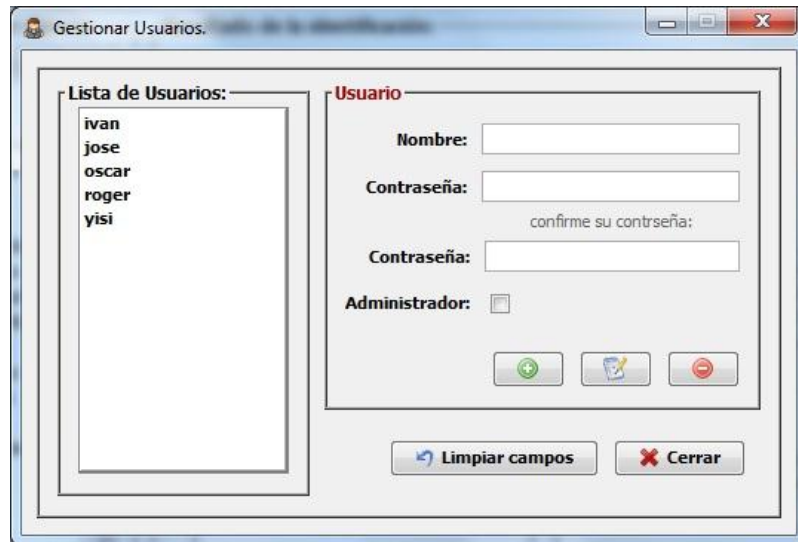


Figura 3.4: Gestionar Usuarios.

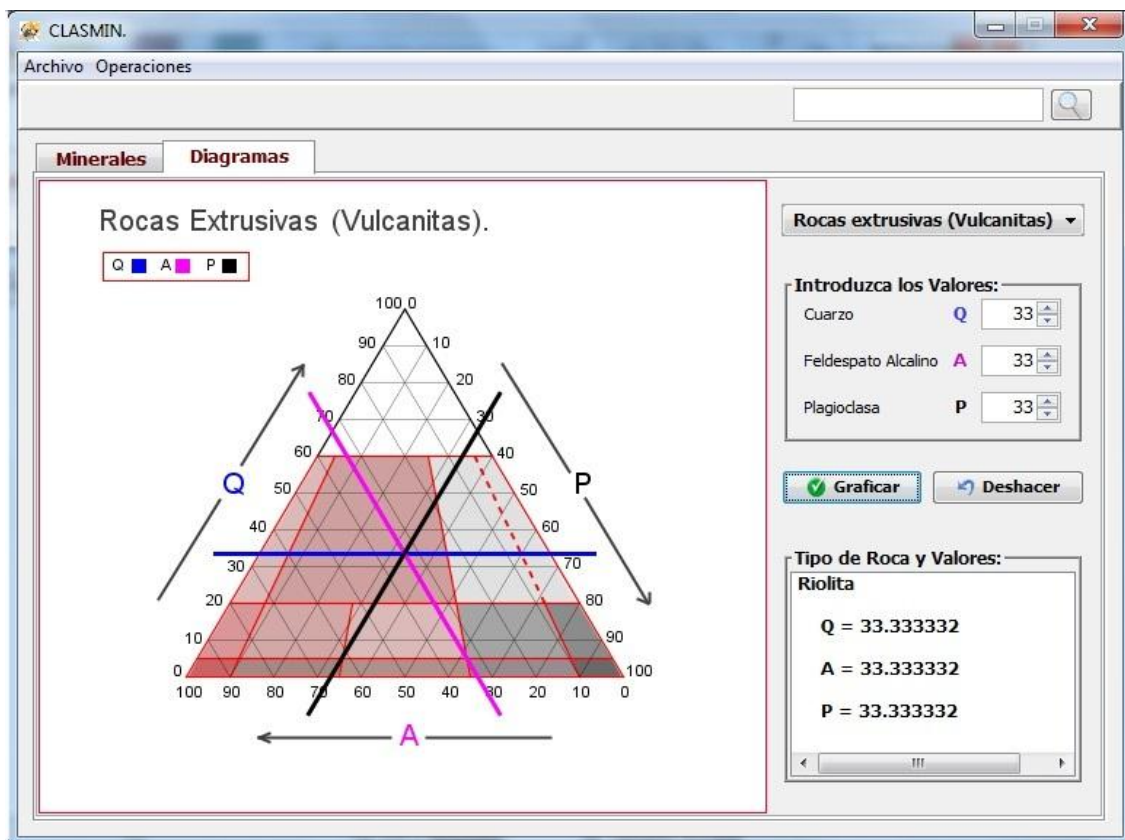


Figura 3.5: Ejemplo de identificación de tipo de roca vulcanita.

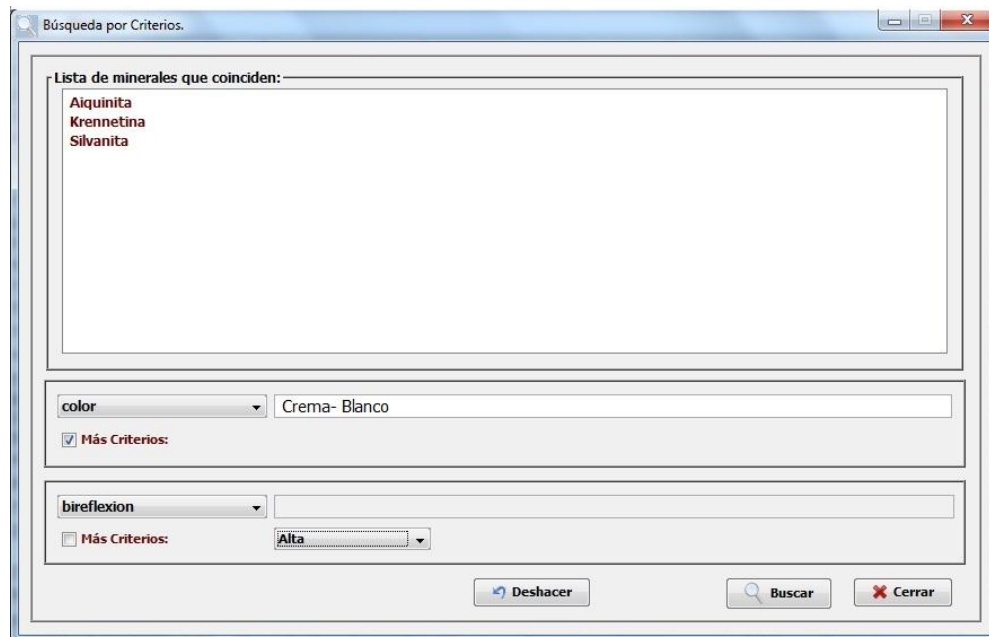


Figura 3.6: Búsqueda por Criterios.

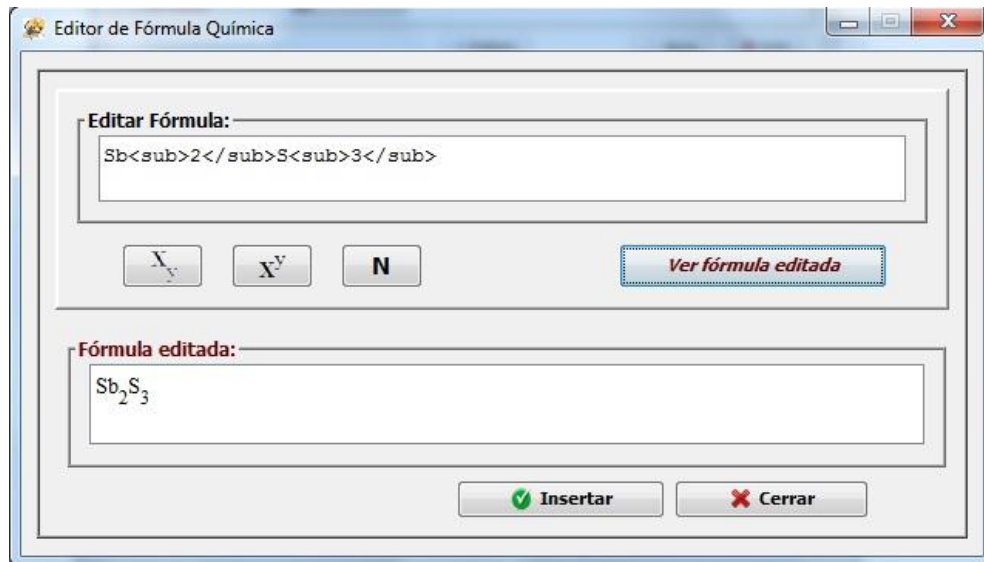


Figura 3.7: Generador de Fórmulas.



Figura 3.8: Gestionar Minerales.

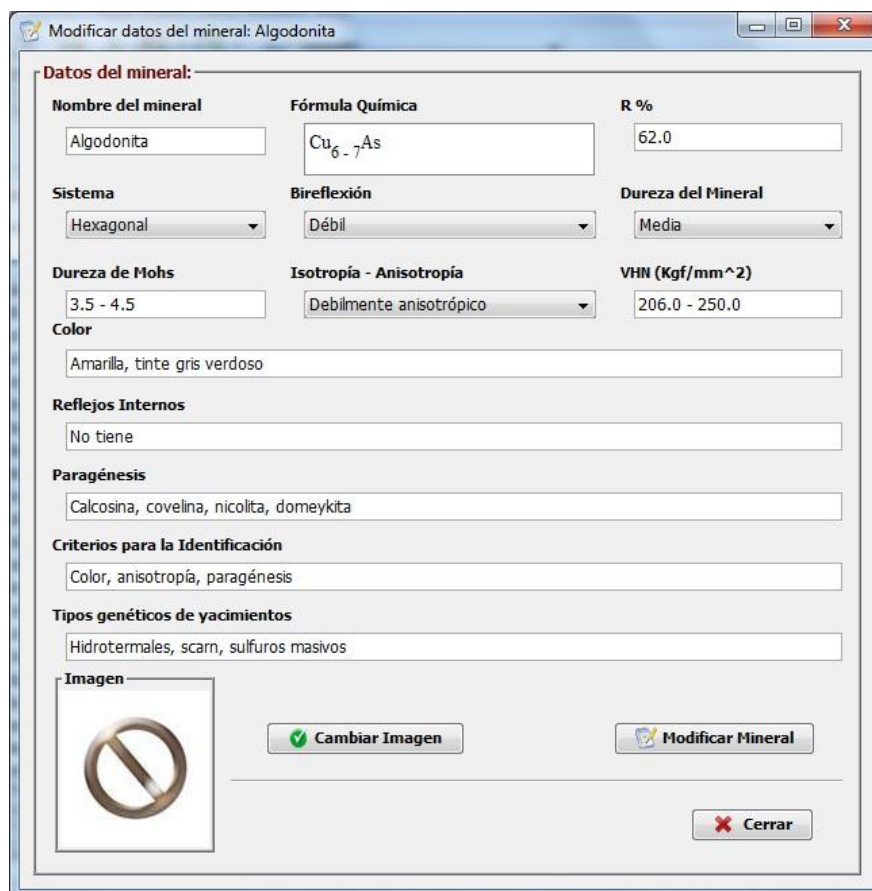


Figura 3.10: Modificar Mineral.



Anexo 5: Pruebas de aceptación

Tabla 3.32: Prueba de aceptación HU 1: Gestionar usuarios.

Prueba de Aceptación
HU:1
Nombre: Prueba para gestionar usuarios.
Descripción: El administrador selecciona en el menú Operaciones, Administrar, la opción de gestionar usuarios.
Condiciones de ejecución: Debe llenar el formulario para comprobar que es administrador. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Si los datos son escritos correctamente se mostrará un formulario en el que podrá insertar un usuario, luego podrá seleccionarlo de la lista para modificarlo o eliminarlo. Se mostrará un mensaje de confirmación. Se emite un mensaje de error en caso de que: <ul style="list-style-type: none">✓ El nombre es igual a otro existente en la lista.✓ Las contraseñas no coincidan.
Resultado: Gestiona correctamente el usuario.
Evolución de la prueba: Aceptada.



Tabla 3.33: Prueba de aceptación HU 2: Gestionar Mineral.

Prueba de Aceptación
HU: 2
Nombre: Prueba para gestionar minerales.
Descripción: El especialista selecciona en el menú Operaciones, Administrar, la opción de gestionar minerales.
Condiciones de ejecución: Se deben de introducir los datos que se piden en pantalla. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Los datos introducidos son para comprobar si es administrador. Elegirá insertar, modificar o eliminar. Se mostrarán mensajes de error cuando: <ul style="list-style-type: none">✓ No se introducen los datos.✓ El usuario o contraseña no sean correctos.✓ No se elige una opción
Resultado: Muestra el formulario de acuerdo a la opción elegida.
Evolución de la prueba: Aceptada.

Tabla 3.34: Prueba de aceptación HU 2: Gestionar Mineral.

Prueba de Aceptación
HU: 2
Nombre: Prueba para insertar mineral.
Descripción: El especialista selecciona en el menú Operaciones, Administrar, la opción: Gestionar Mineral. Se muestra un formulario. Selecciona la opción de insertar.
Condiciones de ejecución: Se deben de introducir los datos que se piden en pantalla. Se comprobará que tiene permiso para realizar esta operación. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Se mostrará un formulario que llenará con los datos, si presiona el botón aceptar, mostrará un mensaje de confirmación.
Resultado: Inserta correctamente el mineral.
Evolución de la prueba: Aceptada.



Tabla 3.35: Prueba de aceptación HU 2: Gestionar Mineral.

Prueba de Aceptación
HU: 2
Nombre: Prueba para modificar minerales.
Descripción: El especialista selecciona en el formulario de Gestionar Minerales la opción Modificar.
Condiciones de ejecución: Se deben de introducir los datos que se piden en pantalla. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Se mostrará un formulario con una lista de los minerales existentes en la que podrá seleccionar uno o escribir el nombre directamente en el campo nombre, luego presionará el botón modificar y se mostrará el formulario con los datos para que modifique lo que desee, si presiona el botona de modificar, mostrará mensaje de confirmación.
Resultado: Modifica correctamente el mineral.
Evolución de la prueba: Aceptada.

Tabla 3.36: Prueba de aceptación HU 2: Gestionar Mineral.

Prueba de Aceptación
HU: 2
Nombre: Prueba para eliminar minerales.
Descripción: El administrador selecciona en el formulario de Gestionar Minerales la opción Eliminar.
Condiciones de ejecución: Se deben de introducir los datos que se piden en pantalla. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Se mostrará un formulario con una lista de los minerales existentes en la que podrá seleccionar uno o escribir el nombre directamente en el campo nombre, luego presionará el botón eliminar. Mostrará mensaje de confirmación.
Resultado: Elimina correctamente el mineral.
Evolución de la prueba: Aceptada.



Tabla 3.37: Prueba de aceptación HU 3: Listar minerales por tipo de dureza.

Prueba de Aceptación
HU: 3
Nombre: Prueba para verificar si se lista correctamente por tipo de dureza.
Descripción: Se muestran las listas por tipo de dureza con los nombres de los minerales existentes en la Base de Datos.
Condiciones de ejecución: Debe de existir uno o más mineral en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Por defecto se muestra la lista de dureza baja con los nombres de los minerales. Se selecciona mediante un clic cualquier nombre lo que mostrará un formulario con todos los datos del mineral donde se comprobará si el tipo de dureza corresponde con el de la lista. Lo mismo para las demás listas. Si se inserta, modifica o elimina la lista se actualiza de acuerdo al tipo de dureza que sea asignado a dicho mineral.
Resultado: Lista correctamente los minerales.
Evolución de la prueba: Aceptada.

Tabla 3.38: Prueba de aceptación HU 4: Identificar minarles.

Prueba de Aceptación
HU: 4
Nombre: Prueba para identificar minerales.
Descripción: El especialista introduce los datos en los campos especificados los cuales son tomados para la realización de la identificación.
Condiciones de ejecución: Deben de existir minerales en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Si los datos son correctos se muestran en la lista de resultados el o los nombres de los minerales que coincidan con los datos introducidos. Si no se introduce ningún dato y se presiona el botón identificar, no ocurrirá nada.
Resultado: Identifica correctamente los minerales.
Evolución de la prueba: Aceptada.



Tabla 3.39: Prueba de aceptación HU 5: Visualizar información de un mineral.

Prueba de Aceptación
HU: 5
Nombre: Visualizar información de un mineral.
Descripción: El usuario selecciona el nombre de un mineral y se muestra un formulario con los datos de dicho mineral.
Condiciones de ejecución: Debe de existir el mineral en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Seleccionar mineral en la lista por tipo de dureza o lista de resultados de la identificación o lista de minerales que coinciden o realizando un búsqueda por nombre. Se mostrará un formulario con todos los datos del mineral.
Resultado: Muestra correctamente los datos del mineral.
Evolución de la prueba: Aceptada.

Tabla 3.40: Prueba de Aceptación HU 6: Buscar datos de un mineral por nombre.

Prueba de Aceptación
HU: 6
Nombre: Probar buscar datos de un mineral por nombre.
Descripción: El usuario introduce el nombre de un mineral y se muestra un formulario con los datos de dicho mineral.
Condiciones de ejecución: Debe de existir el mineral en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Introducir nombre del mineral en el campo de búsqueda que está en la barra de herramientas. Mostrará un formulario con los datos de dicho mineral.
Resultado: Busca correctamente.
Evolución de la prueba: Aceptada.



Tabla 3.41: Prueba de aceptación HU 7: Buscar minerales por criterios de búsqueda.

Prueba de Aceptación
HU: 7
Nombre: Buscar minerales por criterios de búsqueda.
Descripción: El usuario selecciona el o los tipos de criterios para realizar la búsqueda.
Condiciones de ejecución: Debe de existir al menos un mineral en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: seleccionar criterios e introducir los datos en los campos que sean necesarios presionar dar clic en el botón buscar. Si los datos son correctos se mostrará en lista los nombres de los minerales que coinciden con los criterios de búsquedas. Se mostraran mensajes de error cuando: <ul style="list-style-type: none">✓ No seleccionen los tipos de criterios.✓ No se llenen los campos que sean necesarios.
Resultado: Busca correctamente.
Evolución de la prueba: Aceptada.

Tabla 3.42: Prueba de aceptación HU 8: Generar reporte.

Prueba de Aceptación
HU: 8
Nombre: Generar reporte
Descripción: El usuario selecciona en el menú Operaciones, Generar Reporte.
Condiciones de ejecución: Debe de existir al menos un mineral en la BD. Debe estar conectado a la Base de Datos.
Entrada/Pasos ejecución: Al seleccionar generar reporte se mostrará un reporte con la lista de minerales en el que podrá guardar la misma en formato (pdf), e imprimir si así lo desea.
Resultado: Se genera correctamente el reporte.
Evolución de la prueba: Aceptada.



Tabla 3.43: Prueba de aceptación HU 9: Identificación de tipo de rocas ígneas.

Prueba de Aceptación
HU: 9
Nombre: Identificar tipos de Rocas Intrusivas (Plutonitas).
Descripción: El usuario selecciona en la interfaz la opción diagrama, se muestra una interfaz creada para trabajar con diagrama ternario, Rocas intrusivas o Rocas Extrusivas.
Condiciones de ejecución: Haber introducidos los datos de Q, A, P.
Entrada/Pasos ejecución: Se selecciona el diagrama de Rocas Intrusivas. Introduce los datos, da clic sobre el botón graficar. Si la suma de $Q+A+P < 100$ muestra un mensaje de confirmación pidiendo si deseas recalculas. En la parte inferior derecha se muestra el nombre del tipo de roca identificada y los valores de Q, A, P. Se mostrará un mensaje de confirmación: <ul style="list-style-type: none">✓ Cuando al suma de $Q + A + P < 100$. Se mostrará un mensajes de error cuando: <ul style="list-style-type: none">✓ Cuando la suma de los datos de $Q+A+P > 100$.
Resultado: identifica correctamente el tipo de roca.
Evolución de la prueba: Aceptada.