

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 6



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

“Herramienta para la creación de semivariogramas”

Autor:

David González González

Tutor:

Ing. Dagoberto Antonio Suárez Morales

La Habana, 7 de junio de 2013
“Año 55 de la Revolución”

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

David González González

Firma del Autor

Dagoberto Antonio Suárez Morales

Firma del Tutor

DATOS DE CONTACTO

DATOS DE CONTACTO

Nombre del tutor: Dagoberto Antonio Suárez Morales

E-mail: dasuarez@uci.cu

Profesión: Ingeniero en Ciencias Informáticas.

Años de graduado: 5

AGRADECIMIENTOS

AGRADECIMIENTOS

A mi mamá por intentar siempre convertirme en un hombre de bien, por no cansarse nunca conmigo y siempre estar dispuesta a perdonar mis errores. No hubiera encontrado una mejor.

A mi papá por ser mi guía, mi ídolo, mi mejor amigo y apoyarme siempre especialmente en los momentos difíciles por los que he pasado, por estar siempre ahí y ponernos a mi hermana y a mí por encima de todas las cosas y ser el mejor padre del mundo.

A mi hermanita linda, por ser mi amiga, mi confidente, mi principal defensora, mi aliada número uno.

Contigo hasta el fin del mundo, eres la mejor y espero que podamos triunfar juntos.

A mi familia por ser la mejor del mundo en especial a mi tío Pepe, a mi tía Mary, mi tía Nuris por el apoyo de siempre y ayudarme a seguir adelante.

A mi tutor Dagoberto por su inagotable paciencia, regaños y enseñanzas, a Eddy por ayudarme cada vez que lo necesité, a mi oponente Armando y los miembros del tribunal por sus aportes y recomendaciones que han sido de gran ayuda.

A mi amigo de siempre Javier, con el que he compartido los problemas, las mejores fiestas, los buenos y malos momentos. Eres como un hermano.

A mis amigos: Emir, Javier, Yadir, Alejandro, Renier, Arianna, Dailenka, Oscar, Darcy, Heriberto, Eduardo, Carlos, Jesús en general al piquete del 95. En especial a Jorge por empujarme en los momentos que me detuve.

A todos gracias.

DEDICATORIA

DEDICATORIA

*A mis padres por ser mis guías en la vida, sin duda los
mejores.*

RESUMEN

En el Centro de Geoinformática y Señales Digitales (GEYSED), se cuenta con el proyecto Sistema Minero Cubano, en el cual se desarrollan soluciones informáticas para campos de la minería y la geología. Por la necesidad de describir la variabilidad y correlación espacial de una variable regionalizada, distribuida espacialmente a partir de las muestras de un yacimiento mineral, se decidió desarrollar una herramienta geoestadística que facilitara dicha descripción: el semivariograma, este es un gráfico que describe la variabilidad y la correlación espacial de una variable regionalizada distribuida espacialmente. Se definió como objetivo general desarrollar una herramienta para la creación de semivariogramas a partir de las muestras de un yacimiento mineral. Con el fin de cumplir dicho objetivo se realizó la modelación de diagramas que permitieron una mejor comprensión del negocio y facilitaron el proceso de implementación. Se aplicaron un conjunto de pruebas de tipo caja negra y al final del proceso se obtuvo la Herramienta para la Creación de Semivariogramas.

PALABRAS CLAVE

Correlación espacial, geoestadística, minería, muestra, semivariograma, variabilidad espacial, variable regionalizada, yacimiento mineral.

TABLA DE CONTENIDO

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	5
1.1 Conceptos asociados al dominio del problema	5
1.2 Descripción del objeto de estudio	6
1.3 Soluciones existentes	15
1.4 Conclusiones parciales	19
CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN	20
2.1 Metodología de desarrollo	20
2.1.1 Proceso Unificado de Desarrollo de Software (RUP)	20
2.2 Lenguaje de modelado	21
2.3 Herramienta CASE	22
2.3.1 Visual Paradigm 8.0.....	22
2.4 Lenguaje de programación	23
2.5 Framework y librería para el desarrollo	23
2.5.1 Descripción del framework de desarrollo Qt.....	23
2.5.2 Qwt.....	24
2.6 Entorno de desarrollo integrado a utilizar	25
2.7 Sistema Gestor de Base de Datos a utilizar	25
2.7.1 PostgreSQL.....	26
2.8 Conclusiones parciales	27
CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA.....	28
3.1 Dominio del sistema.....	28
3.1.1 Diagrama de clases del Modelo del Dominio	29
3.1.2 Descripción de las clases del modelo de dominio	29
3.1.3 Descripción textual	30
3.2 Requisitos del software	30
3.2.1 Requisitos funcionales.....	30
3.2.2 Requisitos funcionales del sistema	30

TABLA DE CONTENIDO

3.2.3	Requisitos no funcionales del sistema	31
3.3	Definición de los casos de uso del sistema	32
3.3.1	Diagrama de casos de uso del sistema	32
3.3.2	Descripción de los casos de uso.....	33
3.4	Arquitectura y diseño	35
3.4.1	Patrones	36
3.4.2	Patrones arquitectónicos	37
3.4.3	Arquitectura	37
3.4.4	Modelo Vista Controlador	37
3.4.5	Patrones generales de software de asignación de responsabilidades	39
3.5	Modelo de diseño	40
3.5.1	Diagramas de clases del diseño	40
3.6	Diseño de la base de datos.....	45
3.6.1	Diagramas de clases persistentes	45
3.6.2	Modelo físico	46
3.6.3	Descripciones de las tablas de la base de datos.....	46
3.7	Conclusiones Parciales.....	47
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DE LA HERRAMIENTA DESARROLLADA		48
4.1	Arquitectura de información	48
4.2	Implementación	48
4.3	Diagrama de componentes.....	49
4.4	Modelo de despliegue.....	51
4.5	Diagrama de despliegue	52
4.6	Prueba.....	52
4.6.1	Pruebas unitarias o de unidad	52
4.6.2	Técnica de caja negra.....	53
4.6.3	Partición equivalente	53
4.7	Casos de prueba	54
4.8	Resultado de las pruebas	57
4.9	Conclusiones Parciales.....	58

TABLA DE CONTENIDO

CONCLUSIONES	59
RECOMENDACIONES	60
REFERENCIAS BIBLIOGRÁFICAS	61
BIBLIOGRAFÍA	63
GLOSARIO DE TÉRMINOS	65

ÍNDICE DE FIGURAS

Figura 1. Método de los polígonos. Se calcula el área de 7 polígonos.....	7
Figura 2. Método del inverso de la distancia a la potencia alfa. Se calculan las distancias entre los datos y el centro del bloque.....	8
Figura 3. Continuidad espacial de terrenos.....	9
Figura 4. Fórmula del semivariograma.....	10
Figura 5. Muestras de yacimientos, sus matrices de sus semivariogramas y sus semivariogramas experimentales.	12
Figura 6. Ecuación del modelo esférico.	13
Figura 7. Ecuación del modelo exponencial.....	13
Figura 8. Ecuación del modelo lineal o de potencia.	14
Figura 9. Ecuación del modelo gaussiano.....	14
Figura 10. Modelos básicos de ajuste.....	15
Figura 11. Interfaz del software Isatis.....	16
Figura 12. Interfaz de la herramienta Gemcom Minex.....	17
Figura 13. Interfaz del software SGems.....	18
Figura 14. Diagrama de clases del Modelo del Dominio.....	29
Figura 15. Diagrama de casos de uso del sistema propuesto.	32
Figura 16. Diagrama de paquetes del diseño de la aplicación.	36
Figura 17. Representación del funcionamiento del patrón de arquitectura Modelo Vista Controlador.	38
Figura 18. Diagrama de clases del diseño de la solución propuesta.	41
Figura 19. Diagrama de clases de diseño de la vista del sistema.	42
Figura 20. Diagrama de clases de diseño del controlador del sistema.....	43
Figura 21. Diagrama de clases de diseño del modelo del sistema.	44
Figura 22. Diagrama de clases persistentes de la solución propuesta.	45
Figura 23. Modelo físico de la solución propuesta.....	46
Figura 24. Diagrama de componentes de la herramienta desarrollada.	50
Figura 25. Diagrama de componentes del ejecutable de la herramienta desarrollada.....	51
Figura 26. Diagrama de despliegue de la aplicación desarrollada.....	52

ÍNDICE DE TABLAS

ÍNDICE DE TABLAS

Tabla 1. Descripción del caso de uso del sistema "Pintar semivariograma experimental".....	33
Tabla 2. Caso de prueba "Pintar semivariograma experimental".....	54
Tabla 3. Descripción de las variables para el caso de prueba "Pintar semivariograma experimental".....	56
Tabla 4. Matriz de datos del caso de prueba "Pintar semivariograma experimental".....	57

INTRODUCCIÓN

En las ciencias naturales existen diversos problemas que pueden ser clasificados en tres grupos:

- De los que se conoce la ley física que rige el fenómeno, por lo que pueden ser descritos de forma exacta a través de leyes de carácter determinista.
- De los que aun cuando no existe una ley física que los caracterice, se les conoce su ley empírica por lo que pueden ser descritos a través de modelos matemáticos por analogía.
- De los que no se conoce ni su ley física ni su ley empírica, para los que sólo es posible describir el fenómeno en término de estimaciones.

Ejemplo del tercer problema descrito es la estimación de recursos minerales, así como la caracterización de los parámetros porosidad y saturación, que permiten cuantificar las reservas en yacimientos de petróleo. La estimación de minerales como el oro, la plata y el níquel a partir de muestras de un yacimiento mineral, pueden ser citados en el campo de las ciencias naturales. Para la ejecución de estas estimaciones se toman las muestras de los minerales que caracterizan el yacimiento mineral con el objetivo de obtener un conjunto de datos representativos e inferir el valor de las variables en el resto del yacimiento.

La distribución espacial de los recursos minerales es un fenómeno natural aleatorio y es imposible a simple vista determinar cómo están distribuidas las concentraciones de minerales en un área determinada, por lo que es necesario el uso de diferentes métodos que permitan la comprensión del mismo, describiendo su comportamiento y brindando información que permita realizar una estimación de los recursos en una zona dada.

En un principio, para realizar estimaciones de recursos mineros, se utilizaron métodos matemáticos, los cuales no consideran la estructura del cuerpo mineralizado y traen como consecuencia la sobre-estimación y la sub-estimación de los minerales con concentraciones altas y bajas respectivamente. Con el desarrollo de la estadística se comenzaron a utilizar métodos estadísticos para resolver este tipo de problemas, pero estos métodos no tienen en cuenta la ubicación espacial de las muestras y solo se

INTRODUCCIÓN

centran en los valores numéricos de los datos, por lo que se pierde información y los resultados son alejados de la realidad (SIRONVALLE 2000).

A partir de investigaciones desarrolladas a mediados de la década de 1950 sobre la aplicación de la matemática en el cálculo de reservas minerales, el ingeniero francés Georges Matheron desarrolla una serie de trabajos que culmina en su tesis doctoral de 1965 *“Las variables regionalizadas y su estimación”*, surgiendo de esta forma la geoestadística. La geoestadística es definida por Matheron como la ciencia que estudia estadísticamente los fenómenos naturales y cuenta con métodos para lograr la estimación de recursos minerales, entre los que se encuentran la familia de los Kriging y los Cokriging.

Para utilizar estos métodos se tienen en consideración los valores numéricos y la posición espacial de las muestras, así como la descripción de la variabilidad y correlación espacial de las mismas. Para dar solución a dicho problema se utiliza el semivariograma, gráfica que ilustra la variación de los valores entre las muestras y su correlación espacial (GIL and CRUZ 2007).

Realizar el cálculo de semivariogramas de forma manual, para obtener la descripción de la variabilidad y la correlación espacial de las muestras de un yacimiento mineral, es una tarea con un alto grado de complejidad. Entre las razones fundamentales que dificultan este proceso están: la alta complejidad matemática del proceso de construcción del semivariograma y la extensión del mismo, provocando que se incurra en errores durante su realización y además, que los especialistas utilicen mucho tiempo para cumplir con la tarea, pudiendo entorpecer el cumplimiento de algún cronograma de ejecución.

En la Universidad de las Ciencias Informáticas (UCI) se desarrolla, en el centro Geoinformática y Señales Digitales (GEySED), el proyecto Sistema Minero Cubano. Este proyecto tiene como objetivo dotar a la industria minera cubana de un software que permita interpretar geológicamente yacimientos mineros; modelar, estimar y reportar recursos minerales, así como optimizar, diseñar y planificar la actividad minera cubana. En la segunda fase de desarrollo del software que se implementa, Sistema de Análisis y Modelado de Yacimientos Minerales (SYAM), se necesita realizar la estimación de recursos minerales utilizando métodos geoestadísticos, por lo que es necesaria la obtención de la descripción de la variabilidad espacial de las muestras de los yacimientos minerales.

INTRODUCCIÓN

A partir de todo lo anteriormente explicado, se plantea como **problema a resolver** que: El cálculo de la descripción de la variabilidad y la correlación espacial de las muestras de forma manual es un proceso complejo y engorroso para los especialistas geólogos y mineros, provocando la aparición de errores durante el proceso y a su vez una estimación de recursos minerales errónea.

Definiéndose como **objeto de estudio** el proceso de estimación de recursos minerales con métodos geoestadísticos enmarcado en las herramientas informáticas para la construcción de semivariogramas como **campo de acción**.

Determinándose como **objetivo general**: Desarrollar una herramienta informática para la construcción de semivariogramas a partir de las muestras de un yacimiento mineral.

Por lo tanto la **idea a defender** quedaría definida de la siguiente forma: El desarrollo de una herramienta informática para la construcción de semivariogramas permitirá disminuir los errores en el proceso de descripción de la variabilidad y la correlación espacial de las muestras de un yacimiento mineral y, como consecuencia, en la estimación de recursos minerales.

Para darle cumplimiento al objetivo trazado se realizarán las siguientes tareas que guiarán el curso de la investigación.

1. Elaborar el diseño teórico – metodológico de la investigación.
2. Describir los conceptos asociados al dominio del problema.
3. Seleccionar la metodología de desarrollo a utilizar en la investigación.
4. Seleccionar el lenguaje de programación a utilizar en la investigación.
5. Seleccionar el sistema gestor de bases de datos a utilizar en la investigación.
6. Describir tecnologías identificadas para la utilización en la creación de semivariogramas.
7. Especificar los requisitos funcionales de la herramienta que se quiere desarrollar.
8. Elaborar el análisis de la herramienta que se quiere desarrollar.
9. Implementar la herramienta que se desea desarrollar.
10. Diseñar las pruebas a realizar a la herramienta informática obtenida.

INTRODUCCIÓN

A continuación se realizará una breve explicación de los métodos utilizados para llevar a cabo la investigación

Se utilizaron los siguientes métodos teóricos:

Método Analítico-Sintético: Fue utilizado para gestionar la información relacionada con la geoestadística en especial, los diferentes tipos de semivariogramas que se utilizan en los software mineros, después de haber realizado un estudio de la bibliografía encontrada sobre el tema de investigación.

Método Histórico-Lógico: Para la recopilación de información sobre la existencia de herramientas que sean capaces de crear semivariogramas, así como el surgimiento y la evolución de la investigación.

El presente trabajo de diploma constará de cuatro capítulos:

Capítulo 1: Fundamentación teórica de la investigación. En este capítulo se realizó un estudio del estado del arte, centrado en las soluciones existentes. Además se caracterizaron los procesos de estimación de recursos minerales y los conceptos principales asociados a los mismos.

Capítulo 2: Tendencias y tecnologías actuales para el desarrollo de la solución. En este capítulo se describen las principales tecnologías y herramientas definidas para el desarrollo de la solución propuesta como son: la metodología de desarrollo de software, los lenguajes de modelado y de programación, librerías y frameworks existentes, entre otros.

Capítulo 3: Análisis y diseño de la herramienta propuesta. En este capítulo se realiza un análisis de la solución propuesta, describiendo los procesos de la misma y los requerimientos necesarios para su implementación. Se realiza también una descripción del diseño de la herramienta que se desea desarrollar y su arquitectura.

Capítulo 4: Implementación y prueba de la herramienta desarrollada. En este capítulo se describe el proceso de implementación del software que se desarrolla, analizando y describiendo los artefactos generados por la metodología utilizada en la solución de la investigación. También se analizan los resultados de pruebas que se le realizaron a la herramienta para su validación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En este capítulo se presentan los fundamentos teóricos de la investigación que permiten comprender la situación problemática existente en el país. Se exponen, además, algunos conceptos asociados al dominio del problema y se hace un breve reseña de algunas de las soluciones existentes en el mundo.

1.1 Conceptos asociados al dominio del problema

Uno de los conceptos que es necesario conocer para la comprensión de este problema es el de geoestadística. El término geoestadística designa el estudio estadístico de fenómenos naturales, donde un fenómeno natural puede ser caracterizado por la distribución en el espacio de una o más variables llamadas variables regionalizadas, y que tiene por objetivo describir cuantitativamente variables naturales distribuidas en el espacio como por ejemplo la concentración de un mineral en una mina (CASSIRAGA 2007). Matheron fue el primero en utilizar éste término y propone la siguiente definición: La geoestadística es la aplicación del formalismo de las funciones aleatorias al reconocimiento y estimación de fenómenos naturales (CASSIRAGA 2007).

Otro concepto fundamental es el de semivariograma, ya que en su cálculo y graficación consiste principalmente esta investigación. El método más simple de comparar dos valores de muestras es calcular la diferencia entre dos muestras (KRAJEWSKI and GIBBS 2010).

Cuando se comparan un gran número de pares de muestras pueden ocurrir los siguientes casos:

- Algunas diferencias pueden ser positivas y algunas negativas. Por elevación al cuadrado de las diferencias, todas se convierten en positivas.
- Las diferencias son medidas entre muestras a distancias iguales, luego son elevadas al cuadrado y promediadas.

Según A .G. Journel y C. J. Huijbregts, en 1978, el semivariograma se define como la media aritmética de todos los cuadrados de las diferencias entre pares de valores experimentales separados una distancia h .

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Los parámetros del semivariograma caracterizan tres elementos importantes en la variabilidad de un atributo: la discontinuidad en el origen (existencia de efecto pepita), el valor máximo de variabilidad (meseta), y el área de influencia de la correlación.

El efecto pepita (nugget en inglés) es el valor de la función gamma (función que permite construir un semivariograma experimental) en $X = 0$. Describe el nivel de variabilidad entre las muestras cuando la distancia entre ellas es cercana a 0 (DOMINY *et al.* 2001). En principio, puede ocurrir solamente si existen discontinuidades en la función aleatoria. En la práctica, su existencia se debe a la variación espacial que no puede explicar el semivariograma debido a la escala del muestreo. Para un material continuamente variable, el efecto pepita se produce a partir de la contribución de los errores de medición y la variación a distancias mucho menores que el intervalo de muestreo más pequeño (VIERA 2002).

La meseta es el valor sobre el eje Y (valor gamma) en el cual el semivariograma se pone en forma horizontal o plana. Más específicamente, la meseta es la mitad de la desviación media cuadrática en la cual las diferencias entre los valores de las muestras no pueden seguir siendo observadas.

El rango es la distancia (en unidades de campo) sobre la cual las muestras aparecen independientes unas de otras, o la distancia sobre la cual diferencias significativas entre los valores de las muestras no pueden seguir siendo observadas.

1.2 Descripción del objeto de estudio

Con la llegada de una revolución científico técnica en el área de la geología y la minería, ha existido un gran aumento de la extracción y utilización de minerales, para la industria y economía del país. En la actualidad, los recursos minerales presentan altísimos valores en el mercado mundial, convirtiéndose en una gran fuente de lucro la extracción y explotación de los recursos minerales. Por estos motivos ha tomado una gran importancia la exploración y evaluación geológica – económica de yacimientos minerales, por ser de suma importancia para las empresas y compañías geomineras.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

La estimación de recursos minerales es una operación de gran responsabilidad, ya que determina en gran medida el valor de un yacimiento mineral y facilita el trabajo de las compañías y empresas geomineras, asegurando que estas no inviertan tiempo y recursos en la explotación de yacimientos sin valor.

En un principio, en la estimación de recursos minerales se utilizaron métodos matemáticos, entre los que encontramos:

- **La media aritmética:** es un método algo primitivo, sencillo y fácil de usar, donde todos los datos tienen el mismo peso, pero no es un método eficiente cuando existen agrupaciones de datos y presenta problemas en las estimaciones locales porque quedan bloques sin información.
- **El método de los polígonos:** funciona mejor que la media aritmética con las agrupaciones de datos, aunque tampoco es eficiente con las estimaciones locales y es bastante difícil de implementar en tres dimensiones. Asigna a cada punto del espacio la ley del dato más próximo para estimar una zona S y se ponderan las leyes de los datos por el área (o volumen) de influencia (SIRONVALLE 2000).

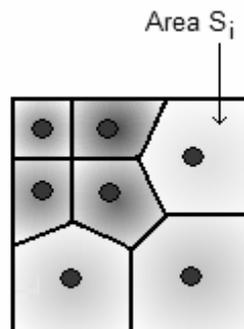


Figura 1. Método de los polígonos. Se calcula el área de 7 polígonos.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

- **El método inverso de la distancia:** es bastante simple y fácil de calcular, se adapta mejor a las estimaciones locales que globales, convirtiéndose en la excepción de los métodos clásicos. Algunas de sus deficiencias son que atribuye demasiado peso a las muestras cercanas al centro de gravedad y que no funciona bien con agrupaciones de datos (Ver Figura 2).

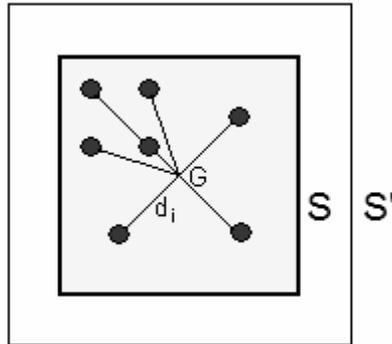


Figura 2. Método del inverso de la distancia a la potencia alfa. Se calculan las distancias entre los datos y el centro del bloque.

En estos métodos se utilizan fundamentalmente valores medios o medias ponderadas para la estimación de bloques definidos convenientemente, por lo que su uso se relaciona con problemas de precisión. Además, la utilización de los métodos clásicos no se recomienda porque, aun cuando permiten cuantificar las reservas a escala global, no son adecuados para la caracterización local de las reservas.

Otros de los métodos utilizados para la estimación de recursos minerales fueron los métodos estadísticos. Sin embargo, estos son puramente numéricos y no tienen en cuenta la variabilidad y correlación espacial de las muestras, lo que provoca estimaciones erróneas. A continuación, en la Figura 3, se muestra como terrenos con valores estadísticos exactamente iguales, pueden diferir en su continuidad espacial (SIRONVALLE 2000).

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

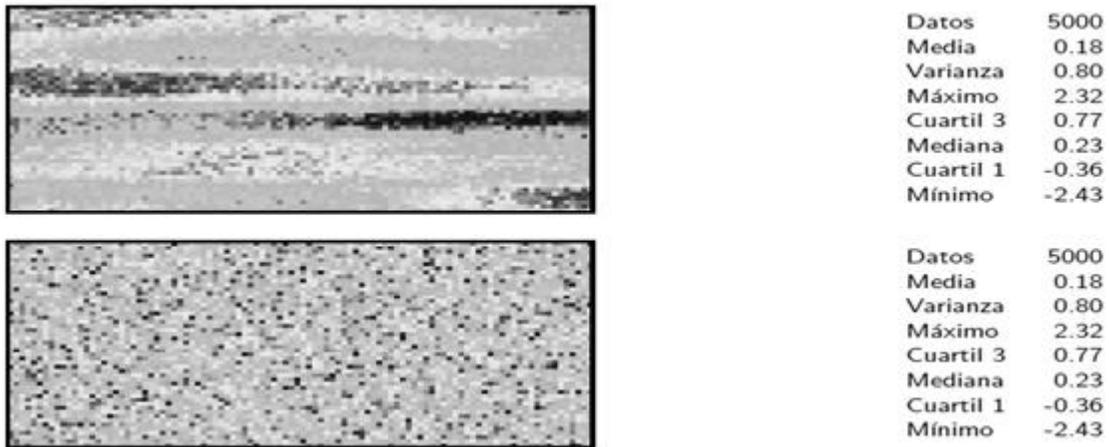


Figura 3. Continuidad espacial de terrenos.

Con el desarrollo de la geoestadística y su aplicación en la rama de la minería y la geología, se desarrollaron métodos muy eficientes a la hora de realizar estimaciones. Los métodos geoestadísticos se fundamentan en el uso de técnicas de regresión, teniendo como premisa fundamental la elaboración de las estimaciones a partir de las características de variabilidad y correlación espacial de los datos originales, utilizando métodos de interpolación (CASSIRAGA 2007).

La geoestadística es el resultado de la aplicación de la Teoría de Funciones Aleatorias al reconocimiento y estimación de fenómenos naturales, fundamentalmente los relacionados con la actividad geólogo minera. Es precisamente en esta actividad donde surge, se desarrolla y consolida en los últimos 35 años como ciencia aplicada, dando respuesta a necesidades prácticas y concretas (GIL and CRUZ 2007).

Entre las principales técnicas usadas por la geoestadística a la hora de la estimación de recursos minerales se encuentra el cálculo y ajuste de semivariogramas. El semivariograma es una función que mide el promedio de las discrepancias cuadráticas entre valores asociados a un atributo perteneciente a pares de puntos separados por una distancia cualquiera. Este permite representar el grado de continuidad o discontinuidad espacial de la variable regionalizada en estudio (TULCANZA 1999).

El semivariograma es una es una función creciente que depende de la distancia entre los puntos que se analizan, ya que los valores tomados en varios puntos son, en promedio, tanto más diferentes cuando

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

más alejados están unos de los otros. De este modo el semivariograma da un contenido preciso a la noción tradicional de zona de influencia de una muestra. El crecimiento más o menos rápido del este representa la manera más o menos rápida con la cual se deteriora la influencia de una muestra sobre las zonas más lejanas del yacimiento (TULCANZA 1999).

Los métodos geoestadísticos requieren conocer el semivariograma para cualquier distancia de separación y cualquier dirección, en la cual es considerada la separación, para lo que se necesita ajustar un modelo matemático al semivariograma experimental (KRAJEWSKI and GIBBS 2010).

Su proceso de creación comienza con la construcción del semivariograma experimental, donde se grafican las distancias de las muestras (nombrada con la letra h) en el eje X , y las diferencias medias cuadráticas promedio entre muestras (nombrada con la letra griega gamma (γ)) es graficada en el eje Y .

Luego se definen los puntos definidos por los pares ordenados de estas variables y se trazan segmentos para unirlos.

$$\gamma(h) = \frac{1}{2N(h)} \sum_{(i,j)|h_{ij}=h} (v_i - v_j)^2$$

Figura 4. Fórmula del semivariograma.

A continuación se nombran un conjunto de propiedades fundamentales del semivariograma:

- **El efecto pepita:** representa la variabilidad de la muestra a distancias pequeñas, donde esta variabilidad es causada por controles geológicos o mineralógicos a pequeñas escalas, y también, por la presencia y magnitud de errores en la ubicación, muestreo, ensayo y representatividad en las tareas de muestreo. Este valor constituye la diferencia entre el origen del semivariograma (0,0) y el valor sobre el eje Y donde el semivariograma intercepta a dicho eje. La pepita no debe ser más

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

de 1/3 del valor de la meseta para lograr interpretaciones con un sentido correcto (KRAJEWSKI and GIBBS 2010).

- **El rango:** es la distancia en la cual las muestras se presentan independientes una de otra, o la distancia a la cual las diferencias significativas entre los valores de las muestras no pueden seguir siendo observadas (KRAJEWSKI and GIBBS 2010). Es descrito como una medida del área de influencia de la muestra, debido a que el mismo es la distancia en alguna dirección sobre la cual los datos son correlacionados. Los rangos comúnmente varían con la dirección en depósitos minerales
- **La meseta:** es el valor sobre el eje Y (valor γ) en el cual el semivariograma se pone en forma

horizontal o plana, por eso se le llama meseta. La meseta es el valor de gamma en el rango del mismo. Más específicamente, la meseta es la mitad de la desviación media cuadrática en la cual las diferencias entre los valores de las muestras no pueden seguir siendo observadas (KRAJEWSKI and GIBBS 2010).

- **La pendiente:** es el cambio en gamma sobre una distancia dada h para el semivariograma en el área entre el origen o pepita y donde este alcanza la meseta en el rango. La pendiente es un indicador de la continuidad de los datos dentro del depósito. Una pendiente baja indica un cambio gradual sobre largas distancias (buena continuidad o rango largo), mientras una pendiente fuerte, empinada o abrupta indica un cambio rápido sobre cortas distancias (continuidad de rango corta o pobre) (KRAJEWSKI and GIBBS 2010).

Dentro de los semivariogramas se encuentran los semivariogramas omnidireccionales, que permiten tener una idea inicial acerca de los parámetros relativos a distancia (incremento y alcance de la distancia de separación) y los direccionales, que permiten determinar la existencia de anisotropía¹ y sus direcciones principales.

¹ Característica de algunas sustancias de variar sus propiedades dependiendo la dirección en que se midan.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

El paso final en el análisis consiste en comparar el semivariograma experimental con modelos conocidos, los cuales pueden ser caracterizados por funciones matemáticas. Este proceso de comparación es mostrado como el modelaje del mismo. A partir de este paso, las inferencias sobre la variabilidad estadística espacial de los datos pueden ser hechas haciendo uso de métodos geoestadísticos (GIL 2008).

El modelo es la función matemática continua que representa correctamente los datos experimentales. El modelo matemático es llamado la función intrínseca debido a que la misma describe el comportamiento espacial de la variable regionalizada dentro de un yacimiento mineral y es un hecho específico de semejante regionalización (KRAJEWSKI and GIBBS 2010).

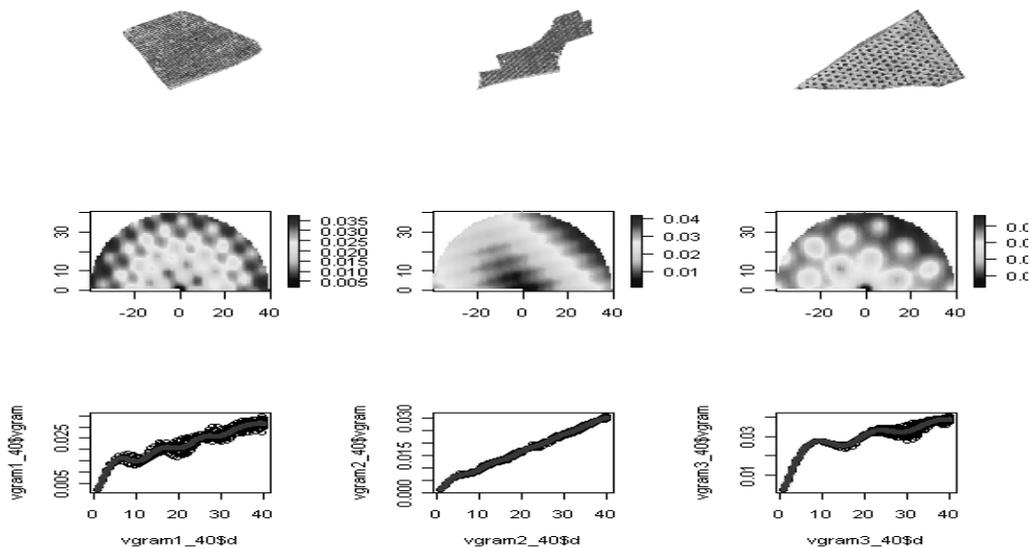


Figura 5. Muestras de yacimientos, sus matrices de sus semivariogramas y sus semivariogramas experimentales.

Dentro de los modelos más comunes se encuentran:

- **Modelo esférico:** Es el más común, con curvas que se incrementan a medida que se aumentan las distancias hasta el rango (límite de influencia). Más allá del rango, la desviación media

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

cuadrática no cambia y la curva queda plana. Los modelos esféricos han sido utilizados para describir diversos tipos de yacimientos como minas de hierro, cobre porfírico, depósitos de plomo-zinc de estrato controlado, bauxita, níquel laterítico, uranio, carbón y fosfatos (CASSIRAGA 2007). El rango, la meseta y el efecto pepita son los parámetros necesarios para esta modelación. La meseta para estos modelos es obtenida usando 2/3 del rango (OLEA 1999). En la Figura 6 el efecto pepita está definido por C , h es la distancia entre las muestras y a es el rango.

$$\gamma(h) = \begin{cases} C\left(\frac{3h}{2a} - \frac{1}{2}\left(\frac{h}{a}\right)^3\right), & 0 \leq h < a \\ C, & a \leq h \end{cases}$$

Figura 6. Ecuación del modelo esférico.

- **Modelo exponencial:** Los modelos tienen curvas que se incrementan a medida que aumenta la distancia. Este modelo alcanza la meseta de forma asintótica (OLEA 1999). Los modelos exponenciales son característicos de datos con relaciones pobres de áreas de influencia o distancias de continuidad muy largas (CASSIRAGA 2007). La meseta para un semivariograma exponencial se obtiene utilizando 1/3 del rango.

$$\gamma(h) = C\left(1 - e^{-\frac{3h}{a}}\right)$$

Figura 7. Ecuación del modelo exponencial.

- **Modelo lineal o de potencia:** Los modelos lineales tienen curvas que son líneas rectas. Los valores gamma se incrementan con una relación lineal a medida que se incrementa la distancia. Los modelos lineales son característicos de datos que varían proporcionalmente con la distancia, como son los datos topográficos. La variación es continua, incrementándose a medida que se

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

incrementa la distancia. Los modelos lineales no tienen meseta o rango por definición (CASSIRAGA 2007).

$$\gamma(h) = C_0 + Ch$$

Figura 8. Ecuación del modelo lineal o de potencia.

- **Modelo de Gauss:** Los modelos gaussianos tienen curvas que se incrementan ligeramente con el aumento de la distancia y también se amplían rápidamente con el aumento de la misma, alcanzando entonces el valor de la meseta en un rango definible. En estos modelos se alcanza la meseta de forma asintótica y tienen forma de parábola en las cercanías al origen de coordenadas (OLEA 1999). Como resultado, los modelos gaussianos tienen curvas que se asemejan y parecen a una "S". Son característicos de datos que tienen tendencia alta, rango corto y un alto grado de continuidad regional. El rango para ellos se obtiene utilizando la raíz cuadrada de tres amplitudes (CASSIRAGA 2007).

$$\gamma(h) = C \left(1 - e^{-3\left(\frac{h}{a}\right)^2} \right)$$

Figura 9. Ecuación del modelo gaussiano.

En la Figura 10 se observan los modelos básicos para el ajuste de semivariogramas que existen.

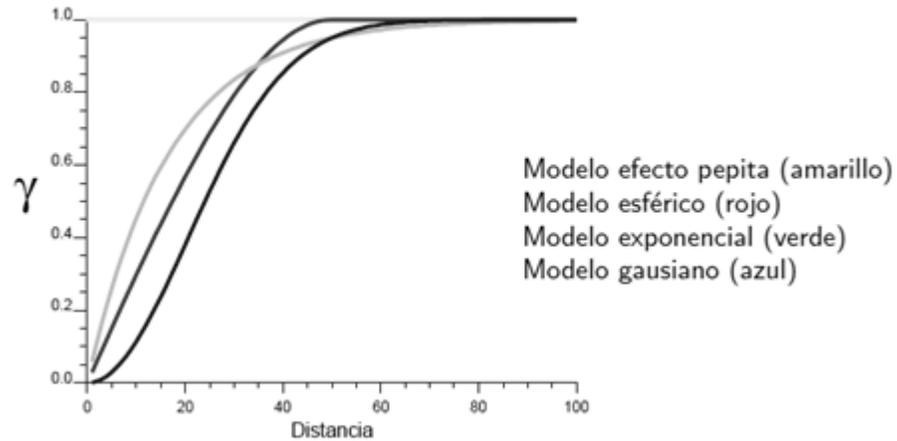


Figura 10. Modelos básicos de ajuste

Con el modelo ajustado concluye el proceso de construcción del semivariograma, quedando todo listo para la estimación de recursos minerales mediante métodos geoestadísticos.

1.3 Soluciones existentes

Con el creciente desarrollo de la informática y la extensión de los métodos geoestadísticos a diversos campos de la ciencia, se han creado las condiciones necesarias para la divulgación y aplicación de estos métodos a un grupo cada vez mayor de problemas. En el mercado se pueden encontrar una gran gama de programas informáticos que usan estos métodos para darle solución a diversos problemas; entre los que se encuentran:

- **ISATIS:** Es una herramienta que contiene todos los métodos geoestadísticos, desde los simples hasta los más avanzados para realizar la estimación de recursos mineros y análisis de riesgos, esta herramienta ofrece un análisis interactivo entre los datos en 2D y 3D con mapas y variogramas, permitiendo el uso para la estimación de métodos como: (kriging simple y ordinario) convirtiendo esta herramienta en una de las más completas que existe (ONLINE 2013)

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Esta es una herramienta inaccesible para el autor por lo cual se hace muy difícil realizar un estudio a fondo de la misma, ya que es privativa. Por esta razón no se lograron obtener experiencias de cómo construir el semivariograma experimental y como realizar el ajuste a modelos matemáticos conocidos. Se obtuvieron experiencias de como una herramienta profesional realiza visualización de este tipo de gráficos y proporcionó al autor de esta investigación una idea de cómo estructurar la interfaz gráfica de la herramienta que se desea desarrollar.

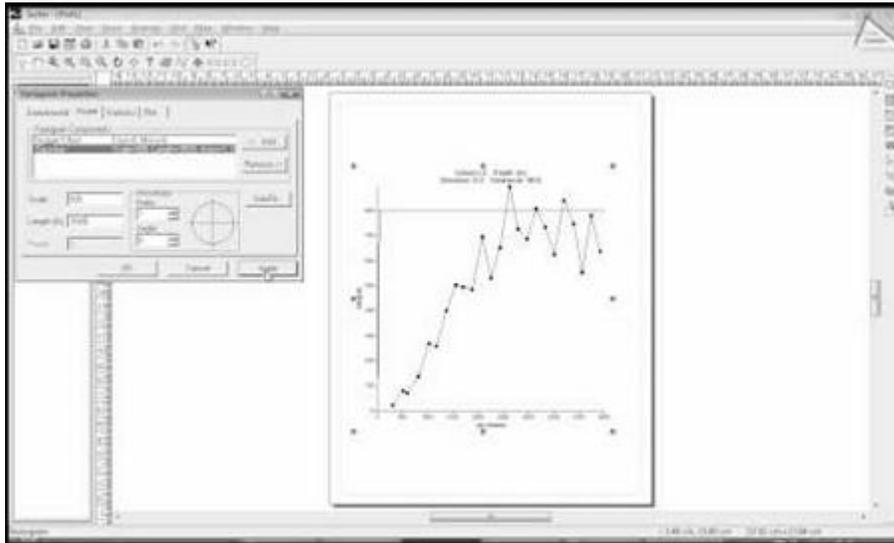


Figura 11. Interfaz del software Isatis.

- **GEMCOM MINEX:** Es el único software integrado de principio a fin que ha sido diseñado específicamente para depósitos estratificados. Esta es una herramienta de estimación estándar para industrias múltiples con una interfaz a GSLIB para Kriging. Permite el modelado de semivariogramas con ajuste de intervalos dinámicos para identificar los mejores para los datos (GEMCOM 2012).

De esta herramienta no se logró obtener información sobre cómo realiza la construcción de semivariogramas experimentales y el ajuste de este a modelos matemáticos conocidos, porque la misma es privativa y es imposible acceder a su código, y la bibliografía que existe no explica cómo

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

se realizan estos procesos. Se utilizó como apoyo para definir el ambiente visual de la herramienta desarrollada.

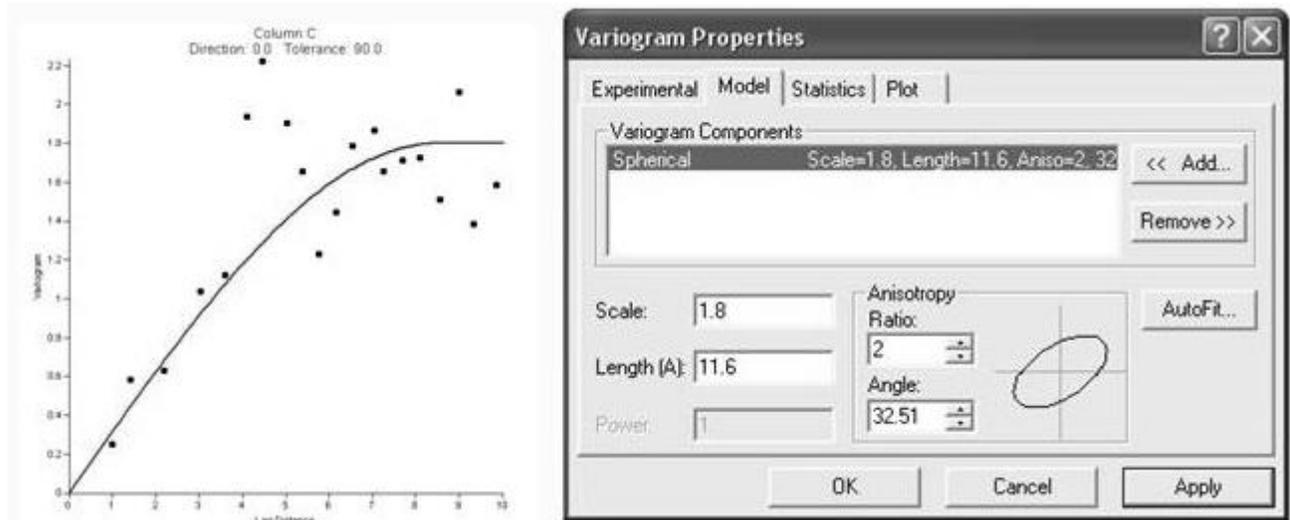


Figura 12. Interfaz de la herramienta Gemcom Minex.

- SGeMS: Software de modelado Geoestadístico de la Tierra (GEMS, por sus siglas en inglés). Es una herramienta de código abierto desarrollada en Stanford para dale solución a problemas de variables regionalizadas. Fue diseñado para proveer a las compañías de un software de modelado amigable para el usuario, ya que ofrece un amplio rango de herramientas geoestadísticas. El software utiliza la librería GSLIB para el análisis geoestadístico (SGEMS 2013).

En su construcción se utilizó GsTL, que no es más que una biblioteca de plantillas para ejecutar rutinas geoestadísticas como son:

- Kriging.
- Multi-variable kriging (co-kriging).
- Secuencial Gaussiano simulación.
- Secuencial indicador de simulación.
- Multi-variable gaussiana secuencial y el indicador de simulación.
- Múltiples puntos de las estadísticas de la simulación.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Esta herramienta no permite el ajuste del semivariograma experimental a más de un modelo al mismo tiempo, lo que dificulta la realización del ajuste a varios modelos. No se obtuvieron experiencias de cómo construir el semivariograma experimental ni de la forma de realizar el ajuste del mismo a modelos matemáticos conocidos; pero sí en cómo construir el semivariograma experimental y en la forma de realizar el ajuste a modelos matemáticos conocidos y su visualización.

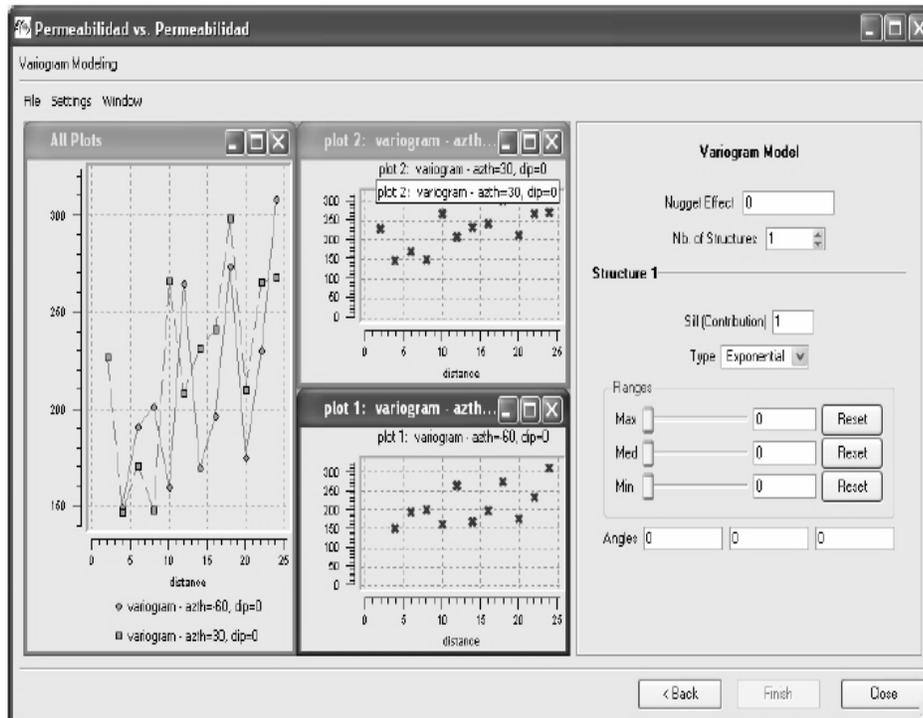


Figura 13. Interfaz del software SGems.

Después del análisis de las soluciones estudiadas, se llegó a la conclusión de que la mayoría de las herramientas que existen para realizar la descripción de la variabilidad y la correlación espacial de las muestras son privativas, lo que las convierte en herramientas inaccesibles para las empresas mineras cubanas. De las soluciones estudiadas se tomaron ideas de cómo realizar las interfaces gráficas de la herramienta que se desarrollará.

1.4 Conclusiones parciales

Después de haber realizado un estudio de los procesos de estimación de recursos minerales, se concluye lo siguiente:

- En la actualidad, para realizar la estimación de recursos mineros con un mínimo de error posible, es necesario utilizar métodos geoestadísticos.
- Para la utilización de métodos geoestadísticos es imprescindible la descripción de la variabilidad y correlación espacial de las muestras, que permitirán la estimación de recursos mineros.
- La mayoría de las herramientas existentes que permiten la descripción de la variabilidad y la correlación espacial de un conjunto de muestras son privativas.
- De las herramientas estudiadas, se obtuvo una idea de cómo realizar la representación visual del semivariograma experimental y del modelo al cual se ajustará.

CAPÍTULO 2. TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

En este capítulo se hace un análisis de la metodología y las herramientas a utilizar en el desarrollo del sistema, haciendo énfasis en aquellas de software libre y código abierto.

2.1 Metodología de desarrollo

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental, que ayuda a los desarrolladores a realizar un nuevo software. Sin el apoyo de una metodología, nunca se llegaría a satisfacer las necesidades de los clientes para los cuales se trabaja. Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. Las metodologías de desarrollo se clasifican en ágiles y pesadas (COCKBURN 2006).

2.1.1 *Proceso Unificado de Desarrollo de Software (RUP)*

El Proceso Unificado de Desarrollo de Software (Rational Unified Process, RUP) es una metodología de software que permite el desarrollo de aplicaciones a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Es una de las metodologías robustas o pesadas, que presenta entre sus características ser un proceso de desarrollo orientado a objetos, utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Tiene tres características fundamentales: es iterativo e incremental, centrado en la arquitectura y dirigido por Casos de Usos.

Dirigido por Casos de Uso: Los requisitos de software que son capturados durante su levantamiento son representados por casos de uso. Estos son los que rigen la planificación del proyecto y los desarrolladores los implementan en términos de clases y subsistemas, verificándose, al finalizar su implementación, que los mismos realizan la funcionalidad esperada (JACOBSON 2000).

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

Centrado en la arquitectura: La arquitectura representa la forma del sistema a través de vistas arquitectónicas. Va madurando en las iteraciones, comenzando con los casos de uso más significativos desde el punto de vista arquitectónico (JACOBSON 2000).

Iterativo e incremental: El Proceso Unificado propone que cada fase se desarrolle en iteraciones, debido a la factibilidad de dividir en partes pequeñas sistemas complejos. Cada una de estas partes es una iteración que resulta en un incremento e involucra actividades de todos los flujos de trabajo (JACOBSON 2000).

El período de vida del software esta particionado en ciclos, cada ciclo consta de cuatro fases: concepción o inicio, elaboración, construcción y transición. Cada vez que termina un ciclo se produce una versión del sistema. Esta metodología es ideal para proyectos cuyos requisitos no son variables y para grandes equipos de desarrollo.

La razón por la que se elige RUP es la posibilidad de contar con la planificación de tareas y artefactos que deben ser generados en un orden y no es necesario que el cliente forme parte del equipo de desarrollo. Otra razón es que para utilizar metodologías ágiles es necesario tener un avanzado nivel de programación que no posee el desarrollador encargado de desarrollar esta herramienta, además de mantener el estándar utilizado en el proyecto Sistema Minero Cubano.

2.2 Lenguaje de modelado

El lenguaje de modelado UML (Lenguaje Unificado de Modelado), creado por Ivar Jacobson, Grady Booch y James Rumbaugh, se ha convertido en un lenguaje estándar de análisis y diseño para aplicaciones informáticas, permite modelar los elementos que integran un producto de software que responde a un enfoque orientado a objeto (LARMAN 2003).

Es el lenguaje de modelado de sistemas informáticos más conocido y utilizado en la actualidad. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software, previo al proceso intensivo de escribir código.

La decisión de utilizar UML como lenguaje de modelado está apoyada en que permitió:

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

- Modelar el sistema que se desarrolló utilizando técnicas orientadas a objetos.
- Especificar todas las decisiones de análisis y diseño, construyéndose modelos precisos y completos.
- Documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Facilitar el proceso de desarrollo mediante la modelación del proceso.

2.3 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Computación (Computer Aided Software Engineer, CASE) se pueden considerar como métodos y técnicas a través de las cuales las personas pueden comprender las capacidades de las computadoras, por medio de programas, procedimientos y su respectiva documentación. Pueden ayudar a la realización de algunas tareas durante el ciclo de vida del desarrollo de software como son: realización del diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores (CASE 2012).

2.3.1 *Visual Paradigm 8.0*

Visual Paradigm for UML es una herramienta que soporta el ciclo de vida completo del desarrollo de software: negocio, requerimiento, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite funcionalidades como:

- Modelar diagramas de cualquier flujo de trabajo, así como generar código y documentación.
- Permite visualizar, en un mismo entorno, la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Permite realizar ingeniería inversa desde sistema gestores de bases de datos a diagramas entidad-relación.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

Visual Paradigm será utilizado porque permitirá realizar la construcción de los diferentes artefactos que genera la metodología de desarrollo de software seleccionada para la investigación, permitiendo la creación de los diagramas que deben ser generados en su ciclo de vida.

2.4 Lenguaje de programación

C++ es un lenguaje de programación diseñado por Bjarne Stroustrup como extensión del lenguaje C a mediados de la década del 80. En la actualidad, existe un estándar ISO C++, al que se han incluido la mayoría de los compiladores modernos y es considerado uno de los lenguajes más potentes, debido a que permite trabajar a alto y bajo nivel.

Entre sus principales características se encuentra el soporte a la programación orientada a objetos, de plantillas o programación genérica. C++ abarca tres paradigmas de la programación: programación estructurada, programación genérica y programación orientada a objetos (LOPEZ 2011).

Se definió C++ como lenguaje de programación a utilizar para que pueda integrarse las funcionalidades desarrolladas al producto que actualmente está implementando el Sistema Minero Cubano. Además, este lenguaje posee una gran cantidad de documentación que permitió al autor de esta investigación un buen aprendizaje del mismo y, al ser un lenguaje multiplataforma, las soluciones desarrolladas en C++ podrán ser utilizadas tanto en Windows como en Linux.

2.5 Framework y librería para el desarrollo

El framework de desarrollo Qt ofrece disímiles facilidades, entre las que se encuentran: permitir realizar aplicaciones con una interfaz gráfica potente, aumentar la productividad en la implementación, liberar al programador de las operaciones básicas de la programación y construir de forma rápida y sencilla de aplicaciones multiplataforma a partir de una misma base de código (THELIN 2007).

2.5.1 Descripción del framework de desarrollo Qt

Las aplicaciones desarrolladas con Qt presentan una buena respuesta y un uso aceptable de la memoria. El desarrollo con este framework es apropiado para proyectos de gran escala, tanto para los de

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

distribución libre como para los comerciales. Las herramientas de Qt tienen un enfoque más eficiente a la hora que los programadores gestionan la memoria. Cuando un objeto es suprimido, todos los dependientes de este se eliminan automáticamente. La biblioteca Qt está concebida como multiplataforma, permitiendo escribir código que se compilará y ejecutará en distintas plataformas, incluyendo Unix, Linux, FreeBSD o incluso Windows (PROJECT 2012).

Qt trabaja sobre las plataformas más importantes, El API de la biblioteca tiene métodos para acceder a bases de datos mediante sentencias SQL, así como gestión de hilos, uso de XML, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros para el manejo de ficheros, además de estructuras de datos tradicionales (PROJECT 2012).

El framework Qt permitirá (PROJECT 2012):

- Desarrollar una herramienta multiplataforma con un sólo código fuente.
- Diseñar interfaces gráficas con Qt Designer.
- Compilar el sistema en diferentes plataformas utilizando qmake.
- Obtener una documentación muy amplia con ejemplos utilizando Qt Assistant.

2.5.2 Qwt

La biblioteca Qwt contiene componentes GUI, widgets y clases que son útiles para programas con un perfil técnico, además de un marco para gráficos 2D que ofrece escalas, deslizadores, diales, brújulas, termómetros, ruedas y botones para controlar o mostrar valores, matrices o rangos de tipo *double*. El poder real de Qwt está en la capacidad de graficar distintos tipos de diagramas entre los que se encuentran: diagramas de dispersión, gráficos de curvas, histogramas con o sin líneas de contorno. Esta librería permitió la realización de los diferentes diagramas que se crearon en la herramienta que se desarrolló (THELIN 2007).

2.6 Entorno de desarrollo integrado a utilizar

QtCreator es un entorno de desarrollo integrado (IDE por sus siglas en inglés), que permite una fácil integración entre el marco de trabajo y el lenguaje de programación, lo que facilita la implementación de la solución resultante de la siguiente investigación. Fue creado para desarrollar aplicaciones con el framework de desarrollo Qt, que está diseñado para desarrollar aplicaciones e interfaces de usuario y desplegarlas en sistemas operativos de escritorios y móviles. También permite que un equipo de desarrollo pueda compartir un proyecto a través de diferentes plataformas como son (Microsoft Windows ®, Mac ® OS X, y Linux ®) (Nokia, 2009).

Cuenta con un editor de texto con autocompletado, que agiliza el proceso de implementación, posee un sistema de depuración que facilita el proceso de detección de errores, un diseñador de interfaces gráficas para realizar diseños sin una alta complejidad, soporta diferentes lenguajes como son: C#.NET Lenguajes (Mono), Python: PyQt y PySide, Ada, Pascal, Perl y PHP. Estas características lo convierten en uno de los IDE más potentes que existen en la actualidad (PROJECT 2012).

2.7 Sistema Gestor de Base de Datos a utilizar

El software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista se denomina **sistema de gestión de bases de datos** (SGBD). (García, 1999)

Un SGDB 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).

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

2.7.1 PostgreSQL

PostgreSQL es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD. Como muchos otros proyectos libres Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (IBARRA and FORES 2008).

Algunas de las principales características con las que cuenta PostgreSQL son según (IBARRA and FORES 2008):

- Permite aproximar los datos a un modelo objeto - relacional y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, optimización de consultas, herencia, y arreglos.
- Las tablas pueden ser configuradas para heredar características de una tabla padre. Los datos son compartidos entre las tablas padre e hija(s). Las tuplas insertadas o eliminadas en la tabla hija serán insertadas o eliminadas en la tabla padre respectivamente.
- Soportará operadores funcionales métodos de acceso y tipos de datos definidos por el usuario.
- Soportar la integridad referencial la cual es utilizada para garantizar la validez de los datos de la base de datos.

Se selecciona PostgreSQL debido a que es un sistema gestor de bases de datos libre y multiplataforma, que puede ser utilizado en cualquier sistema operativo. Además, posee una extensión para el tratamiento de la información espacial, que puede ser considerado para su utilización en el futuro de esta investigación, y permitirá una mejor integración con la solución desarrollada en el proyecto Sistema Minero Cubano.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE LA SOLUCIÓN

2.8 Conclusiones parciales

Después de realizado un estudio de las herramientas y tecnologías que se utilizaron para el desarrollo de la herramienta se concluye lo siguiente:

- Las herramientas seleccionadas para desarrollar la solución propuesta cumplen con las políticas de soberanía tecnológica que impulsa la universidad y el país.
- Se realizó una selección de las versiones recientes de las herramientas con que se trabajará.
- Se tuvo en cuenta para la selección de las mismas, la futura integración de la presente investigación con el trabajo que se realiza en el proyecto Sistema Minero Cubano.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

En este capítulo se realiza el análisis y diseño de la propuesta de solución de la Herramienta para la Creación de Semivariogramas. En este se recogen los artefactos generados en los flujos de requisitos y diseño de la metodología RUP, como metodología que dirige el desarrollo de esta solución. Se aborda, además, una descripción de la arquitectura utilizada en la implementación de esta herramienta informática.

3.1 Dominio del sistema

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales (término utilizado en la primera edición del libro (LARMAN 2003)), modelo de objetos del dominio y modelos de objetos de análisis.

El modelo de dominio, es un subconjunto del modelo de negocio y se utiliza cuando los procesos del negocio no pueden ser definidos con claridad. Este recoge los tipos de objetos más importantes que existen en el sistema, y tiene por objetivo ayudar a comprender mejor los conceptos que utilizan los usuarios.

3.1.1 Diagrama de clases del Modelo del Dominio

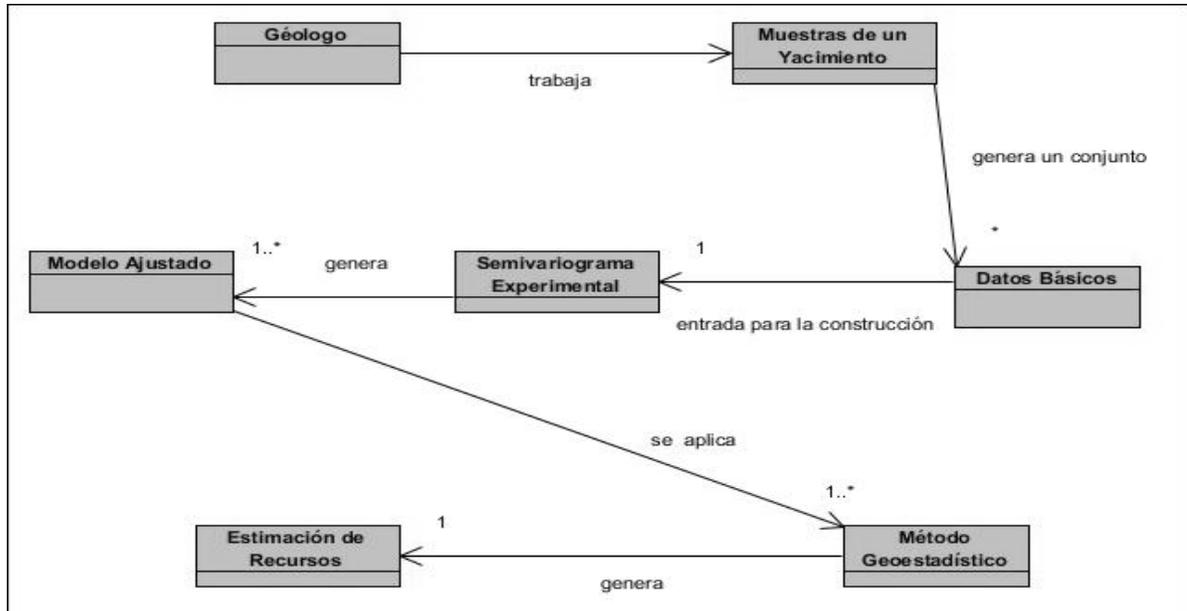


Figura 14. Diagrama de clases del Modelo del Dominio.

3.1.2 Descripción de las clases del modelo de dominio

A continuación se realizará una descripción de las clases del modelo de dominio.

- **Geólogo:** Profesional de la rama de la minería y la geología encargado de trabajar con la aplicación.
- **Muestras de un Yacimiento:** Son el conjunto de muestras tomadas en el yacimiento que se desea estimar.
- **Datos básicos:** Datos obtenidos a partir de las muestras tomadas en el área que se desea realizar las estimaciones.
- **Semivariograma experimental:** Gráfico construido a partir de las muestras de un yacimiento mineral, que representa la mitad de la desviación media cuadrática, utilizando las distancias a las que se tomaron las muestras.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

- **Modelo Ajustado:** Modelos a los cuales se ajusta el semivariograma experimental; a partir de la información que brinda se pueden utilizar métodos geoestadísticos.
- **Métodos geoestadísticos:** Método que se aplica para lograr la estimación de recursos.
- **Estimación de Recursos:** Proceso que permite conocer el aproximado de los recursos minerales que existen en un área.

3.1.3 Descripción textual

Un geólogo trabaja con las muestras de un yacimiento mineral, donde cada muestra genera un conjunto de datos básicos. Estos consisten en los datos de entrada para construir un semivariograma experimental, el cual debe ser ajustado a un modelo matemático conocido, con el objetivo de aplicarle un método geoestadístico para lograr la estimación de los recursos minerales del yacimiento muestreado y lograr su explotación con extrayendo los elementos con mayor precisión.

3.2 Requisitos del software

Requisito, condición o capacidad que tiene que ser alcanzada por un sistema para satisfacer un contrato estándar u otro documento impuesto formalmente. Define qué es lo que el sistema debe hacer, para lo que se identifican las funcionalidades requeridas y las restricciones que se imponen. Se clasifican en funcionales y no funcionales (PRESSMAN 2002).

3.2.1 Requisitos funcionales

Son declaraciones de las funcionalidades que debe proporcionar el sistema, cómo debe reaccionar a entradas particulares y su comportamiento en este tipo de situaciones. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (SOMERVILLE 2005).

3.2.2 Requisitos funcionales del sistema

RF_1 El sistema debe ser capaz de cargar datos de la base de datos.

RF_2 El sistema debe ser capaz de graficar semivariogramas experimentales.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

RF_3 El sistema debe ser capaz de graficar semivariogramas esféricos.

RF_4 El sistema debe ser capaz de graficar semivariogramas lineales.

RF_5 El sistema debe ser capaz de graficar semivariogramas pepita.

RF_6 El sistema debe ser capaz de graficar semivariogramas de gauss.

3.2.3 Requisitos no funcionales del sistema

Los requisitos no funcionales especifican o restringen las propiedades emergentes del sistema. Pueden especificar el rendimiento, la protección, disponibilidad, interfaces de usuario, mantenimiento, portabilidad y otras cualidades del sistema que se desarrolla (SOMERVILLE 2005).

Usabilidad

La herramienta debe presentar una interfaz amigable. Un usuario debe poder construir un semivariograma y de poder ajustarlo sin necesidad de tener un gran conocimiento de informática, permitiendo que el sistema pueda ser utilizado sin ningún problema.

Portabilidad

El sistema debe ser multiplataforma, permitiendo a los usuarios utilizar el sistema en las principales plataformas como son Windows y Linux.

Interfaces de hardware

Para las PCs clientes:

Al menos 512MB de RAM.

Para los servidores:

El Servidor de BD tenga como mínimo 2GB de RAM con al menos 10 GB libres en el disco duro.

Procesador 2.4 GHz como mínimo.

Interfaces de software

Se requiere como gestor de bases de datos PostgreSQL 9.1.

3.3 Definición de los casos de uso del sistema

Los casos de uso son una descripción de los pasos o las actividades que deberán realizarse, para llevar a cabo los diferentes procesos. Los personajes o entidades que participan en el caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas (JACOBSON *et al.* 1992).

3.3.1 Diagrama de casos de uso del sistema

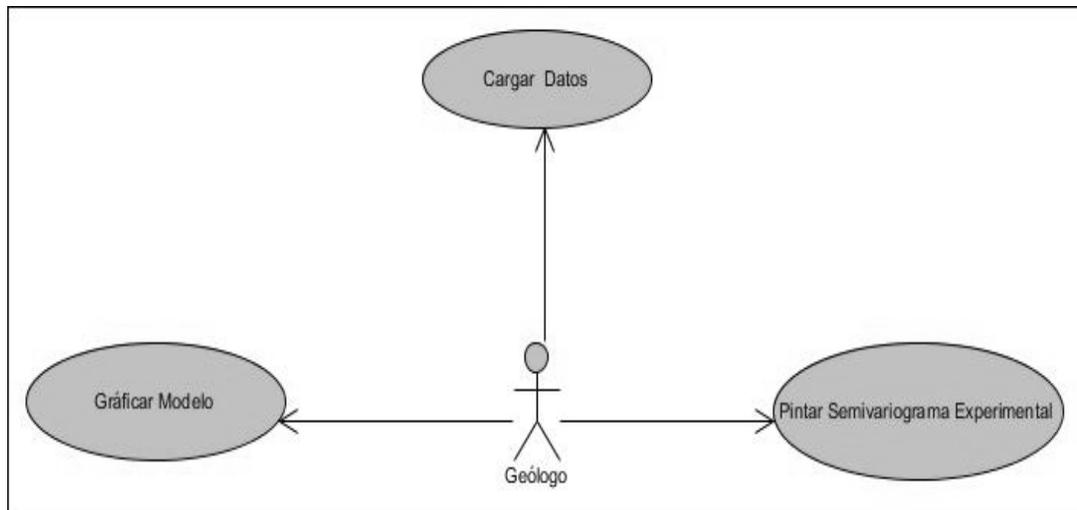


Figura 15. Diagrama de casos de uso del sistema propuesto.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

3.3.2 Descripción de los casos de uso.

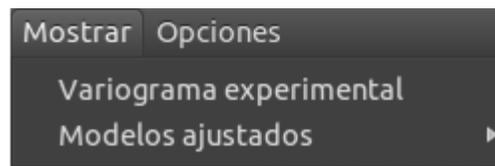
Tabla 1. Descripción del caso de uso del sistema "Pintar semivariograma experimental".

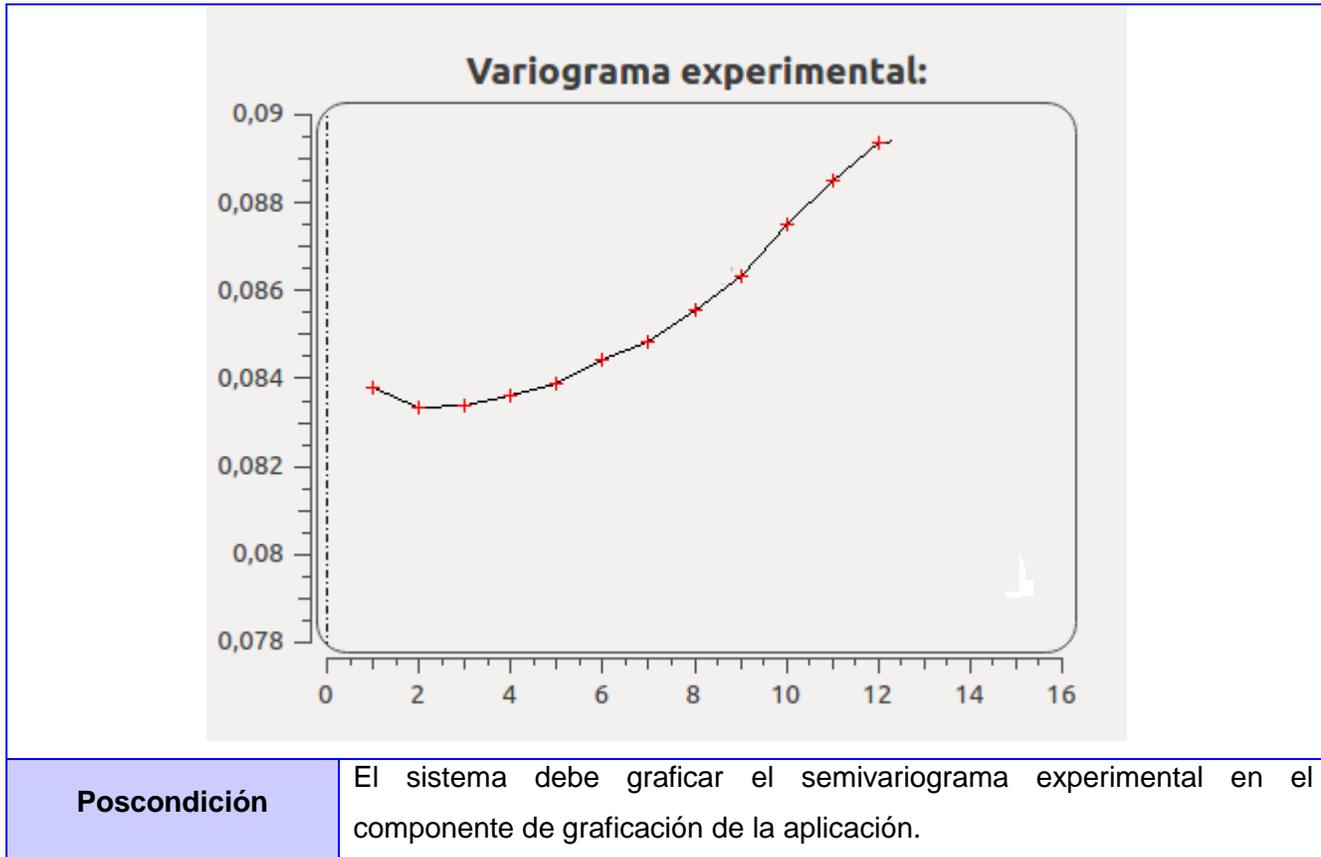
Caso de uso	Pintar semivariograma experimental	
Actores	Geólogo	
Resumen	El caso de uso se inicia cuando el geólogo selecciona la opción Semivariograma experimental del menú Mostrar. El sistema muestra una interfaz para que el usuario entre los parámetros a partir de los cuales se construirá el semivariograma experimental	
Precondiciones	Caso de uso Cargar datos.	
Referencias	RF#2	
Prioridad	Alta	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El caso de uso inicia cuando el geólogo selecciona la opción Semivariograma Experimental del menú Mostrar de la aplicación.	2. El sistema muestra una interfaz para que el usuario entre el conjunto de parámetros necesarios para construir el semivariograma experimental y las opciones: <ul style="list-style-type: none"> • Aceptar • Cancelar 	
3. El geólogo entra los datos requeridos por el sistema y selecciona la opción Aceptar.	4. El sistema valida los datos entrados por el usuario y grafica el semivariograma experimental. Finaliza el caso de uso.	
Flujo Alternativo "Cancelar"		
3. a El geólogo selecciona la opción Cancelar	4. a El sistema cierra la interfaz mostrada para entrar los parámetros y finaliza el caso	

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

	de uso.
Flujo Alternativo "Errores en los datos"	
3. b El geólogo entra los datos requeridos por el sistema y selecciona la opción Aceptar.	4. b El sistema valida los datos y vuelve al paso 3 del flujo normal de eventos.

Prototipo de Interfaz





3.4 Arquitectura y diseño

El modelo de diseño es un modelo físico y concreto, debe ser mantenido durante todo el ciclo de vida del software. Este modelo le da forma al sistema, es un plano de la implementación. El modelo de diseño es una realización del diseño del sistema.

Con este modelo se logra (GAMMA *et al.* 1995):

- Adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia y tecnologías de interfaz de usuario.
- Crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos o subsistemas individuales, interfaces y clases.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

- Descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo.
- Capturar las interfaces entre los subsistemas antes en el ciclo de vida del software, lo cual es muy útil cuando utilizamos interfaces como elementos de sincronización entre diferentes equipos de desarrollo.

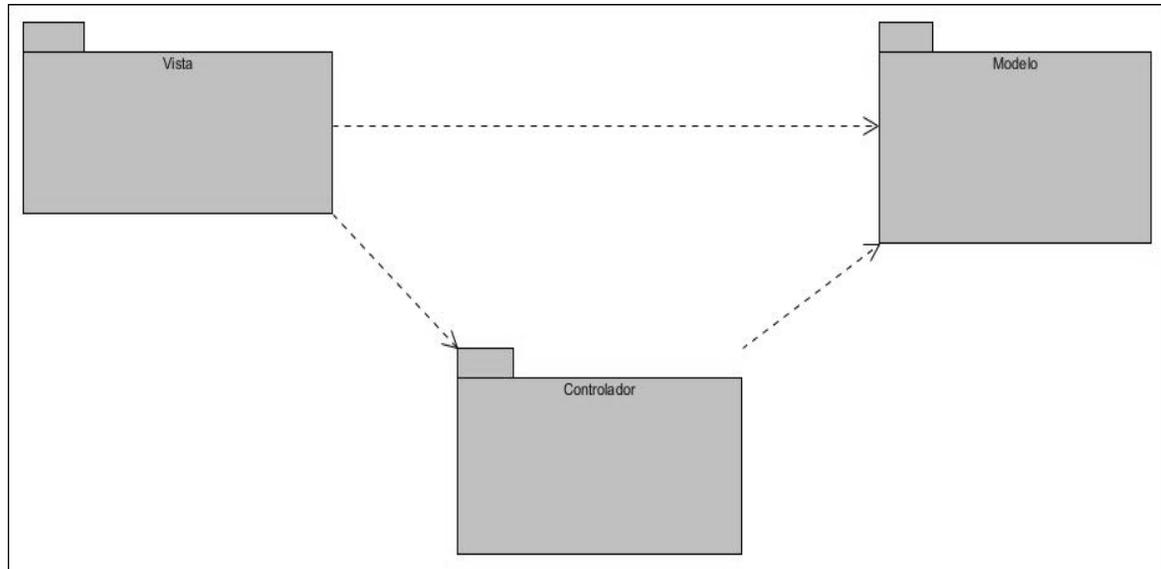


Figura 16. Diagrama de paquetes del diseño de la aplicación.

3.4.1 Patrones

Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Este define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones.

Los patrones se categorizan según la escala o nivel de abstracción, sin embargo cada una de las categorías de patrones define un mismo nivel de abstracción o escala de aplicabilidad (PRESSMAN 2006).

3.4.2 Patrones arquitectónicos

Un estilo arquitectónico, a veces denominado un patrón arquitectónico, es un conjunto de principios. Este mejora la separación y promueve la reutilización del diseño, aportando soluciones a los problemas que se repiten con frecuencia. Más específicamente, un estilo arquitectónico determina el vocabulario de los componentes y conectores que se pueden utilizar en casos de ese estilo, junto con un conjunto de restricciones sobre cómo pueden ser combinados. Estos pueden incluir restricciones topológicas en las descripciones arquitectónicas (GARLAN 1994).

3.4.3 Arquitectura

La arquitectura de una aplicación es la vista conceptual de su estructura. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según como está distribuido este código (CORNEJO 2006).

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas de software complejos como los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración. Para solventarlos se ha generalizado la división de las aplicaciones en capas: una capa que sirve para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y una interfaz gráfica que facilite al usuario el uso del sistema.

3.4.4 Modelo Vista Controlador

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software, usado principalmente en aplicaciones que manejan una gran cantidad de datos y transacciones complejas, donde se requiere una mejor separación de los conceptos, para que el desarrollo sea estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en tres capas: Modelo, Vista y Controlador.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

Modelo: Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros, que puestos en el contexto del sistema proveen de información al usuario, o a la aplicación misma.

Vista: Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En las aplicaciones de escritorio son las interfaces usuarios del sistema.

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

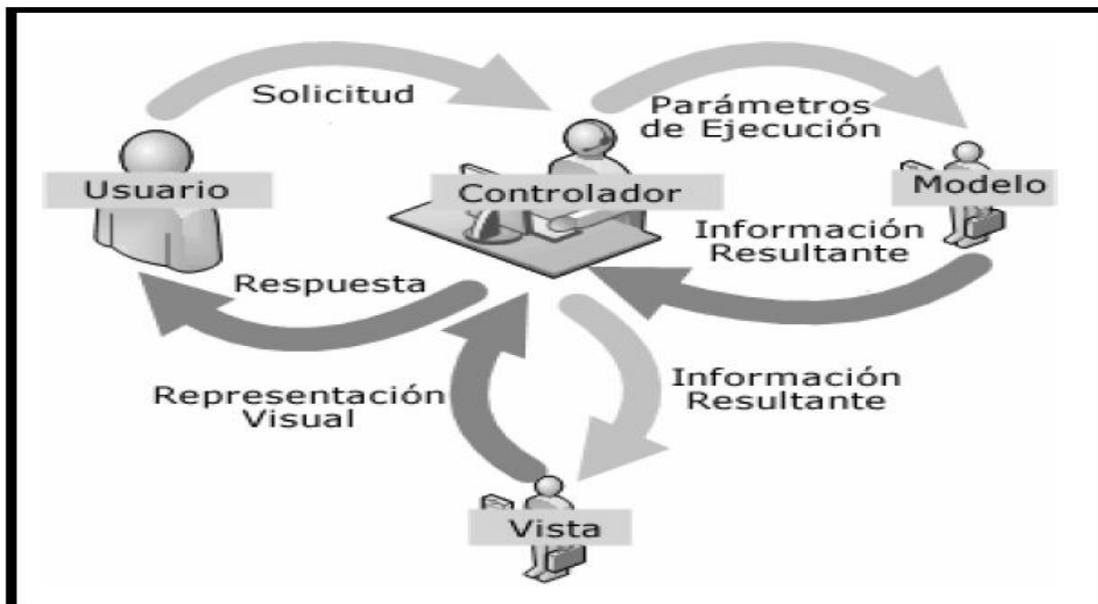


Figura 17. Representación del funcionamiento del patrón de arquitectura Modelo Vista Controlador.

Una vez concluido el estudio acerca de la aplicación de dicha arquitectura y los resultados de dicho ejercicio se definen una serie de ventajas que se listan a continuación:

- La separación del Modelo de la Vista.
- Es mucho más sencillo agregar múltiples representaciones de los datos o información.
- Facilita agregar nuevos tipos de datos según sean requeridos por la aplicación, ya que son independientes del funcionamiento de otras capas.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.
- Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.
- Permite el escalamiento de la aplicación en caso de ser requerido.

3.4.5 Patrones generales de software de asignación de responsabilidades

Se pueden definir como: soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan. Los patrones de diseño no son fáciles de entender, pero una vez entendido su funcionamiento, los diseños serán mucho más flexibles, modulares y reutilizables (GARCÍA 2005).

Se definió que era necesaria la aplicación de patrones GRASP (LARMAN 2003), los cuales se relacionan a continuación:

- **Experto:** Asignar una responsabilidad al experto en información, es decir la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón es usado en todas las clases de la aplicación porque la responsabilidad de realizar una funcionalidad específica se le dio a la clase que era experta en la información necesaria para realizar dicha funcionalidad.
- **Creador:** Asignarle a la clase B la responsabilidad de crear una instancia de clase A. Como ejemplos del uso de este patrón encontramos la clase *Variograma* la cual es la encargada de crear instancias de las clases *Conexión*, *Recta3D*, *PuntoVariograma* y *Muestra*.
- **Alta Cohesión:** Cada elemento del diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos, y auto-identificables. Este patrón es utilizado por las clases: *ModeloEsferico*, *ModeloLineal*, *ModeloExponencial* y *ModeloGauss*. Estas clases presentan funcionalidades estrechamente relacionadas y son utilizadas con un único objetivo en el sistema.
- **Bajo Acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. Este patrón es usado entre las clases *Variograma* y las clases *ModeloEsférico*, *ModeloExponencial*, *ModeloLineal* y *ModeloGauss*, dependiendo cada una de ellas de la menor cantidad de clases posibles, permitiendo, si existen cambios en otras clases, una menor afectación.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

- **Controlador (Controller):** Asigna la responsabilidad del manejo de mensajes, eventos o funcionalidades del sistema a una clase. En el caso de la presente investigación, la clase *Variograma* es la clase controladora del sistema.

3.5 Modelo de diseño

El modelo de diseño es planteado como un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación (LARMAN 2003).

3.5.1 Diagramas de clases del diseño

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y las relaciones entre ellas. Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. Cuando se crea un diagrama de clases, se está modelando una parte de los elementos y relaciones que configuran la vista de diseño del sistema (LARMAN 2003).

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

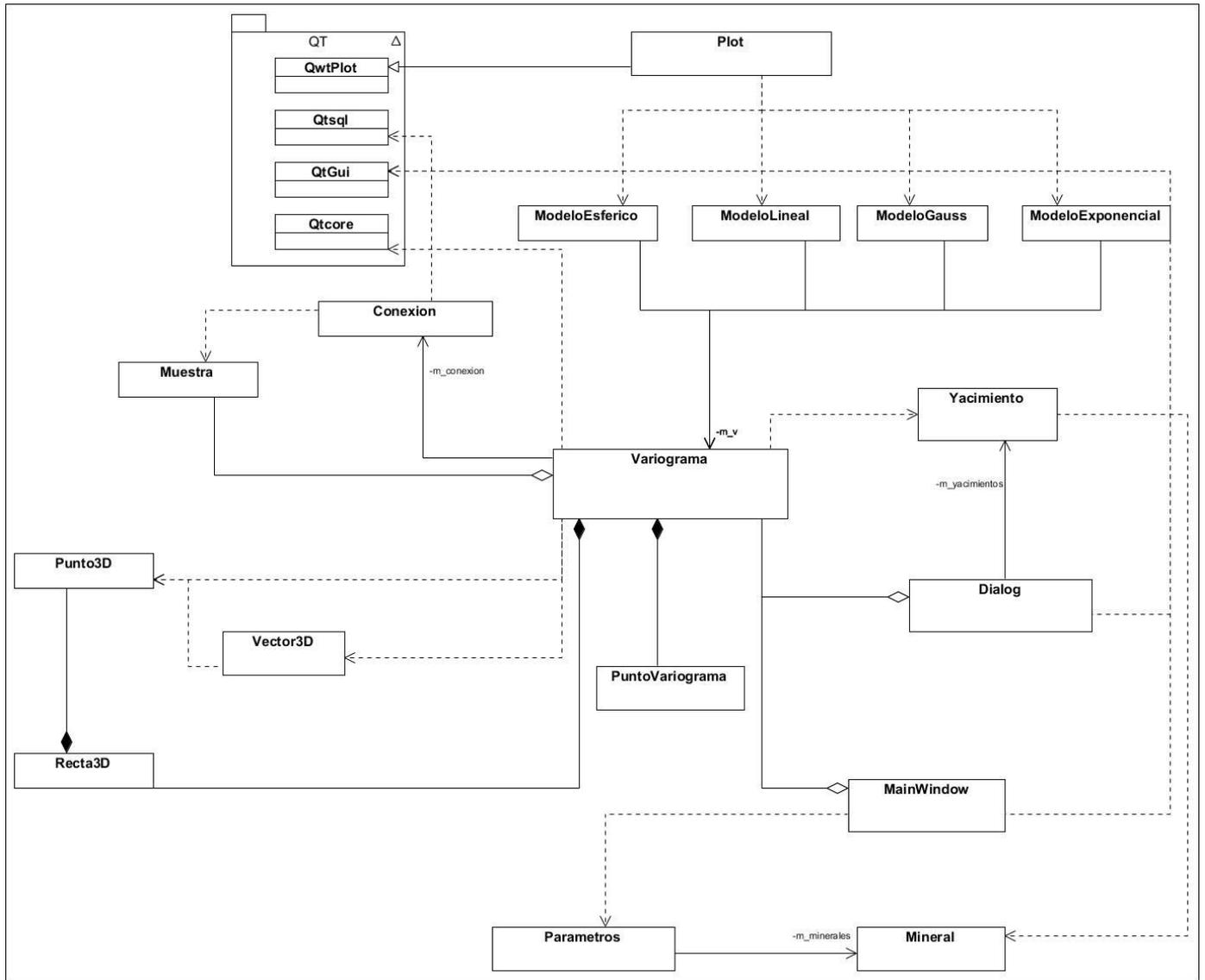


Figura 18. Diagrama de clases del diseño de la solución propuesta.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

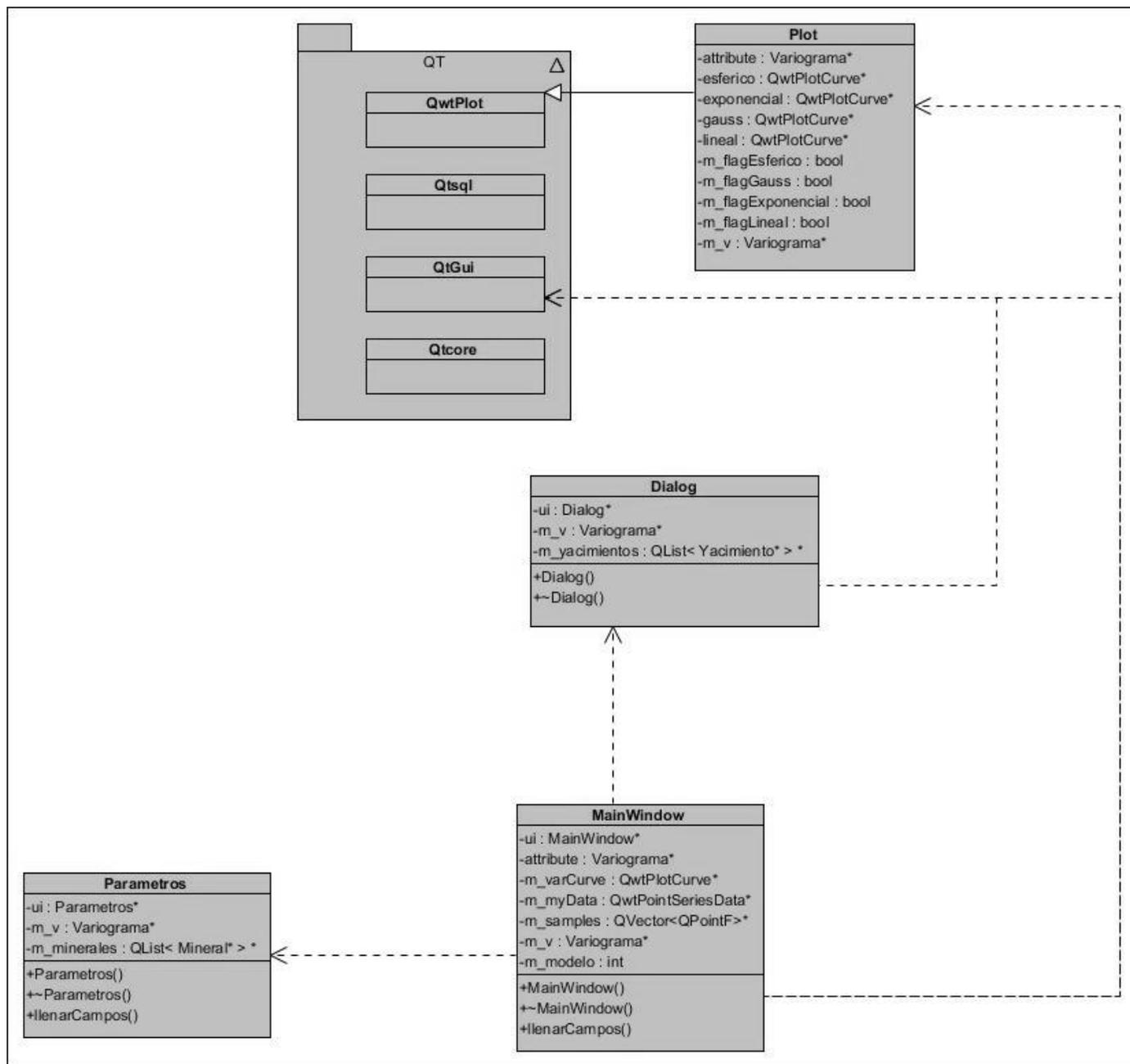


Figura 19. Diagrama de clases de diseño de la vista del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

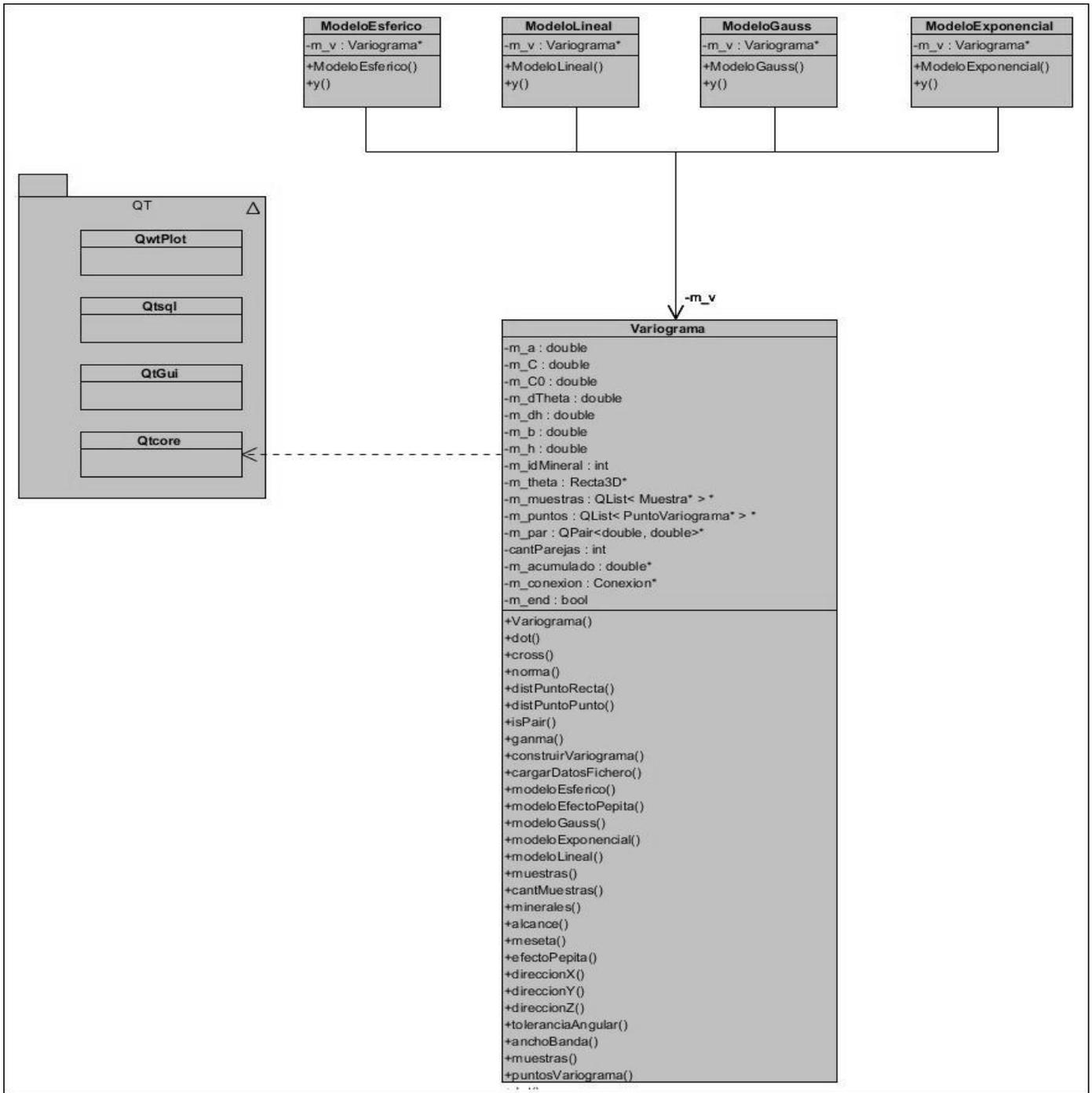


Figura 20. Diagrama de clases de diseño del controlador del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

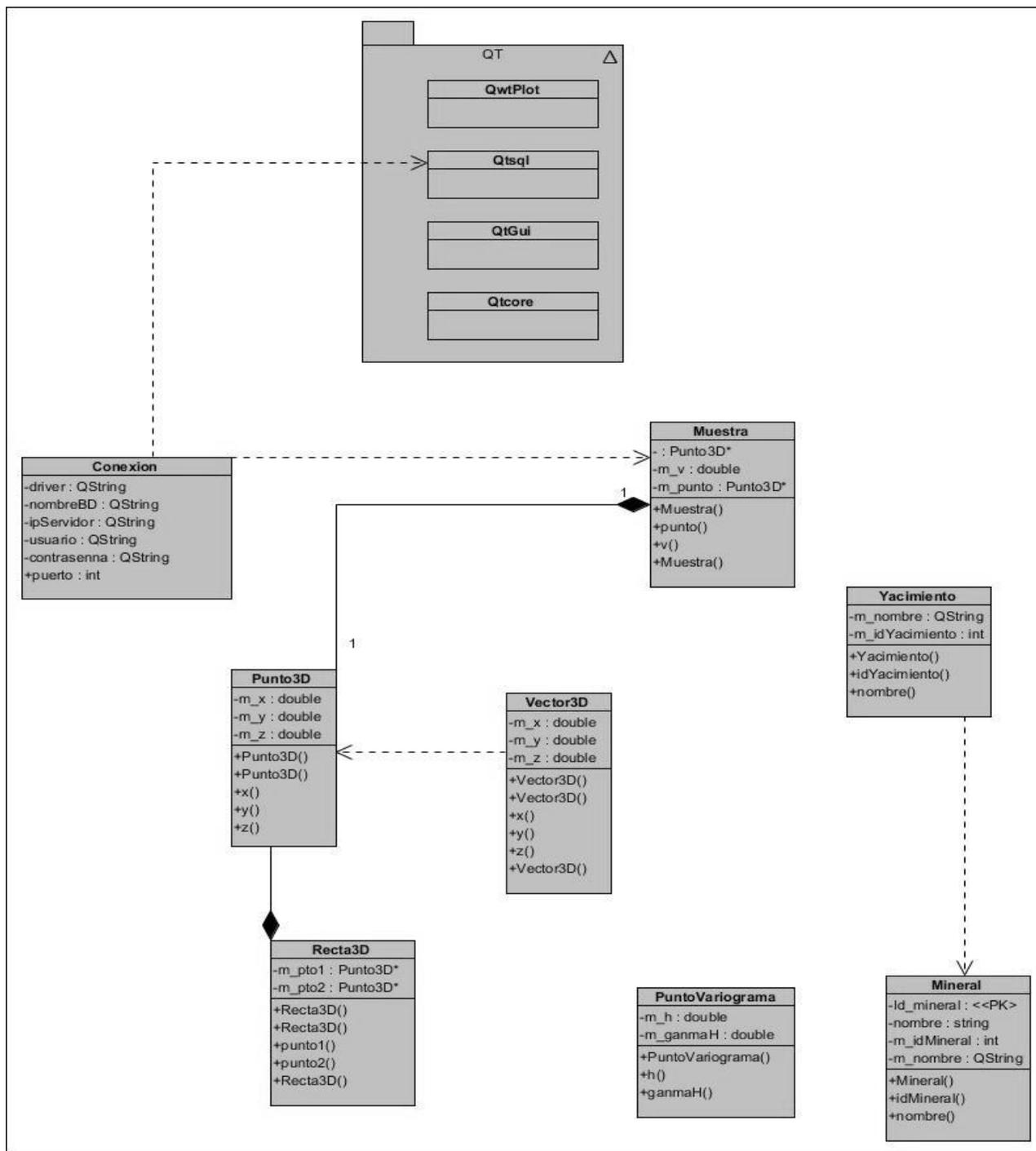


Figura 21. Diagrama de clases de diseño del modelo del sistema.

3.6 Diseño de la base de datos

El diseño de la base de datos es un paso crucial en el desarrollo de la herramienta que dará solución al problema a resolver de la presente investigación. Una correcta selección de las tablas, de la información que contendrá cada una y sus relaciones, y tener en cuenta las prácticas adecuadas de normalización, proporcionarán un rendimiento del sistema adecuado a las exigencias del cliente.

Las bases de datos necesitan de una definición de su estructura que le permitan almacenar datos, reconocer el contenido, y recuperar la información. A partir de las clases del diseño y la realización del diagrama del diseño se realiza el diseño de la base de datos.

3.6.1 Diagramas de clases persistentes

Todas las clases identificadas antes de adentrarse en el diseño no tienen que ser necesariamente persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Lo contrario son las clases temporales que son manejadas y almacenadas por el sistema en tiempo de ejecución por lo que dejan de existir cuando termina el programa. En la siguiente figura se muestra el diagrama de clases persistentes correspondientes al sistema planteado.

Los diagramas de clases también pueden contener a paquetes o subsistemas, que se usan para agrupar elementos del modelo en partes más grandes (por ejemplo, paquetes que a su vez contienen a varios diagramas de clases) (PRESSMAN 2005).

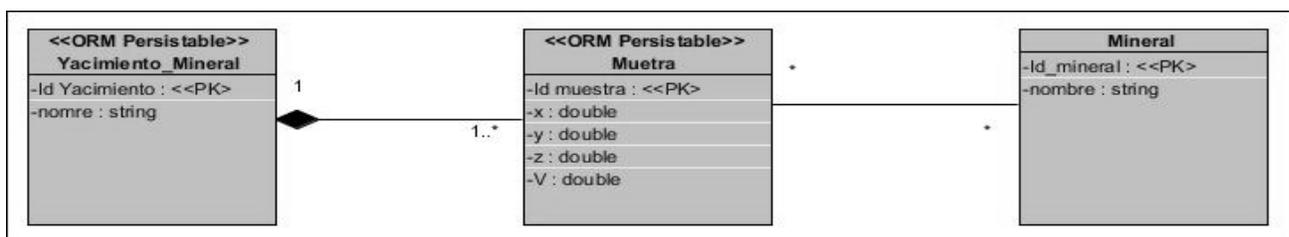


Figura 22. Diagrama de clases persistentes de la solución propuesta.

3.6.2 Modelo físico

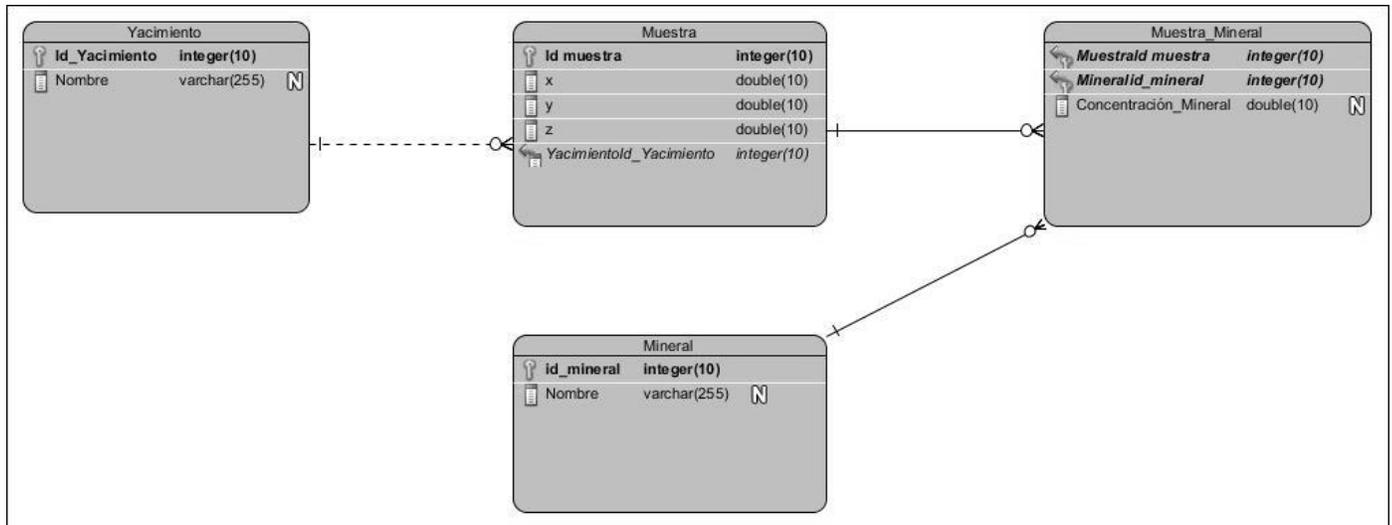


Figura 23. Modelo físico de la solución propuesta.

3.6.3 Descripciones de las tablas de la base de datos.

- **Yacimiento:** Contiene los yacimientos que se cargan en la aplicación. Presenta dos campos: *Id_Yacimiento*, que es un entero consecutivo y llave primaria de la tabla y *Nombre*, que es una cadena.
- **Muestra:** Contiene el conjunto de muestras relacionadas con cada yacimiento. Cuenta con *Id_Muestra*, que es un entero consecutivo y llave primaria de la tabla, X, Y, Z, de tipo doble que representan las coordenadas espaciales de la muestra e *Id_Yacimiento* como llave foránea.
- **Mineral:** Almacena los minerales presentes en las muestras de un yacimiento. Posee como atributos a *Id_Mineral*, que es un entero consecutivo y llave primaria de la tabla y *Nombre*, como una cadena.
- **Muestra_Mineral:** Almacena la concentración de cada mineral que posee el yacimiento para cada muestra del mismo. Tiene como atributos *Id_Muestra* e *Id_Mineral* como llaves primarias y

CAPÍTULO 3: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PROPUESTA

foráneas de la tabla y *Concentracion_Mineral*, valor doble que representa la concentración del mineral seleccionado en dicha muestra.

3.7 Conclusiones Parciales

En este capítulo se realizó una descripción de la solución propuesta. Después de realizado un estudio del capítulo, se concluye lo siguiente:

- Con la definición del modelo de dominio se obtuvo una mejor comprensión del negocio.
- Con la descripción de los requisitos funcionales y la modelación de los casos de uso del sistema se logró definir la forma en que se desarrollará el sistema.
- Con la descripción de los requisitos no funcionales, se logró conocer las características que debe tener el sistema, así como las necesidades de hardware y software que presenta el mismo.
- La selección de la arquitectura permitió definir la estructura principal de la herramienta que se desea desarrollar.
- El uso de patrones generales de software para asignación de responsabilidades permitió realizar un buen diseño de la herramienta que se desarrolló.
- La realización del modelo de diseño facilitó el inicio de la implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA DE LA HERRAMIENTA DESARROLLADA

En el siguiente capítulo se realizarán las actividades correspondientes a los flujos de Implementación y Pruebas, definiéndose los fines de la implementación y la selección de las pruebas, la técnica y el tipo que se le realizaron con el objetivo de encontrar los errores en la herramienta que se implementó.

4.1 Arquitectura de información

Para el adecuado diseño de la aplicación se propone seguir una serie de estándares que van encaminados a garantizar su consistencia:

- Brindar una interfaz sencilla, de manera tal que cualquier persona con un mínimo dominio de la computación pueda aprender a trabajar con la aplicación.
- Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- Mostrar al usuario, siempre que vaya a realizar una acción relevante sobre el sistema, un mensaje de confirmación que le permita asegurarse que es correcta la opción seleccionada.

4.2 Implementación

En la implementación es donde se describe como los elementos del modelo de diseño se implementan en términos de componentes. Se realiza una descripción de la organización de los componentes, de acuerdo con los mecanismos de estructuración disponibles en el lenguaje utilizado.

Esta disciplina explica cómo desarrollar y realizar pruebas de unidad así como la integración de componentes desarrollados.

La implementación tiene como fin según (CORP 2006):

REFERENCIAS BIBLIOGRÁFICAS

- Definir la organización del código, en términos de los subsistemas de implementación, organizados en capas.
- Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros).
- Probar y desarrollar componentes como unidades.
- Integrar los resultados producidos por los implementadores individuales, o equipos en un sistema ejecutable.

4.3 Diagrama de componentes

Los diagramas de componentes son utilizados para estructurar el modelo de implementación en términos de subsistemas y para modelar las vistas de un sistema, describiendo sus elementos físicos y relaciones. Además, contienen los elementos de arquitectura física necesarios para determinar los componentes e interfaces a desarrollar en el resto del ciclo de vida (AREBA 2001).

REFERENCIAS BIBLIOGRÁFICAS

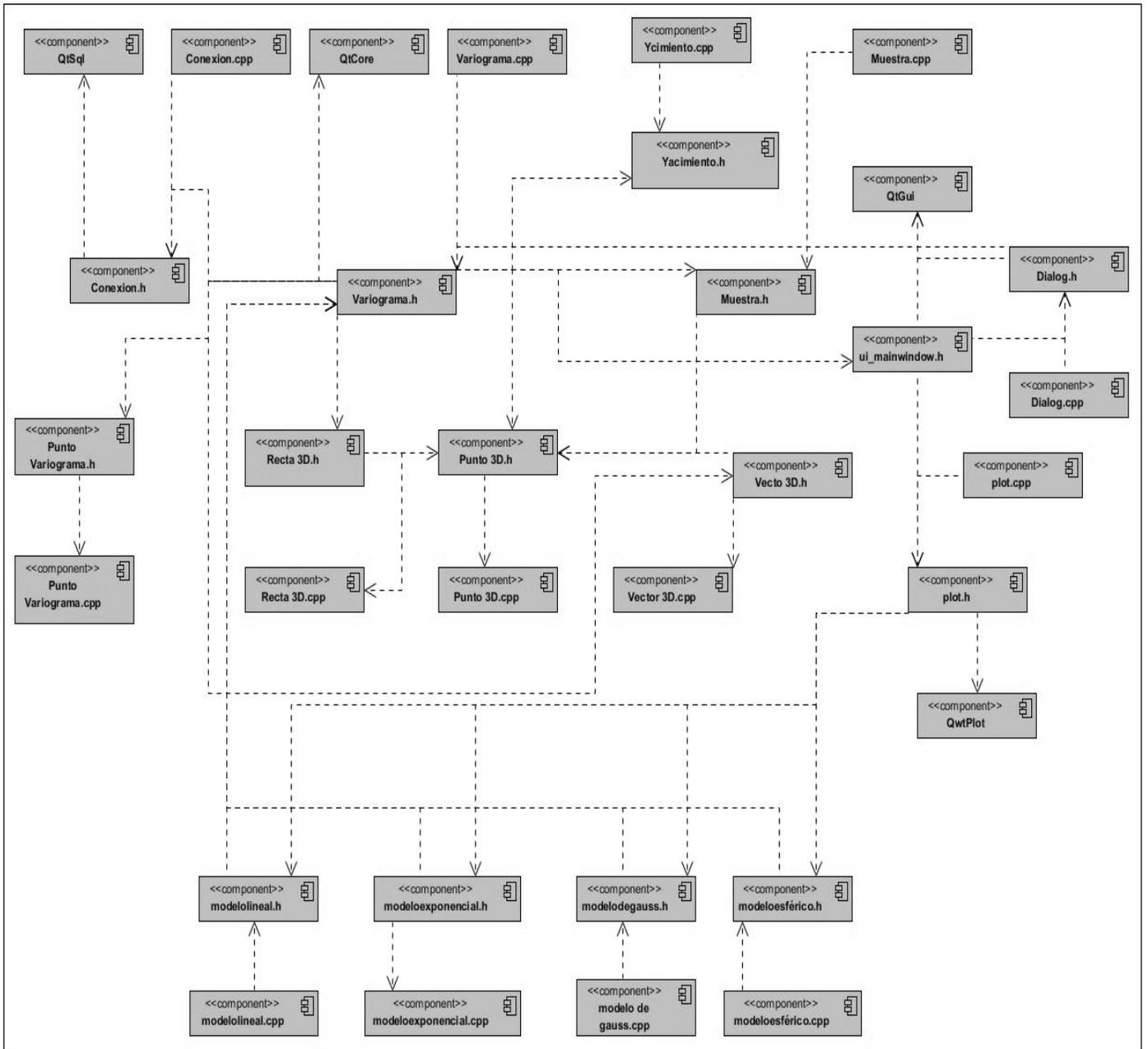


Figura 24. Diagrama de componentes de la herramienta desarrollada.

REFERENCIAS BIBLIOGRÁFICAS

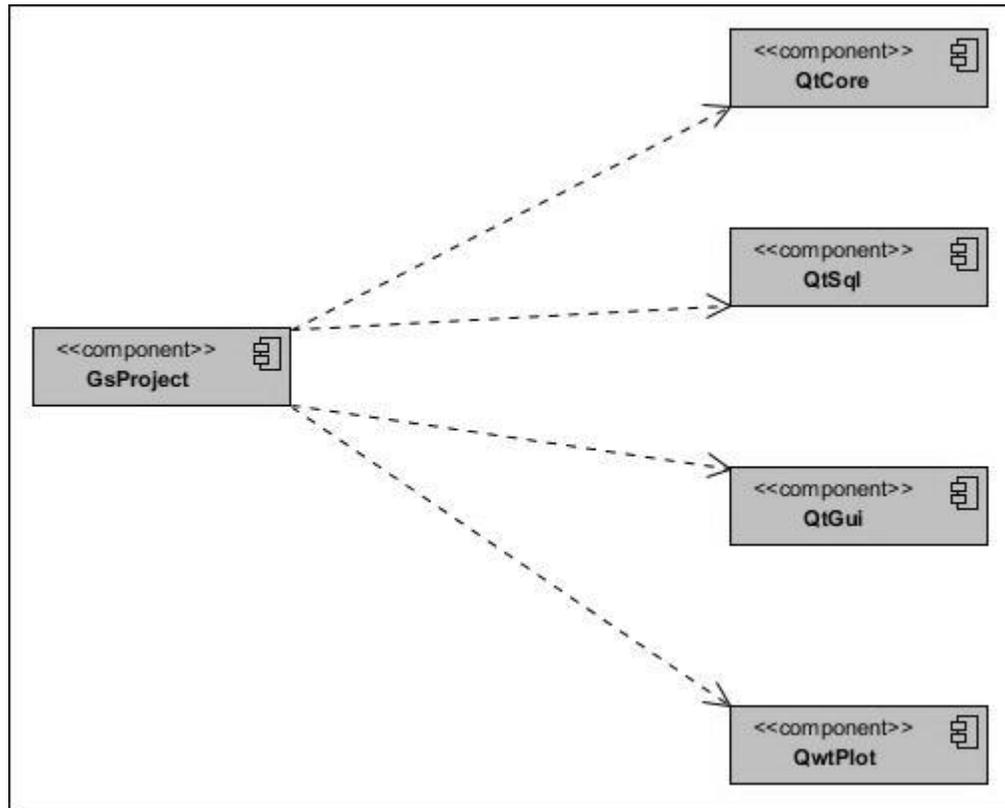


Figura 25. Diagrama de componentes del ejecutable de la herramienta desarrollada.

4.4 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño (JACOBSON 2000).

4.5 Diagrama de despliegue



Figura 26. Diagrama de despliegue de la aplicación desarrollada.

4.6 Prueba

Hay una manera de probar cualquier producto construido (y casi cualquier cosa): si se conoce la función específica para la que se diseñó el producto, se aplican pruebas, que demuestren que cada función es plenamente operacional, mientras se buscan errores de cada función a esta prueba se le determinó como prueba de caja negra (PRESSMAN 2005).

4.6.1 Pruebas unitarias o de unidad

Las pruebas unitarias aíslan cada parte del programa y muestran que las partes individuales son correctas. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto (CALZADILLA and ECHEMENDÍA 2010).

Estas pruebas aisladas proporcionan ciertas ventajas básicas:

- Fomentan el cambio: las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.

REFERENCIAS BIBLIOGRÁFICAS

- Simplifica la integración: permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- Separación de la interfaz y la implementación: la única interacción entre los casos de prueba y las unidades bajo prueba son las interfaces de estas últimas.
- Los errores están más acotados y son más fáciles de localizar: se prueban pequeñas porciones del software.

4.6.2 Técnica de caja negra.

La técnica de caja negra se centra en los requisitos funcionales del *software*. Es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Se trata de un enfoque que intenta descubrir diferentes tipos de errores que no se encuentran con los métodos de caja blanca (PRESSMAN 2006). Este método de prueba permite verificar los aspectos funcionales del sistema, sin tener en cuenta su estructura lógica interna.

Para verificar que las principales funcionalidades del sistema efectúen de forma satisfactoria la operación que realiza, se decidió realizar pruebas de caja negra a los principales casos de uso del sistema. Se utilizó la técnica de partición equivalente para representar, a través de escenarios, una descripción de la acción del usuario, la respuesta que se espera y el resultado obtenido, lo que permite comprobar el buen funcionamiento del sistema.

4.6.3 Partición equivalente

Este es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (CALZADILLA and ECHMENDÍA 2010).

REFERENCIAS BIBLIOGRÁFICAS

Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica.

El objetivo de la partición equivalente es reducir el posible conjunto de casos de prueba en uno más pequeño, un conjunto manejable que evalúe bien el software. Se toma un riesgo porque se escoge no probar todo, así que se necesita tener mucho cuidado al escoger las clases.

En el diseño de casos de prueba para partición equivalente se procede en dos pasos:

1. Se identifican las clases de equivalencia: son identificadas tomando cada condición de entrada (generalmente una oración o una frase en la especificación) y repartiéndola en dos o más grupos. Es de notar que dos tipos de clases de equivalencia están identificados: las válidas representan entradas válidas al programa, y las inválidas que representan el resto de los estados posibles de la condición.
2. Se definen los casos de prueba: se asigna un número único a cada clase de equivalencia, hasta que todas las clases de equivalencia válidas han sido cubiertas por los casos de prueba, se escribe un nuevo caso de prueba que cubra la clase de equivalencia válida y por último hasta que los casos de prueba hayan cubierto todas las clases de equivalencia inválidas, se escribe un caso de la prueba que cubra una, y solamente una, de las clases de equivalencia inválidas descubiertas.

4.7 Casos de prueba

Es de vital importancia asegurar la calidad de la realización de las pruebas, para esto es necesario que la construcción de los casos de prueba que se ejecutan para comprobar el correcto funcionamiento del sistema se realice bajo la guía de casos de prueba.

Tabla 2. Caso de prueba "Pintar semivariograma experimental".

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo central
----------------------	--------------------------	---------------------------------	---------------

REFERENCIAS BIBLIOGRÁFICAS

<p>SC 1: "Pintar Semivariograma Experimental"</p>	<p>EC 1.1: "Pintar Semivariograma Experimental con éxito.</p>	<p>El usuario selecciona la opción Semivariograma experimental del menú Mostrar. El sistema habilita una interfaz en la que aparecen varios campos que el geólogo debe llenar, un campo para seleccionar el mineral, otro para entra el ancho de banda, uno para la tolerancia angular y tres campos para entrar las coordenadas de un punto que junto al origen de coordenadas forman la recta que da la dirección al semivariograma experimental. El geólogo entra los datos requeridos por el sistema y selecciona la opción Pintar semivariograma. El sistema pinta el semivariograma experimental.</p>	<p>Herramienta para la Creación de Semivariogramas.</p> <ol style="list-style-type: none"> 1) El geólogo selecciona la opción Semivariograma experimental del menú Mostrar. 2) El sistema habilita una interfaz con los campos de entrada de datos. <ul style="list-style-type: none"> • Mineral(seleccionar) • Ancho de banda • Tolerancia angular • X • Y • Z 3) El geólogo introduce los datos en los campos requeridos por el sistema y selecciona la opción Pintar semivariograma. 4) El sistema valida los datos y grafica el semivariograma experimental.
	<p>EC 1.2: "Pintar Semivariograma Experimental"</p>	<p>Los datos entrados por el usuario presentan problemas.</p>	<p>Herramienta para la Creación de Semivariogramas.</p>

REFERENCIAS BIBLIOGRÁFICAS

	sin éxito.		<ol style="list-style-type: none"> 1) El geólogo selecciona la opción Semivariograma experimental del menú Mostrar. 2) El sistema habilita una interfaz con los campos de entrada de datos. <ul style="list-style-type: none"> • Mineral(seleccionar) • Ancho de banda • Tolerancia angular • X • Y • Z 3) El geólogo introduce los datos en los campos requeridos por el sistema y selecciona la opción Pintar semivariograma. 4) El sistema valida los datos y muestra un mensaje de error "Revisar formato de los datos"
--	------------	--	--

Tabla 3. Descripción de las variables para el caso de prueba "Pintar semivariograma experimental".

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Mineral	Seleccionable	No	El geólogo
2	Ancho de banda	Campo de texto	No	El geólogo
3	Tolerancia angular	Campo de texto	No	El geólogo
4	X	Campo de texto	No	El geólogo

REFERENCIAS BIBLIOGRÁFICAS

5	Y	Campo de texto	No	El geólogo
6	Z	Campo de texto	No	El geólogo

Tabla 4. Matriz de datos del caso de prueba "Pintar semivariograma experimental".

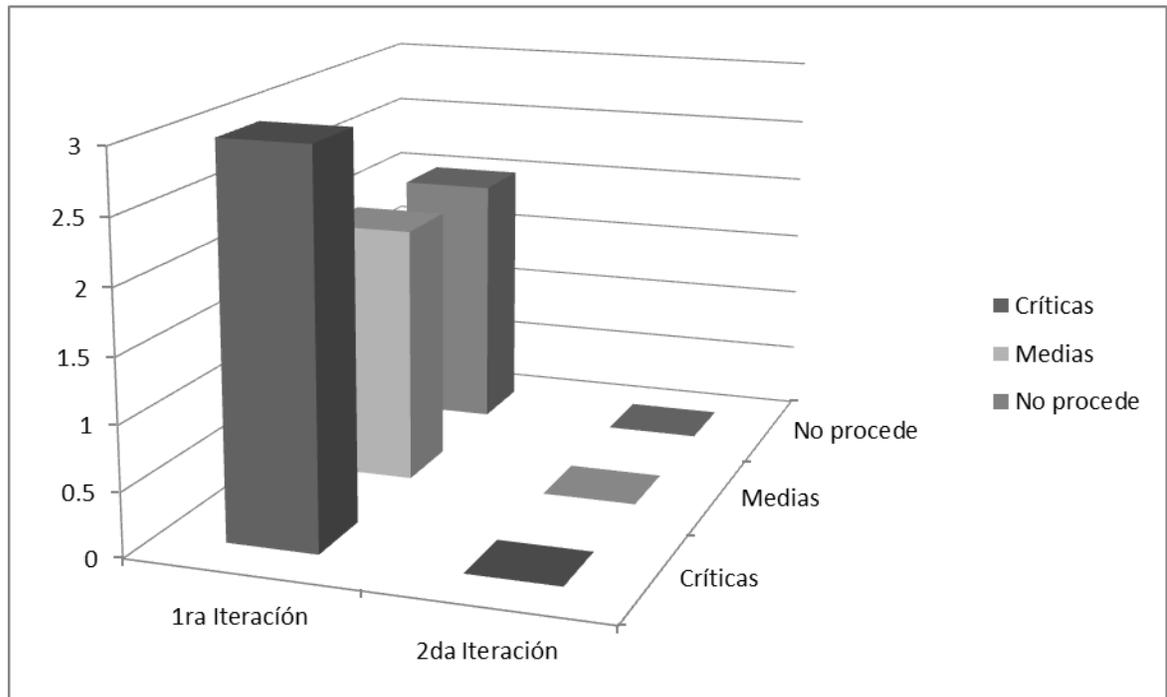
Id del escenario	Mineral	Ancho de banda	Tolerancia angular	X	Y	Z	Respuesta del sistema	Resultado de la prueba
EC 1.1	V	V	V	V	V	V	El sistema habilita los campos Efecto Papita, Meseta y Alcance.	Satisfactorio
	Ni	45	45	1	1	1		
EC 1.2	V	I	V	V	V	V	El sistema muestra el mensaje de error "Revisar formato de los datos"	Satisfactorio
	Ni	aa	45	1	1	1		
	V	I	V	I	V	V	El sistema muestra el mensaje de error "Revisar formato de los datos"	Satisfactorio
	Ni	aa	45	c	1	1		
	V	V	I	V	V	V	El sistema muestra el mensaje de error "Revisar formato de los datos"	Satisfactorio
	Ni	45	\$%	1	1	1		

4.8 Resultado de las pruebas

En la realización de la primera iteración de las pruebas fueron detectadas siete no conformidades, de ellas tres afectan al sistema de forma crítica, dos de forma media y dos no procedieron.

Después de finalizada la primera iteración se corrigieron los errores encontrados y se realizó una segunda iteración en la que no se encontraron no conformidades pudiéndose apreciar el correcto funcionamiento del sistema.

REFERENCIAS BIBLIOGRÁFICAS



4.9 Conclusiones Parciales

Después de concluido este capítulo se concluye lo siguiente:

- Con la implementación del sistema se obtuvo la herramienta que permite la creación y ajuste de semivariogramas.
- Los requisitos funcionales definidos fueron correctamente implementados.
- Los diseños de casos de prueba desarrollados como parte de las pruebas de caja negra, permitieron la validación de la interfaz de cada caso de uso.
- La realización de las pruebas permitió corregir los errores encontrados en la aplicación de las mismas.

CONCLUSIONES

Una vez cumplida la investigación se concluye lo siguiente:

- Los artefactos que se generaron en el proceso de construcción de la solución permitirán continuar mejorando la misma en el futuro.
- La selección de la arquitectura y la aplicación de patrones generales de software para asignación de responsabilidades permitió obtener una estructura sólida para la herramienta que se desarrolló
- Los diseños de casos de prueba desarrollados como parte de las pruebas de caja negra, permitieron la validación de la interfaz de cada caso de uso.
- La realización de las pruebas permitieron corregir los errores encontrados durante las mismas.
- Se obtuvo una herramienta que logra la descripción de la variabilidad y la correlación espacial de las muestras de un yacimiento mineral.
- La utilización de la herramienta obtenida permite dar solución a una parte necesaria de la estimación de recursos minerales con métodos geoestadísticos.

RECOMENDACIONES

RECOMENDACIONES

En versiones posteriores de este sistema se recomienda lo siguiente:

- Se recomienda en futuras versiones realizar ajuste del semivariograma experimental a los modelos Tendencia y Efecto Hole.
- Se recomienda lograr la integración de la herramienta para la creación de semivariogramas, con el Sistema de Análisis y Modelado de Yacimientos Minerales (**Syam**).

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- AREBA, J. B. D. *Metodología del Análisis Estructurado de Sistemas*. Madrid, 2001. p.
- CALZADILLA, Y. R. and Y. T. ECHEMENDÍA. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas: Estrategia de pruebas de software para el proyecto Fuerza de Trabajo Calificada*. La Habana, UCI, 2010. p.
- CASE. Case, 2012. [Disponible en: <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>]
- CASSIRAGA, E. *Curso de Geoestadística*, 2007.
- COCKBURN, A. *Agile Software Development*, 2006.
- CORNEJO, J. E., Ed. *Arquitectura en Capas DNA. Un camino hacia los procesos distribuidos*, 2006.
- CORP, I. *Rational Software Architect*, 2006.
- DOMINY, S. C.; P. R. STEPHENSON, et al. Clasification and Reporting of Mineral Resources for High-Nugget Effect Gold Vein Deposits *Explor. Mining Geol.*, 2001, 10(3): 215-233.
- GAMMA, E.; R. HELM, et al. *Patrones de Diseño*, 1995.
- GARCÍA, J. *Patrones de diseño. Diseño de Software Orientado a Objetos.*, 2005.
- GARLAN, D. *An Introduction to Software Architecture School of Computer Science*, 1994.
- GEMCOM. *Gemcom Minex. Geology and Mine Planning for Stratified Deposits*, 2012.
- GIL, J. Q. C., Ed. *Análisis de Datos Distribuidos Espacialmente*, 2008.
- GIL, J. Q. C. and E. E. CRUZ. *Conceptos Fundamentales de la Geoestadística*, 2007.
- IBARRA, A. A. and M. A. M. FORES. *Postgres SQL*, 2008. [Disponible en: <https://iessanvicente.com/colaboraciones/postgreSQL.pdf>]
- JACOBSON, P. *Ingeniería de Software*, 2000.
- JACOBSON, P.; M. JONSSON, et al., Eds. *Ingeniería de Software orientada a Objetos- Un acercamiento a través de los casos de uso*, 1992.
- KRAJEWSKI, S. A. and B. L. GIBBS *Variogram Premier*, 2010.
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Prentice Hall, 2003. p. 970-17-0261-1
- LOPEZ, A. Q., Ed. *Herramienta de Pruebas Automatizadas para el módulo HMI del editor Phoenix* La Habana, 2011.

REFERENCIAS BIBLIOGRÁFICAS

- OLEA, R. A. *Geostatistics for engineers and earth scientists*. Kluwer Academic Publishers, 1999. 303 p. 0-7923-8523-3
- ONLINE, T. E. I. *ISATIS - Geostatistics software solution for mining*, Enviromental Expert.com, 2013. [Disponible en: <http://www.environmental-expert.com/software/isatis-geostatistics-software-solution-for-mining-41613>]
- PRESSMAN, R., Ed. *Ingeniería de Requisitos Parte 1, Cap 07*, 2002.
- PRESSMAN, R., Ed. *Ingeniería de Software. Un enfoque práctico* La Habana, 2006.
- PRESSMAN, R. *Ingeniería del Software*, 2005.
- PROJECT, Q. *Qt Project*, 2012. [Disponible en: <http://qt-project.org/wiki/QtCreatorWhitepaper>]
- SGEMS. *SGeMS Stanford Geostatistical Modeling Software*, 2013. [Disponible en: <http://sgems.sourceforge.net/>]
- SIRONVALLE, M. A. A., Ed. *Estadística*, 2000.
- SOMERVILLE, I., Ed. *Ingeniería de Software* Madrid, 2005. 84-7829-074-5
- THELIN, J. *Foundations of Qt Development*. New York, Apress, 2007. p. 978-1-59059-831-3
- TULCANZA, E. *Evaluación de recursos y negocios mineros.*, Impresos Universitaria S.A., 1999. p. 956-288-407-4
- VIERA, M. A. D. *Sobre Geoestadística*, 2002.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Anselin, L. 2003, 'An Introduction to Variography using Variowin', Illinois.
- Areba, J. B. d. 2001, *Metodología del Análisis Estructurado de Sistemas*, Madrid.
- Arechavala, Y. G. (ed.) 2001, *Calidad de Software 1*.
- Arquitectura 2010, 'Capítulo II Arquitectura del Software'.
- Buschmann F, M., R, R., Sons, S. I. & Wiley, J. 1996, *Pattern-Oriented Software Architecture: A System of*.
- Calzadilla, Y. R. & Echemendía, Y. T. 2010, *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas: Estrategia de pruebas de software para el proyecto Fuerza de Trabajo Calificada*, UCI.
- Case 2012, Case, [<http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>].
- Cassiraga, E. 2007, 'Curso de Geoestadística'.
- Chiles, J.-P. & Delfiner, P. 1999, *Geostatistics: Modeling Spatial Uncertainty*, John Wiley & Sons 695 p.
- Cockburn, A. 2006, 'Agile Software Development'.
- Cornejo, J. E. (ed.) 2006, *Arquitectura en Capas DNA. Un camino hacia los procesos distribuidos*.
- Corp, I. 2006, 'Rational Software Architect'.
- Dominy, S. C., R. Stephenson, P. & Annels, A. E. 2001, 'Clasificación and Reporting of Mineral Resources for High-Nugget Effect Gold Vein Deposits', *Explor. Mining Geol.*, vol. 10, no. 3, pp. 215-233.
- Gamma, E., Helm, R., Johnson, R. & Vlissides, J. 1995, 'Patrones de Diseño'.
- García, J. 2005, 'Patrones de diseño. Diseño de Software Orientado a Objetos'.
- García, R. M. M. 1999, *Diseño de bases de datos*.
- Garlan, D. 1994, 'An Introduction to Software Architecture School of Computer Science'.
- Gemcom 2012, 'Gemcom Minex. Geology and Mine Planning for Stratified Deposits'.
- Gil, J. Q. C. 2005, 'ESTUDIOS DE ESTIMACIÓN Y SIMULACIÓN GEOESTADÍSTICA PARA LA CARACTERIZACIÓN DE PARAMETROS GEOLOGO-INDUSTRIALES EN EL YACIMIENTO LATERÍTICO DE PUNTA GORDA', Holguín.
- Gil, J. Q. C. (ed.) 2008, *Análisis de Datos Distribuidos Espacialmente*.
- Gil, J. Q. C. & Cruz, E. E. 2007, 'Conceptos Fundamentales de la Geoestadística'.
- Guimson, L. 2012, *Metodologías ágiles y desarrollo basado en conocimiento*.
- Ibarra, A. A. & Fores, M. A. M. 2008, *Postgres SQL*, [<https://iessanvicente.com/colaboraciones/postgreSQL.pdf>].
- Jacobson, P. 2000, 'Ingeniería de Software'.
- Jacobson, P., Jonsson, M. & Overgard, G. (eds.) 1992, *Ingeniería de Software orientada a Objetos- Un acercamiento a través de los casos de uso*.
- Kitanidis, P. K. 1997, *Introduction to Geostatistics: Applications in Hydrogeology.*, Cambridge University Press. 249 p.
- Krajewski, S. A. & Gibbs, B. L. 2010, 'Variogram Premier'.
- Larman, C. 2003, *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, Prentice Hall.
- Lopez, A. Q. (ed.) 2011, *Herramienta de Pruebas Automatizadas para el módulo HMI del editor Phoenix*, La Habana.
- Lujan, S. 2010, *Publicaciones de la Universidad de Alicante*, [<http://publicaciones.ua.es/publica/Detalles.aspx?fndCod=LI9788479088880&idet=958>].
- Olea, R. A. 1999, *Geostatistics for engineers and earth scientists*, Kluwer Academic Publishers. 303 p.
- Online, T. E. I. 2013, *ISATIS - Geostatistics software solution for mining*, [[63](http://www.environmental-</p></div><div data-bbox=)

BIBLIOGRAFÍA

- expert.com/software/isatis-geostatistics-software-solution-for-mining-41613].
- Paradigm, V. 2007, *Paradigma visual para UML (Plataforma Java) (Visual Paradigm for UML [Java Platform]) 6.0*.
- Pressman, R. (ed.) 2002, *Ingeniería de Requisitos Parte 1, Cap 07*.
- Pressman, R. 2004, *Ingeniería de Software. Un enfoque práctico*, Felix Varela, La Habana.
- Pressman, R. 2005, 'Ingeniería del Software'.
- Pressman, R. (ed.) 2006, *Ingeniería de Software. Un enfoque práctico*, La Habana.
- Project, Q. 2012, *Qt Project*, [<http://qt-project.org/wiki/QtCreatorWhitepaper>].
- Sande, M. 2004, 'Programación en C++ con Qt bajo Entorno GNU/Linux'.
- SGems 2013, *SGeMS Stanford Geostatistical Modeling Software*, [<http://sgems.sourceforge.net/>].
- Sironvalle, M. A. A. (ed.) 2000, *Estadística*.
- Sironvalle, M. A. A. 2007, 'Estimación de Recursos Mineros'.
- Somerville, I. (ed.) 2005, *Ingeniería de Software*, Madrid.
- Thelin, J. 2007, *Foundations of Qt Development.*, Apress, New York.
- Tulcanza, E. 1999, *Evaluación de recursos y negocios mineros.*, Impresos Universitaria S.A.
- Viera, M. A. D. 2002, 'Sobre Geoestadística'.
- Wackernagel, H. 1998, *Multivariate Geostatistics: An introduction with applications*, 2nd edn, Springer. 291 p.
- Yailen Rondón Calzadilla, Y. T. E. 2010, *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas: Estrategia de pruebas de software para el proyecto Fuerza de Trabajo Calificada*.

GLOSARIO DE TÉRMINOS

Variabilidad espacial: Propiedad de una variable regionalizada, que define como está distribuida la misma espacialmente.

Correlación espacial: Propiedad de una variable regionalizada, que define el nivel de relación que tienen los valores que toma la misma en diferentes posiciones espaciales.

Semivariograma: Herramienta que permite graficar una función matemática continua, que permite la descripción de la variabilidad y la correlación espacial de una variable regionalizada distribuida espacialmente.

Geoestadística: Ciencia que estudia estadísticamente los fenómenos naturales.

Caso de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.

Clase: Una clase es un contenedor de uno o más datos (variables o propiedad miembro) junto a las operaciones de manipulación de dichos datos (funciones/métodos). Las clases pueden definirse como estructuras, uniones o clases pudiendo existir diferencias entre cada una de las definiciones según el lenguaje. Además las clases son agrupaciones de objetos que describen su comportamiento.

Objeto: Un objeto es la representación de un concepto para un programa, y contiene toda la información necesaria para abstraer dicho concepto: los datos que describen su estado y las operaciones que pueden modificar dicho estado.