

SISTEMAS DE REFERENCIA

Análisis e implementación a través del paradigma Orientado a Objetos

Trabajo de Grado de la Licenciatura en Informática de
Cecilia Islas
Emilio D'Angelo

Dirigido por: Silvia Gordillo

Facultad de Informática
Universidad Nacional de La Plata
Octubre, 2000

Indice

INDICE	2
DEDICACIÓN Y AGRADECIMIENTOS.....	4
1. INTRODUCCIÓN	6
2. CONCEPTOS FUNDAMENTALES	8
2.1 SISTEMAS DE INFORMACION GEOGRAFICA (S.I.G.)	8
2.1.1 Definición de SIG	8
2.1.4. Dato: el componente principal de todo SIG.....	10
2.2 TECNOLOGÍA ORIENTADA A OBJETOS	13
2.2.1 El paradigma de Orientación a Objetos.....	13
2.2.2. Características del Diseño Orientado a Objetos	14
2.2.3 Patrones de diseño	15
2.3 SISTEMAS DE INFORMACIÓN GEOGRÁFICOS ORIENTADOS A OBJETOS (SIGOO)	19
2.3.1. Beneficios de la orientación a objetos para un SIG:	19
2.3.2. Comparación en el uso de bases de Datos:	20
2.3.3. Modelización de BD para SIGOO.....	21
2.3.4. Estimación de tiempo y costos.....	21
3. SISTEMAS DE REFERENCIA	22
3.1 INTRODUCCIÓN A LOS SISTEMAS DE REFERENCIA.....	22
3.2 LA FIGURA DE LA TIERRA	24
3.2.1. El Geoide	24
3.2.2. Elipsoide.....	25
3.3 DATUMS	29
3.3.1. Introducción.....	29
3.3.2 Marcos de referencia [Brunini-Rodríguez].....	30
3.3.3. Transformaciones entre distintos Datums	31
3.4 PROYECCIONES CARTOGRÁFICAS	34
3.4.2 Propiedades de las proyecciones.....	34
3.4.3 Elección de un sistema de proyección adecuado	35
3.4.4. Clasificación de las proyecciones	38
3.5. SISTEMAS DE COORDENADAS	43
3.5.1. Introducción.....	43
3.5.2. Transformación entre sistemas de coordenadas	44
4. MODELO DE SISTEMAS DE REFERENCIA.....	46
4.1. MODELO BASE.....	46
4.1.1. Introducción al modelo base	46

4.2. ANALISIS DEL MODELO	48
4.2.1 Dominio	48
4.2.2. Análisis de los elementos del modelo.....	49
4.2.3 Análisis del comportamiento y colaboraciones entre los elementos	50
4.3. ARQUITECTURA DEL MODELO DE SISTEMAS DE REFERENCIA	52
4.3.1. Descripción y presentación del modelo.....	52
4.3.2. Descripción de Clases de Modelo.....	54
4.3.3. Diagrama de Clases de Sistemas de Referencia	62
4.3.4. Comportamiento y colaboraciones entre las clases	62
4.3.4.1. Transformación entre sistemas de referencia	62
4.3.4.2. Operaciones entre objetos con diferentes sistemas de referencia	65
4.3.5. Patrones de diseño utilizados	66
4.3.5.1. Patrón Strategy (Gamma)	66
4.3.5.2 Patrón “Quantity” (‘Observations and Measurements’ de Fowler).....	67
4.3.6. Aplicación del Patrón Quantity al modelo.....	68
4.3.6.1. Introducción.....	68
4.3.6.2. Clases nuevas derivadas del patrón.....	71
4.3.6.3. Diseño del patrón Quantity	71
4.3.6.4. Diseño de clases del modelo luego de aplicar el patrón ...	73
4.3.7. Modelo Final de Sistemas de Referencia	75
4.3.7.1. Diagrama de Clases	76
4.3.7.2. Diagrama de Objetos	77
5. IMPLEMENTACIÓN.....	78
5.1 IMPLEMENTACION DEL MODELO DE SISTEMAS DE REFERENCIA	78
5.1.1 Introducción.....	78
5.1.2. Implementación del patrón Quantity	78
5.1.3. Implementación de las clases principales del modelo de sistemas de referencia.....	83
5.1.4. Implementación de las Funciones de Conversión	86
5.1.4. Instanciación del modelo	88
5.2 IMPLEMENTACION DE LA APLICACION	91
5.2.1 Objetivo de la aplicación.....	91
5.2.2 Características Generales	91
5.2.3 Ejemplo de uso de la Aplicación.....	92
6. CONCLUSIONES	97
6.1. CONCLUSIONES.....	97
6.2. CONTRIBUCIONES	97
6.3. EXTENSIONES	98
7. BIBLIOGRAFÍA	99

Dedicación y Agradecimientos

Dedicamos este trabajo a nuestros padres y abuelos, quienes nos acompañaron y ayudaron durante toda la carrera.

Queremos agradecer a Silvia Gordillo por habernos orientado y apoyado durante el desarrollo de esta tesis y los trabajos realizados en el Laboratorio. También agradecemos a todo el grupo GIS del LIFIA con quienes compartimos muchas charlas y discusiones.

“ Tomar posesión del espacio es el primer gesto de los seres vivientes, de los hombres y de los animales, de las plantas y de las nubes, manifestación fundamental de equilibrio y de duración. La primera prueba de existencia es ocupar el espacio”

Le Corbusier

1. Introducción

Todo SIG posee la capacidad de representar y manipular las características espaciales de cualquier entidad. Dado que toda entidad geográfica tiene una posición sobre la esfera terrestre, esta es usada fundamentalmente para poder ubicarla y para realizar operaciones espaciales como, por ejemplo, medir su distancia a otras entidades.

La manera de representar dicha posición es a través de algún valor (matemático, no matemático, discreto, continuo, etc.), siendo este valor distinto para diferentes entidades. Dicho valor carecería de sentido si no existiera un ente que lo interprete y le provea cualidades tales como, por ejemplo en el caso matemático, distancia a otras entidades, área, volumen, superficie, etc.

Un *Sistema de Referencia* le asigna a una entidad, sentido de ubicación geográfica, así como también define el conjunto de operaciones que se pueden aplicar a una o entre muchas entidades.

Por otra parte, los usuarios SIG se ven generalmente en la necesidad de integrar datos provenientes de diferentes fuentes, usando para esto el mínimo común denominador generalmente disponible en esos casos: la ubicación geográfica. Es por esto que surge la necesidad de utilizar modelos de datos flexibles, que manejen esta heterogeneidad y a la vez preserven la posibilidad de llevar a cabo las tareas usuales y las no usuales. Un ejemplo de esto es la necesidad de visualizar o procesar como un todo a la unión de dos conjuntos de datos obtenidos con dos Sistemas de Referencia distintos, esto significa una transformación de coordenadas entre los Sistemas de Referencia involucrados, cosa que muy posiblemente no sea provista por la aplicación SIG.

La existencia de muchos posibles Sistemas de Referencia hace que sea necesario que, de un valor expresado en un Sistema, pueda conocerse cual sería ese valor en otro Sistema de Referencia. Para lograr esto, se debe contar con los medios necesarios para crear los Sistemas de Referencia involucrados, manipularlos y, además, conocer y definir las metodologías para lograr la transformación entre ellos.

Para lograr lo antes mencionado, la tecnología mas adecuada es la Orientación a Objetos.

Existe un creciente interés hacia el uso de la tecnología Orientada a Objetos para la modelización. La ventaja del uso de modelos Orientados a Objetos es que permiten a los usuarios hacer especificación incremental y reusable, debido a las propiedades de herencia, composición y encapsulamiento. La modelización Orientada a Objetos logra una gran correspondencia entre la implementación física y el modelo lógico. Mediante ella se obtienen definiciones más flexibles y el manejo de las relaciones entre los objetos está disponible, lo cual logra acercar aun más el modelo lógico a la implementación resultante.

En este trabajo se define una arquitectura que brinde a cualquier aplicación construida bajo el paradigma de Orientación a Objetos que lo requiera, la capacidad de crear y manejar distintos Sistemas de Referencia en forma integrada, desacoplando esta nueva funcionalidad del propio modelo de la aplicación de manera de obtener una solución

adaptable sin afectar a la misma. Proveer la capacidad de realizar conversiones entre los distintos Sistemas de Referencia, en el caso de que estas conversiones se conozcan y sean provistas.

Como base para esta definición se utiliza el modelo definido en [Gordillo 97] que parte de un modelo Orientado a Objetos y permite tanto la extensión de aplicaciones convencionales con características geográficas como la modelización de aplicaciones SIG (sistemas de información geográficos) basándose en este paradigma.

En el capítulo 2 se describen los conceptos fundamentales que son necesarios para el entendimiento de esta tesis, describiéndose por consiguiente el paradigma Orientada a Objetos, las aplicaciones SIG (Sistemas de Información Geográficos) y finalmente las ventajas de utilizar aplicaciones SIGOO (Sistemas de Información Geográficos Orientados a Objetos).

El capítulo 3 expone todo lo concerniente a Sistemas de Referencia específicamente, detallando las diferentes maneras de referenciar puntos terrestres, así como también la descripción de sistemas de coordenadas, datums, proyecciones cartográficas, etc., definiendo de esta manera todo el dominio que constituirá la base para el análisis e implementación de esta tesis.

El capítulo 4 describe todo el proceso de análisis de la tesis, aquí se menciona el modelo base sobre el que empezamos a trabajar, se detallan los patrones de diseño aplicados, se describen las clases, se describe el comportamiento del modelo y se obtiene el modelo final de los Sistemas de Referencia.

El capítulo 5 trata sobre los aspectos de la implementación, tanto del modelo de Sistemas de Referencia, como de la aplicación final desarrollada.

Por último el capítulo 6 contiene las conclusiones y las contribuciones que aporta esta tesis para trabajos futuros, así como sus posibles extensiones.

2. Conceptos Fundamentales

2.1 SISTEMAS DE INFORMACION GEOGRAFICA (S.I.G.)

2.1.1 Definición de SIG

La demanda creciente ejercida por una sociedad cada vez más tecnificada sobre los recursos naturales y el medio ambiente hace que cada vez sea más importante contar con información territorial precisa y actualizada. Esta es la razón por la cual se desarrollaron los conocidos Sistemas de Información geográfica (SIG).

Una aplicación SIG varía desde el manejo de información terrestre, como planeamiento urbano, investigación ambiental, diseño catastral etc.; hasta su uso en el planeamiento y manejo de la ingeniería para las industrias que lo requieran. Otra de sus características es proveer métodos flexibles para explorar relaciones entre datos geográficos y asistir a los expertos de diversas disciplinas en combinar su conocimiento para resolver problemas complejos.

Existen muchas definiciones de Sistemas de Información Geográficos, cada autor propone una definición desde su perspectiva.

A continuación se enunciarán las definiciones más usuales:

"...un conjunto de herramientas para guardar y recuperar a voluntad, transformando y mostrando datos espaciales del mundo real para un conjunto particular de propósitos..." (Burrough, 1986,p.6)

"...un sistema automatizado para capturar, almacenar, recuperar, analizar, y mostrar datos espaciales..." (Clarke, 1995, p.13)

Las definiciones anteriores ven al SIG como una herramienta mediante la cual se pueden manipular datos. Otros autores, en cambio, definen un SIG como un sistema de información especializado:

"...un sistema de información que se diseña para trabajar con datos referenciados por coordenadas espaciales o geográficas. En otras palabras, un SIG es tanto un sistema de BD con capacidades específicas para datos espacialmente referenciados, así como un conjunto de operaciones para trabajar con los datos..." (Star y Estes, 1990, pág. 2).

La definición de Duecker propuesta en 1979 ha sobrevivido la prueba de tiempo.

"...todo SIG es un caso especial de sistema de información donde la base de datos consiste en observaciones de objetos, actividades o eventos y con la característica de ser definibles en el espacio como puntos, líneas o áreas. Se proveen, además, operaciones para consultar estas observaciones con el fin del análisis comparativo..."

(Duecker, 1979, p 106).

La definición aceptada por la mayoría:

“Un SIG es una colección organizada de hardware, software, datos geográficos y personal, diseñada para eficazmente capturar, guardar, unir, actualizar, manipular, analizar, comparar y desplegar todas las formas de información geográficamente referenciada, siendo esta información espacial o descriptiva.”(ESRI, 1994).

2.1.2. Características Generales

- ◆ Un sistema de información geográfico (SIG) es un sistema integrado basado en computadora y diseñado para capturar, guardar, revisar, analizar y desplegar información geográfica.
- ◆ Como con otros sistemas gubernamentales, los elementos importantes de un SIG incluyen hardware, software, datos y personas.
- ◆ Un SIG puede realizar numerosas funciones para apoyar la toma de decisiones, como, por ejemplo, analizar regiones convenientes para una determinada actividad.

2.1.3. Componentes de un SIG

Una posibilidad de otorgar jerarquía a las componentes de un SIG podría plantearse así:

- *Personal*: tiene como función previa definir el problema que se desea resolver con un SIG específico, decide sobre la obtención de los datos necesarios, toma decisiones sobre el programa a usar y sobre su soporte informático.
- *Dato*: no existe SIG sin datos. Su adquisición es en general la componente más costosa del sistema. En consecuencia una premisa debe ser que el dato pueda tener aplicaciones múltiples, dicho de otra forma que el dato pueda ser compartido.
- *Programas*: la evolución de la informática ha llevado a que muchos programas SIG sean de uso simple ya que se basan en aplicaciones Windows. Es importante tener en cuenta que un programa comercial de uso múltiple tendrá mayor o menor eficiencia a la hora de analizar un problema en particular.
- *Computadoras*: su capacidad de almacenamiento, su velocidad de operación, sus características gráficas y la compatibilidad con el sistema operativo del programa elegido, deben estar a la medida de la magnitud del problema que se desea analizar.

2.1.4. Dato: el componente principal de todo SIG

- *Características del dato*

Cualquier dato que tenga una representación espacial puede ser motivo de un SIG, los volúmenes de una biblioteca, los enfermos de una epidemia, los elementos de una población, muestras de suelo, componentes de un sistema de servicios públicos, etc.

Cualquiera de estos datos puede estar ligado a un elemento espacial. Un punto, una línea, un polígono o una celda.

En general se dice que el dato es un atributo del elemento geométrico. Un mismo elemento geométrico puede estar ligado a un gran número de atributos. Los elementos geométricos pueden ser desplegados en un mapa digital en el que con algún criterio expresivo pueden ser mostrados sus atributos o funciones de los mismos.

Los elementos geométricos están ligados a la base de datos de los atributos o de información sobre los objetos. Esta conexión permite que los elementos geométricos puedan ser presentados, buscados, analizados, relacionados, etc. En función de sus atributos asociados.

- *Adquisición de la base geométrica*

1. Tomas convencionales
2. Cartografía digital existente
3. Digitalización de cartas topográficas
4. Imágenes satelitales
5. Toma de datos fotogramétricos

Cualquiera sea la forma de adquisición de la base geométrica existe como problema de fondo la adopción de un sistema de coordenadas único. Un sistema de coordenadas permite expresar las ubicaciones dentro de un marco de referencia (Capítulo 3). La cartografía disponible estará ligada a diferentes Sistema de Referencia, con distintos sistemas de proyecciones, a escalas distintas y en consecuencia con diferentes generalizaciones, una situación similar se da para los levantamientos fotogramétricos. La digitalización de cartas trae aparejada una serie de problemas de interpretación y la necesidad de una revisión cuidadosa de sus resultados.

Las imágenes satelitales pueden ser usadas bajo dos formas distintas: como una imagen a la que hay que georreferenciar o como un conjunto de datos numéricos por pixel relacionado de alguna forma al o a los elementos bajo análisis y que también pueden necesitar georreferenciación.

- *Estructura interna de los datos*

Los SIG manejan dos estructuras de datos, las formas convencionales de puntos, líneas y polígonos, conocida como forma vectorial o una estructura de celda (píxeles) conocida como estructura raster.

Las imágenes satelitarias se prestan a un tratamiento raster que puede ser manejado más rápidamente por una computadora. Sin embargo, para muchas aplicaciones una representación vectorial puede resultar conveniente. La situación deseable sería entonces que un SIG fuera capaz de tratar ambos tipos de información.

Cada programa SIG tiene una estructura interna que en general es incompatible con la de otro programa. Existen formatos estándar de intercambio de datos gráficos, pero la situación deseable es que un programa sea capaz de leer la información de otros para facilitar la interacción entre sistemas.

- *Estructura interna de transmisión de la información*

Como la etapa de adquisición de datos es la más costosa no sólo en dinero sino también en tiempo, es importante que en lo posible esa etapa no se repita innecesariamente. Para lograrlo es fundamental que se conozca que información existe, cual es su calidad y bajo que requisitos esa información puede ser compartida. Internet jugará sin duda un papel muy importante en este sentido. En estos momentos existe un importante esfuerzo internacional para normalizar en lo posible la estructura de los datos y la información sobre esos datos. Es posible imaginar una infraestructura global de datos georreferenciados coexistiendo con estructuras nacionales y locales que respondan a criterios de generalización similares a los usados en las diferentes escalas cartográficas.

Disponer de información georreferenciada que cumpla con estándares estrictos de calidad será en el futuro próximo una importante fuente de ingresos a condición que los metadatos que la describen tengan la más amplia difusión. Mientras las fuentes de metadatos deberían estar lo más centralizadas posible, los bancos de datos en si deben estar completamente descentralizados de forma que quien los genera pueda perfeccionarlos cuando sea necesario.

- *Quien genera los datos para un SIG*

Los profesionales de la geodesia, la agrimensura y la fotogrametría han incluido en sus incumbencias la generación del dato cartográfico. Aunque existen diferentes matices entre los problemas que estas especialidades resuelve puede pensarse que un enorme porcentaje de la información cartográfica pasaba por sus manos, hoy la información georreferenciada de alta calidad deberá surgir del especialista en el dato generado. Un médico epidemiólogo detectará y georreferenciará los datos de su estudio, el geólogo georreferenciará fallas o afloramientos, un paleontólogo el emplazamiento de los sitios de algún depósito de fósiles y esa información será usada por urbanistas, médicos sanitarios, historiadores o por especialistas en el negocio inmobiliario.

La cartografía tradicional transmite información, un SIG analiza datos y genera información. Lo importante en él no es la calidad del dibujo sino la capacidad de relacionar los datos y extraer nueva información.

- *Importancia de los SIG para el análisis de los datos*

Incorpora los elementos esenciales de cartografía computacional y BD en un sistema.

La característica más importante de este sistema es que cada característica mapeada es linkeada a un registro en una BD tabular y también puede relacionarse a archivos en otras BD. En otras palabras, los SIG integran datos geográficos y tabulares totalmente.

Esta unión entre los mapas y los datos tabulares hace posible el análisis de los datos geográficos. La cartografía computacional permite la representación, pero no el análisis, de datos geográficos. Los manejadores de BD no contienen información geográfica. Integrando esas dos tecnologías en un SIG se crean "mapas inteligentes" con la habilidad de realizar análisis espacial. Tal análisis puede incluir preguntas espaciales ("dónde está?"), medidas espaciales ("a cuanta distancia se encuentra?"), o problemas más complejos (ruta mejor, correlación espacial, etc.).

Resumen

Un SIG no es una herramienta más de la cartografía, es un instrumento para la toma de decisiones en cualquier dominio en que la distribución espacial del dato sea una característica importante para la generación de nueva información.

2.2 TECNOLOGÍA ORIENTADA A OBJETOS

La tecnología de Orientación a Objetos surgió en la década del '60 pero recién tomo énfasis en la década del '80. Está basada en un conjunto de principios muy simples que buscan maximizar la productividad, la flexibilidad y el reuso de las aplicaciones y sus componentes. Las ideas de encapsulamiento, polimorfismo y herencia dan un conjunto de guías muy poderosas para construir aplicaciones sofisticadas en cualquier dominio de aplicación.. Las aplicaciones construidas con Objetos pueden interactuar eficientemente con bases de datos relacionales, con clientes y servidores de Internet, etc. Al mismo tiempo, las herramientas para construcción de aplicaciones usando Objetos han madurado notablemente. Dichas herramientas soportan el desarrollo de aplicaciones en entornos cliente-servidor, interacción con bases de datos y poseen facilidades para el desarrollo en grupos de desarrolladores.

2.2.1 *El paradigma de Orientación a Objetos*

La Orientación a Objetos es una metodología para el análisis y diseño del software que involucra la identificación de componentes auto-contenidos que pueden interconectarse para producir aplicaciones enteras. Estos componentes (objetos) tienen las características de encapsulamiento de funciones y datos, herencia de características de otros componentes y capacidad de intercomunicación mediante el envío de mensajes de un lado a otro.

Los sistemas Orientados a Objetos se crean a través de la definición de Clases, las cuales son luego utilizadas para la creación de los Objetos de la aplicación.

- **Clase:**

Una colección de uno o más Objetos con un conjunto uniforme de atributos y servicios, incluyendo una descripción de cómo crear nuevos Objetos de la clase. La estructura y comportamiento de Objetos similares se definen en su clase común. (Edward Yourdan)

- **Objetos o Instancias:**

Un Objeto es una entidad que tiene estado, comportamiento e identidad. El estado de un Objeto consiste en sus atributos y los valores actuales de estos atributos. El comportamiento de un Objeto consiste en las operaciones que pueden ser llevadas a cabo en él y sus consecuentes cambios de estado. La identidad de un Objeto es lo que se usa para distinguirlo de otros Objetos.

La diferencia entre una clase y un objeto se encuentra en el hecho de que la clase es la definición desde la cual se crean los Objetos pertenecientes a la misma. Los Objetos existen y se comunican entre si, pero las clases solo existen como cianotipos desde los cuales los Objetos son instanciados.

2.2.2. Características del Diseño Orientado a Objetos

1. Abstracción:

Es la habilidad de representar un problema complejo en un modelo simplificado. Aunque todas las metodologías de programación tienen esta habilidad, la Orientación a Objetos lleva esto al extremo.

2. Encapsulamiento:

Es la habilidad de ocultar dentro de cada objeto, los detalles de la implementación de su comportamiento. Cada objeto provee una interfaz pública para conocer sus atributos y utilizar sus métodos, evitando cualquier intento de su uso que no sea provisto por su interfaz pública.

3. Herencia:

La habilidad de un objeto de tomar toda o parte de su definición desde otro objeto.

4. Ligadura Dinámica:

Cuando un pedido es enviado a un objeto, la operación a llevarse a cabo depende de el pedido y del objeto que lo recibe. Diferentes Objetos que soportan las mismas operaciones pueden tener diferentes implementaciones de las mismas. Esta asociación en tiempo de ejecución entre un pedido a un Objeto y una de sus operaciones se conoce como Ligadura Dinámica. Ligadura Dinámica significa que un pedido no se liga a una implementación en particular hasta el momento de la ejecución. La característica de Ligadura Dinámica permite sustituir en tiempo de ejecución a Objetos que tienen la misma interfaz (soportan el mismo método). Esta capacidad de sustitución se conoce como polimorfismo.

5. Polimorfismo:

Es la habilidad de una operación de ser aplicada a varios Objetos. Existen dos clases de polimorfismo: inherente y AdHoc. El polimorfismo inherente describe la habilidad de una operación única de estar disponible para cada subclase bajo el árbol jerárquico de una clase. El polimorfismo AdHoc es la habilidad de que diferentes operaciones tengan el mismo nombre en diferentes clases.

6. Especialización:

Significa que cada clase en el árbol de definición de clases está más y más especializada en lo que respecta a sus operaciones.

7. *Comunicación entre Objetos:*

Toda actividad dentro de una aplicación consta de Objetos enviando y recibiendo mensajes desde y hacia otros Objetos.

8. *Reusabilidad:*

La idea de reuso de componentes no es exclusiva de la orientación a Objetos. Sin embargo, debido a las características anteriores, es considerablemente más fácil de obtener.

2.2.3 Patrones de diseño

Los patrones de diseño (PsD) constituyen una excelente forma de registrar experiencias de diseño, y reusarlas exitosamente en distintas arquitecturas. Una característica importante es que los PsD no se inventan, *se descubren*. Esto significa que para que un patrón sea tal, se deben probar sus usos previos y exitosos.

Un patrón de diseño (PD) sistemáticamente nombra, abstrae, explica y evalúa un diseño relevante y recurrente en aplicaciones OO [Gamma+95]. Cada PD enfoca un problema particular del diseño Orientado a objetos, describen cuando se aplica y las consecuencias de su uso. Los patrones capturan y hacen explícitas las decisiones de expertos en el contexto específico, de manera de poder reutilizar esa experiencia. Un patrón debe determinar su propósito y consecuencias de manera tal que al momento de desarrollar una nueva aplicación, un diseñador pueda elegir entre distintas alternativas de diseño que hacen a un sistema reusable, y evitar alternativas que comprometan su mantenibilidad, o que no se adecuen al problema específico.

En general, un PD tiene 4 elementos esenciales[Gamma+95]:

1. *Su nombre:* el nombre de un PD debe transmitir la esencia del mismo. Su elección es vital, ya que identificará al patrón en un nuevo vocabulario de más alto nivel, ha ser incorporado y compartido entre los desarrolladores, lo que permite comunicar toda una decisión de diseño con sólo un nombre. Encontrar un nombre adecuado para el PD puede ser, a veces, una tarea difícil.
2. *El problema relacionado:* generalmente, el problema describe la necesidad o no de aplicar el PD. A veces, el problema puede incluir una lista de condiciones que deben darse antes de que tenga sentido aplicar el PD para resolverlo.
3. *La solución:* la cual describe los elementos que realizan el diseño, sus relaciones, responsabilidades y colaboraciones así como la manera de disponer los objetos y clases participantes de manera de resolver el problema.
4. *Las consecuencias:* son los resultados obtenidos al aplicar el patrón. Aunque a menudo las consecuencias no son tomadas en cuenta al describir las decisiones de diseño, estas son críticas para evaluar alternativas de diseño y para entender los costos y beneficios de aplicar el patrón.

Ejemplo: El patrón COMPOSITE. [Gamma+95]

Objetivo:

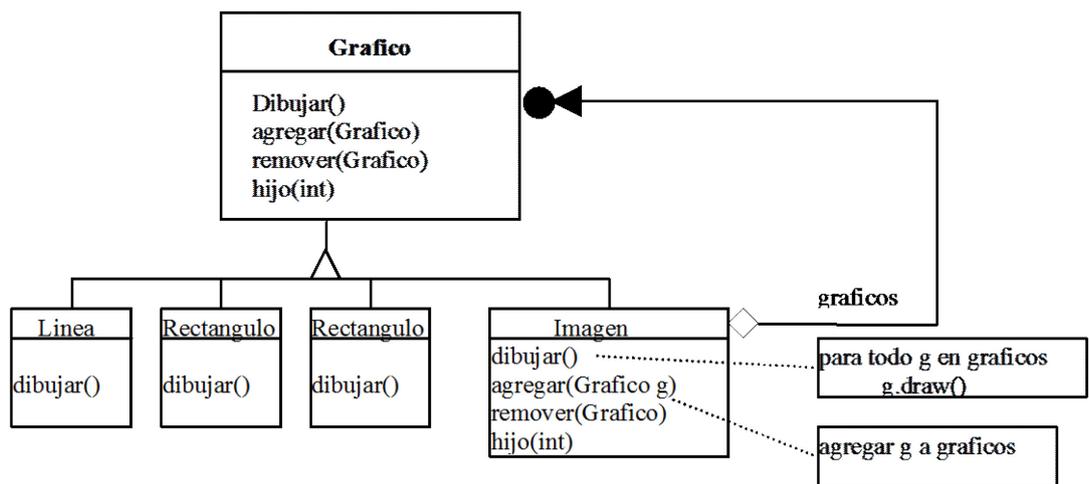
Componer Objetos para representar jerarquías cuyos elementos pueden ser objetos individuales o una composición de objetos . Este patrón permite a los clientes (los Objetos de la aplicación de lo utilizan) tratar a objetos individuales y a las composiciones de manera uniforme.

- Un caso en el que puede utilizarse:

Es sabido que toda aplicación gráfica nos provee la facilidad de agrupar o desagrupar 2 o mas elementos gráficos para obtener un grupo y manipularlo como una unidad e inclusive pueden existir grupos de grupos, etc.

Una primera aproximación de solución Orientada a Objetos seria definir clases básicas como Texto y Línea mas otras clases que actúan como contenedores de estas primitivas para formar los grupos.

El problema de la solución anterior reside en que el código que utiliza estas clases debe tratar de manera diferente a los objetos primitivos y a los contenedores, aunque el usuario los trata de manera similar la mayoría de las veces. Esto implica que se deben tratar estos objetos de manera diferente, lo cual hace mas compleja la aplicación. La aplicación del patrón Composite se basa en la utilización de la composición recursiva para que los clientes no tengan que realizar este tipo de distinciones.



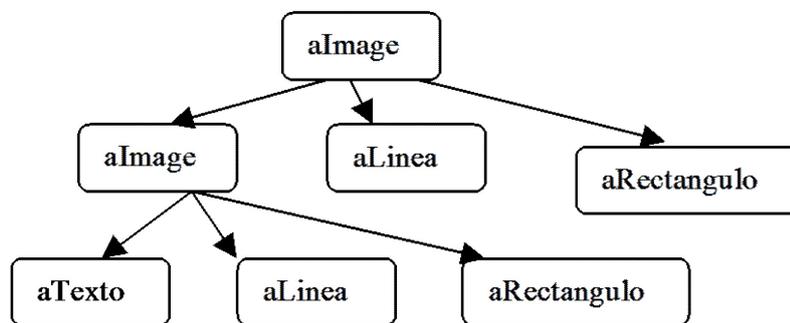
La clave del patrón es una clase abstracta que representa las primitivas y sus componentes. Para la aplicación de ejemplo, esta clase es Gráfico. La clase gráfico declara operaciones como Dibujar(), que son específicas a los objetos gráficos. También declara ope-

raciones que todos los objetos compuestos comparten, como las operaciones para acceder y manejar sus hijos.

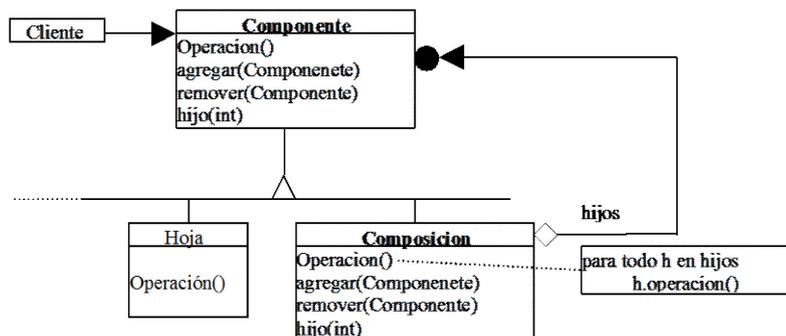
Las subclases Línea, Rectángulo y Texto definen a los objetos gráficos primitivos. Estas clases implementan el método Dibujar() para lograr sus respectivas representaciones gráficas. dado que los gráficos primitivos no tienen hijos gráficos, ninguna de ellas tienen operaciones para manipular hijos.

La clase Imagen define una agregación de objetos Gráfico. Imagen implementa Dibujar() llamando a los Dibujar() de sus hijos y operaciones para manejarlos. Dado que la interface de Imagen es igual que la de Gráfico, los objetos imagen pueden componer Imágenes recursivamente.

El siguiente diagrama muestra una estructura Composite típica con objetos gráfico recursivamente compuestos.



Estructura General del Patrón



- **Posibles Aplicaciones:**

El patrón Composite puede ser utilizado cuando:

1. se quieren representar jerarquías parciales-totales de objetos o
2. se quiere que los clientes sean capaces de ignorar las diferencias entre composiciones de objetos y objetos individuales. Los clientes trataran uniformemente a todos los objetos de la estructura compuesta.

2.3 SISTEMAS DE INFORMACIÓN GEOGRÁFICOS ORIENTADOS A OBJETOS (SIGOO)

Hasta este punto se ha tratado el tema de los sistemas de información geográficos (SIG) y explicado el paradigma Orientado a Objetos.

En el desarrollo de esta tesis se utilizó este paradigma para la implementación de los Sistemas de Referencia. Ahora bien, ¿cuáles serían los beneficios reales de usar tecnología Orientada a Objetos para SIG?

Al utilizar cualquier facilidad provista por un SIG (por ej. utilidades de medición de aguas, electricidad, caminos, áreas, etc.), se tiene relaciones geográficas inherentes que definen la ubicación del elemento en cuestión, y la manera en que cada elemento afecta el estado de otros elementos. Así, nos encontramos ante la necesidad de modelar una Conectividad implícita entre elementos.

La mayoría de los SIG procedurales no tienen en cuenta esta conectividad, lo cual significa que no existe una unión entre estos elementos almacenados en la Base de Datos y es el desarrollador quien tiene que mantener esta conectividad a través del código de la aplicación.

En un ambiente Orientado a Objetos, todo objeto tiene atributos (valores como diámetro, tamaño, referencias a otros objetos), topología (jerarquía de clases) y flexibilidad en la manera en que se conectan a otros objetos. La facilidad de definir la conectividad en un ambiente orientado a objetos hace que la modelización de las entidades del mundo real (y sus operaciones) se vean ampliamente simplificadas.

Las características de un objeto se definen por su pertenencia a una clase. Las clases de objetos generalmente modelan un tipo de objeto del mundo real. Nuevos tipos de objetos pueden ser creados simplemente tomando una definición existente de clase y especializando a través de la herencia. Además, las operaciones de los usuarios pueden ser definidas en clases abstractas con sus métodos asociados. Estos métodos definen y realmente ejecutan las operaciones del usuario.

2.3.1. Beneficios de la orientación a objetos para un SIG:

El uso de tecnología orientada a objetos en SIG, provee 2 tipos de beneficio:

- ✓ Los beneficios generales obtenidos por el uso del paradigma, como la herencia de comportamiento.
- ✓ Aquellos relacionados con la aplicación SIG en particular

Sumarizando:

1. **Corrección:** la habilidad de llevar a cabo las especificaciones de manera adecuada. Esta característica se logra de manera mas simple si se definen los comportamientos adecuados para cada clase.
2. **Reusabilidad:** la habilidad de usar y reutilizar el mismo “código” es especialmente valorable, en los casos en que se tienen comportamientos similares. Es muy importante lograr escribir y testear dicho código una sola vez, para poder luego ser invocado como una parte mas del objeto y evitar, así, las copias u otras tareas propensas de error.

3. **Extensibilidad:** la cual se define como la habilidad de adaptación a los cambios en las especificaciones. En un ambiente Orientado a Objetos, la introducción de cambios es generalmente cambiar el comportamiento de los objetos involucrados, lo cual hace más fácil planificar el cambio y predecir los efectos que pueda ocasionar.
4. **Robustez:** la habilidad de funcionar bajo condiciones anormales tiene mucho que ver con lo que el analista olvidó incluir en la especificación del comportamiento de los objetos o lo que el programador no tuvo en cuenta al escribir los métodos. La naturaleza auto-contenida de la definición de un objeto tiende a limitar el daño solo al objeto o comportamiento y no es probable que el error se propague a través del sistema, como es el caso de la programación procedural.
5. **Compatibilidad:** esta característica no es exclusiva de la programación Orientada a Objetos. Sin embargo, la habilidad de conectarse a través de protocolos standard está predefinida en la mayoría de los sistemas Orientados a Objetos. Dado que muchos atributos de distintos objetos ya existen en una base de datos NO-GIS, la posibilidad de utilizarlos a través de llamadas SQL es importante.
6. **Eficiencia:** es la habilidad de utilizar recursos de Hard y Red con mínimo desperdicio, lo cual significa velocidad y respuesta para el usuario. Dado que los “programas” Orientados a Objetos son ensambles de métodos, es más fácil re-ensamblarlos y ajustar la performance.

2.3.2. Comparación en el uso de bases de Datos:

Generalmente, un sistema de información geográfico SIG utiliza una combinación de archivos gráficos y una BD relacional. Algunos van más allá guardando tanto gráficos y datos en una BD relacional. Considerando las BD relacionales y SIG, usar una metodología orientada a objetos teóricamente, posee por lo menos estas ventajas:

- *Integridad de los datos:* puesto que se mantienen gráficos y datos en la misma estructura de datos, la integridad se asegura (Wang, 1992). No hay riesgo de haber perdido links, o anular datos sin anular el gráfico, o vice-versa;
- *Modelización más flexible y más fuerte:* como se mencionó antes, la modelización orientada a objetos lleva la aplicación física más cerca al diseño lógico. Existe una definición y manejo más flexible de las relaciones entre los objetos llevando al modelo lógico más cerca de la aplicación real;
- *Almacenamiento compacto:* dado que la estructura del objeto contendrá sólo la información que es actualmente la actual, ningún espacio en blanco se guarda, y se espera un mínimo desperdicio de espacio de disco, así como las mejoras en la performance global del sistema;
- *Amigable al usuario:* es mucho más natural para el usuario entender y consultar datos estructurados con el modelo orientado a objetos, dado que ninguna de las transformaciones requeridas por el modelo relacional tiene que ser implementadas.

2.3.3. Modelización de BD para SIGOO

DBMSs convencionales generalmente trabajan muy bien con datos tabulares en los que la estructura es muy simple. Éste no es el caso para los datos utilizados en los SIG como son los datos espaciales. Al usar un DBMS convencional para manejar datos espaciales, se obliga a menudo a los usuarios a que transformen los datos idealizados para que encajen en los requisitos del modelo. Debido a estas transformaciones, la actuación del sistema se degrada y se pone inaceptable cuando se emplean grandes cantidades de datos (Egenhofer y Frank, 1992).

Cuando los usuarios de los SIG necesitan a menudo integrar datos de fuentes diferentes, usan el común denominador más bajo normalmente disponible: *la situación geográfica*. Por consiguiente, se exigen modelos de datos flexibles para acomodar la heterogeneidad, mientras se mantiene la posibilidad de realizar varias tareas usuales.

La creación de un modelo de datos para SIGOO (*sistemas de información geográfico orientado a objetos*) todavía es territorio principalmente desconocido. Las técnicas usuales para el diseño de modelos como entidad-relación muestran rápidamente sus limitaciones cuando se enfrentan a la necesidad de modelar datos espaciales.

2.3.4. Estimación de tiempo y costos

La ventaja de utilización de Objetos es la mayor productividad obtenida en comparación con los GIS procedurales convencionales, debido a que los objetos pueden ser tomados directamente de una clase ya existente o especializar una clase desde una más general, utilizando la herencia. Bajo estas circunstancias, la construcción de la solución no sólo es significativamente más rápida en comparación con la procedural, sino que, en particular, ganancias significativas en fiabilidad se alcanzan debido al reuso de las clases. Esto reduce el costo en las fases de testeo, corrección y subsecuente mantenimiento.

Sin embargo, para obtener una ganancia productiva a través del reuso de clases, deben existir clases disponibles para re-usar, lo cual solo puede lograrse si se realiza un cuidadoso y extensivo diseño. Esto asegura el máximo de fiabilidad y flexibilidad a la hora del reuso.

El acercamiento Orientado a Objetos hace que la comunicación entre los componentes del sistema sea simple y efectiva. Esto deja al equipo de desarrollo más tiempo para la interacción con el usuario, para ajustar sus necesidades.

Dado que las clases pueden modelar entidades del mundo real de manera simple y cercana, la solución es inmediatamente entendible para el desarrollador y el usuario.

Como atributo final, la habilidad de modelar complejas reglas del dominio de la aplicación como comportamiento de los objetos de las clases, ha permitido la adopción de técnicas de solución altamente modulares, obteniendo soluciones que proveen un alto grado de re-uso de sus componentes

3. Sistemas de Referencia

3.1 INTRODUCCIÓN A LOS SISTEMAS DE REFERENCIA

La difusión de los SIG como herramienta cotidiana ha introducido en el lenguaje cotidiano de sus usuarios la palabra *georreferenciar*.

En un sentido abstracto, *georreferenciar* significa asignar algún tipo de coordenadas ligadas al terreno a los objetos de interés, sean estos naturales, obras de ingeniería, los vértices de una parcela, etc.

Adquirir la información que integrará la base de datos, es la operación más laboriosa y costosa involucrada en el desarrollo de un SIG. Georreferenciar dicha base de datos, por el contrario, parecería ser una operación sencilla, sin embargo, si se piensa que las coordenadas son el vehículo que permite que distintos usuarios superpongan en el SIG la capa de información de su interés, se advierte que de una representación correcta depende, en gran medida, el aprovechamiento que pueda hacerse del SIG y multiplica su valor de mercado por cuanto lo hace útil a un mayor número de usuarios.

La expresión *georreferenciación correcta* involucra varias condiciones, como por ejemplo, que los procedimientos de medición y cálculo en base a los que se obtienen las coordenadas cumplan con ciertos estándares pre-establecidos [POSGAR.Brunini y Moirano].

La georreferenciación se logra a través de un Sistema de Referencia, por ello para poder cumplir con lo mencionado anteriormente, las coordenadas deben estar vinculadas al mismo Sistema de Referencia o, de lo contrario, contar con una manera de conocer, dada una coordenada en un Sistema de Referencia, que coordenada representa en otro Sistema.

Existen muchas maneras de referenciar una entidad, por consiguiente existen muchos Sistemas de Referencia.

Algunas definiciones de Sistemas de Referencia son:

“...Un sistema de referencia es una manera de asignar valores a una ubicación. En el sentido más general, un sistema de referencia puede ser pensado como una escala de medida. Esto puede incluir escalas cardinales y ordinales o vocabulario descriptivo. Las escalas pueden ser discretas o continuas, lineales o cíclicas. Un sistema de referencia espacial es una función que asocia a una entidad una tupla de coordenadas en el espacio matemático. Un sistema de referencia temporal es una función que asocia tiempo a una tupla matemática...” [OpenGIS]

“...Un Sistema de referencia ubica unívocamente un objeto sobre la superficie terrestre...”

“...Un Sistema de Referencia le asigna a una entidad, sentido de ubicación geográfica, así como también define el conjunto de operaciones que se pueden aplicar a una o entre muchas entidades...”

Acerca de las definiciones mencionadas anteriormente, cabe destacar la propuesta por OpenGis, si bien menciona un sistema de referencia espacial también expone otro tipo de sistema que es el temporal, en esta tesis solo trataremos a los sistemas espaciales puesto que son los más utilizados y complejos, los temporales podrían ser motivo de otra contribución.

Seguidamente se explicará la manera en la que funcionan los sistemas de referencia, dándose previamente algunos conceptos de geodesia necesarios

3.2 LA FIGURA DE LA TIERRA

Históricamente, los Griegos fueron los primeros interesados en la forma y figura de la tierra. La idea de su esfericidad fue primeramente postulada por Pitágoras, pero Eratóstenes fue el primero en hacer cálculos (bastante precisos para la época) del tamaño de dicha esfera. En los comienzos se consideraba a la Tierra como una esfera perfecta, pero Newton con su profunda intuición de las leyes de la Naturaleza, sentó la hipótesis de que la forma de la Tierra, por efecto de la fuerza de gravitación y de la centrífuga debida a la rotación, debe ser un elipsoide achatado en los polos. En 1743, la expedición de la Academia Francesa al Perú y Botnia confirmó esta teoría de Newton. Posteriormente, se encontraron algunas diferencias que deformaban el elipsoide terrestre, y las verdaderas dimensiones y forma de la tierra constituyen una discutidísima cuestión de Geodesia Superior.

Además de esto, la superficie o *Topografía* de las masas de la tierra muestran grandes variaciones verticales debido a las diferencias de altura entre las distintas formas topográficas existentes (montañas, valles, llanuras, etc.). Así, la tierra es una figura geométrica compleja.

3.2.1. El Geoide

Dado que la superficie terrestre, como la conocemos, es todo menos uniforme, es imposible aproximar la forma de la misma con un modelo matemático relativamente simple.

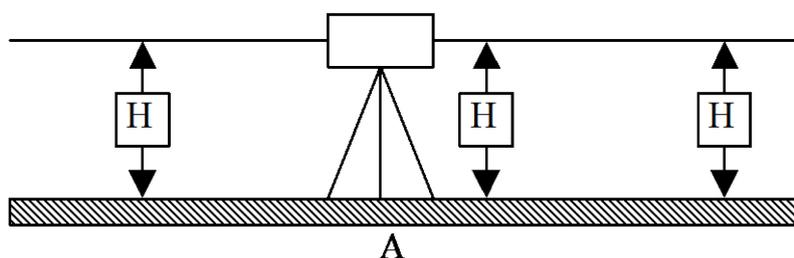


Figura 3.2.1 La Superficie de Nivel sobre una distancia corta

La solución fue simplificar los problemas removiendo la topografía terrestre sobre el nivel medio de los mares y considerar la superficie resultante como el *nivel*. Nivel en este contexto significa la posibilidad de mirar en distancias cortas y en todas las direcciones encontrando que la altura de la línea visual es la misma. Esta línea visual se denomina *línea de nivel* y está situada en lo que se denomina punto de observación.

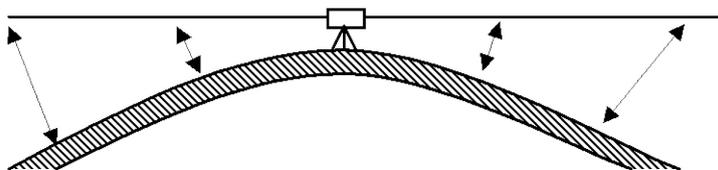


Figura 3.2.2 La Superficie de Nivel sobre una distancia larga

Cualquier línea de nivel en una observación tiene las siguientes características:

- Es perpendicular a la dirección de gravedad en el punto de observación.
- Su dirección será diferente a cualquier otra línea perteneciente a otro punto de observación.

Si se toman infinitas observaciones y en cada una se rotan las líneas de nivel sobre la local vertical, esto formará una “superficie continua de nivel”, la cual está intrínsecamente atada a la gravedad.

Esta superficie de nivel se llama *Geoide* y puede ser definido como la superficie equipotencial que corresponde al nivel medio de los mares. Dado que el geoide es una superficie equipotencial el potencial gravitacional en cualquier punto sobre él será el mismo y la dirección de gravedad será perpendicular a la línea de nivel en cada punto.

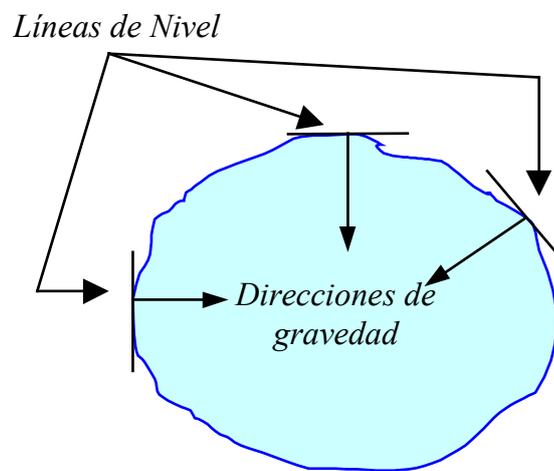


Figura 3.2.3. Líneas de Nivel

Debido a esto, para los puntos que presenten una deficiencia de masas el geoide presentará depresiones, mientras que cuando exista exceso de masas, el mismo presentará elevaciones.

3.2.2. Elipsoide

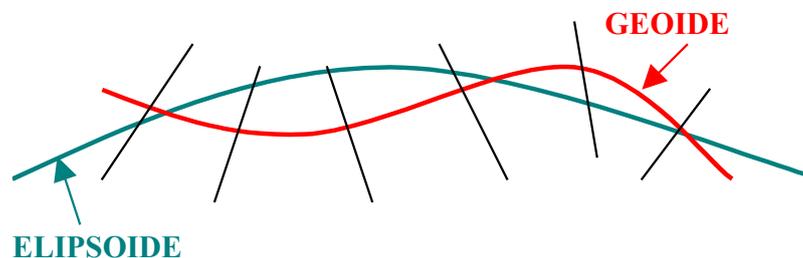


Figura 3.2.4. Efectos de las anomalías de masas sobre el Geoide

Si se desean llevar a cabo cálculos de posiciones, distancias, direcciones, etc. , sobre la superficie de la tierra, es necesario disponer de alguna superficie de referencia matemática.

La más conveniente figura de referencia es el elipsoide ya que es una figura relativamente simple que encuadra con el geoide en un primer orden de aproximación.

Los elipsoides utilizados son elegidos tales que concuerden lo mejor posible con el geoide en la zona de estudio.

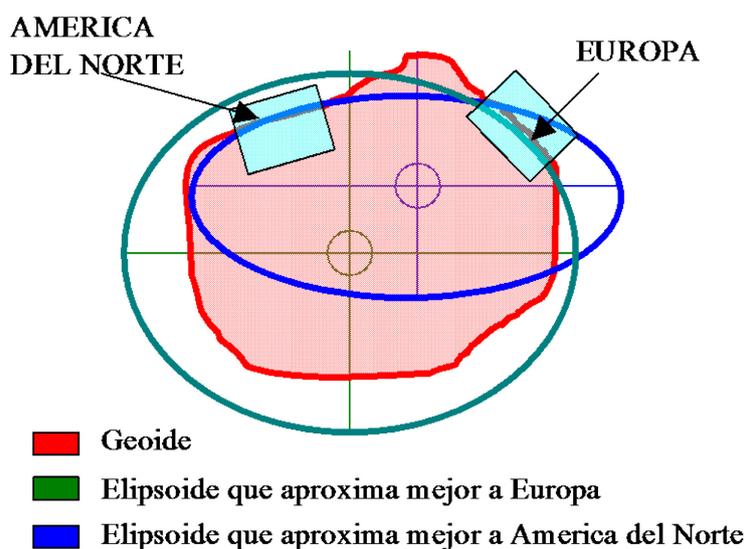


Figura 3.2.5. Geoide y dos elipsoides

El elipse que define un elipsoide o esferoide, es llamado meridiano elipsoidal. Un elipsoide es formado por la rotación de un elipse sobre su eje menor. Una elipse puede ser definido por su semi-eje mayor y su achatamiento. El achatamiento(f) depende del semi-eje mayor(a) y el menor(b).

$$f = \frac{a-b}{a}$$

También se puede definir un elipsoide con la excentricidad(e) y su semi-eje mayor.

$$e^2 = 2f - f^2$$

Como se ha enunciado antes, se pretende que el elipsoide aproxime lo mejor posible a la superficie de la tierra. Para ello, no basta con definir solamente su forma y tamaño, también hay que darle una orientación y desplazamiento, por esto surge el concepto de Datums. Un datum esta formado por un elipsoide orientado y desplazado con respecto a un origen (este concepto será ampliado en la sección Datums).

Parámetro	Internacional de 1924	WGS84	Internacional de 1980 (GRS80)
a[m]	6378388	6378137	6378137
b[m]	6356912	6356752.31424	6356752.31414
f	3.36700x10 ⁻³	3.352810664x10 ⁻³	3.352810681x10 ⁻³

Tabla 1. Parámetros de varios elipsoides usados en nuestro país

3.3 DATUMS

3.3.1. Introducción

En la sección anterior se mencionó la manera de aproximar la figura terrestre, ya sea con un elipsoide o con un geoide. Pero para lograr esta aproximación no basta con esas figuras, sino que es además necesario un marco de referencia para poder ubicarlas y orientarlas de la manera correcta.

Así, un *datum* está formado por un elipsoide orientado y desplazado con respecto a un origen.

$$\text{Datum} = \text{elipsoide} + \text{marco de referencia}$$

Existen muchos marcos de referencia y cada uno depende de la zona de la tierra que se desee aproximar y del elipsoide elegido. Pueden existir marcos de referencia globales como así también locales.

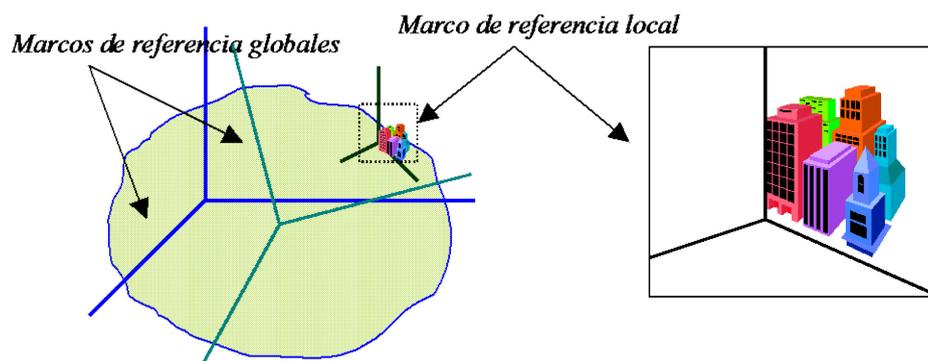


Figura 3.3.1 Diferentes marcos de referencia

El Servicio Internacional de Rotación de la Tierra (IERS) y la Unión Geodésica y Geofísica Internacional definieron un marco de referencia que cumple con las siguientes características:

- El origen del marco de referencia se encuentra en el centro de masa del planeta
- El eje Z apunta al polo norte
- El eje X pasa por el meridiano de Greenwich

Estas convenciones definen un marco de referencia fijo a la tierra, denominado sistema de referencia Terrestre Internacional (ITRS ó ECEF:Earth).

A continuación se expondrá un resumen de un texto que explica 2 de los marcos de referencia utilizados por nuestro país.

3.3.2 Marcos de referencia [Brunini-Rodríguez]

El sistema de referencia* terrestre internacional (ITRS) consiste básicamente en una terna rectangular cuyo origen se encuentra en el geocentro del planeta y cuyos ejes se hallan perfectamente alineados. Esta es una definición rigurosa pero abstracta, pues tanto el origen como los ejes pueden dibujarse sobre una hoja de papel pero son inaccesibles en la práctica. A los fines prácticos un sistema de referencia* se materializa mediante un conjunto de monumentos geodésicos o mojones diseminados en una región, a los que se les han asignado coordenadas. Es claro que tales coordenadas han surgido de un cierto proceso de medición, estando por lo tanto afectadas de errores. A causa de estos errores, las marcas geodésicas materializan al marco de referencia ideal con un cierto grado de exactitud. Se establece entonces la diferenciación entre el marco de referencia ideal o sistema de referencia* y el marco de referencia (su materialización práctica aunque inevitablemente imperfecta). Cuando se asigna coordenadas a un punto en un marco de referencia determinado, lo que se está haciendo es interpolar entre las coordenadas de las marcas más cercanas, trasladando al punto cuyas coordenadas se están determinando, los errores de aquéllas más los que agrega el procedimiento de medición utilizado.

El marco de referencia WGS84

Si bien existen distintas materializaciones del ITRS, la más difundida entre los usuarios de GPS es el Sistema Geodésico Mundial de 1984 (WGS84), ya que a este marco de referencia se vinculan las coordenadas cartesianas geocéntricas que proveen los receptores GPS. WGS84 fue establecido por la Agencia Cartográfica de Defensa (DMA) de los Estados Unidos de América y en la actualidad es mantenido por la Agencia de imágenes y Cartografía (NIMA) de ese país. Está materializado por medio de 10 estaciones de rastreo permanente. Se sabe que WGS84 materializa al ITRS con una exactitud a los 10 centímetros.

El marco de referencia PGA94

Puesto que WGS84 está materializado por solamente 10 puntos en todo el planeta, para que los usuarios de GPS puedan acceder a este marco de referencia en la modalidad de posicionamiento diferencial, en la Argentina se realizó una densificación conocida como POSGAR94 o, PGA94 (Posiciones Geodésicas Argentinas de 1994). El marco de referencia PGA94 está materializado por 127 puntos distribuidos en todo el país, de tal modo que un usuario GPS encuentre, casi siempre, un punto dentro de un radio de aproximadamente 100 km. En la gran mayoría de los casos, los puntos son de fácil acceso. El Instituto Geográfico Militar dio a conocer las coordenadas PGA94 a comienzo de 1995 y en mayo de 1997 lo adoptó oficialmente como marco de referencia geodésico nacional.

Puede afirmarse que PGA94 reúne las siguientes características:

- 1) es un marco de referencia geocéntrico y preciso garantizado por los controles y las verificaciones realizados por los expertos en la materia y con puntos distribuidos en todo el país;
- 2) existen compromisos de mantenimiento y perfeccionamiento;
- 3) está vinculado al marco continental SIRGAS (Sistema de referencia Geocéntrico para América del Sur)
- 4) se encuentran disponibles los parámetros de transformación con respecto a campo Inchauspe 1969;

- 5) tiene el apoyo de las entidades de fomento en la materia (agrimensura, cartografía y geodesia).

Estas características lo convierten en un marco de referencia apto para soportar el intercambio de información georreferenciada en escala nacional y global; para el desarrollo de los sistemas de información geográfica y territorial sobre una base de referencia única; y para la informatización de la cartografía.

*** El concepto utilizado por el autor como sistema de referencia abarca nada más que la ubicación de los ejes para orientar a la figura con que se desea aproximar a la tierra.**

El concepto de sistema de referencia empleado en esta tesis, como se irá planteando en capítulos posteriores, no abarca nada más que eso, es un concepto más amplio e incluye no solo el posicionamiento de los ejes sino también a la figura que aproximará a la tierra, el sistema de coordenadas empleado y el factor de escala elegido (ver modelo de sistema de referencia). La mención que hace el autor a sistema de referencia se toma como marco de referencia ideal.

3.3.3. Transformaciones entre distintos Datums

Primeramente es muy importante tener en cuenta que toda transformación entre datums introduce errores en las coordenadas resultantes, como se explicará luego dicha transformación necesita para su realización una serie de parámetros, que son los que poseen dichos errores. No se gana exactitud al transformar coordenadas de un Datum poco preciso a otro muy preciso, sino que las coordenadas transformadas conservan o incluso empeoran la exactitud que tenían en el original. Por esto, es muy importante contar con una buena y precisa definición de los parámetros involucrados en el Datum a utilizar por el usuario.

La transformación entre diferentes Datums involucra una diferencia de origen que puede expresarse con 3 parámetros, diferente orientación de los ejes que puede expresarse con 3 ángulos que definen la orientación relativa de cada uno de los ejes, y un factor de escala. En definitiva hacen falta 7 parámetros para definir completamente la relación entre 2 sistemas rectangulares espaciales.

Hasta este momento hay que destacar 2 puntos para poder continuar:

- Como se mencionó, la definición de un Datum involucra 3 parámetros para la orientación y 3 parámetros para la diferencia de origen. Ahora bien, esa orientación y ese desplazamiento se van a realizar en base a quien?, es decir, hay que nombrar un marco de referencia base a partir del cual todos calculen sus parámetros, porque sino no hay manera de poder comparar y realizar operaciones entre ellos (salvo contar con la in-

formación de orientación y desplazamiento de uno con todos los demás existentes, pero esto es imposible porque podrían existir infinitos Datums). Por convención se toma que el Datum WGS84 posee sus 6 parámetros en 0 y todos los demás calculan su desplazamiento y orientación en base a éste.

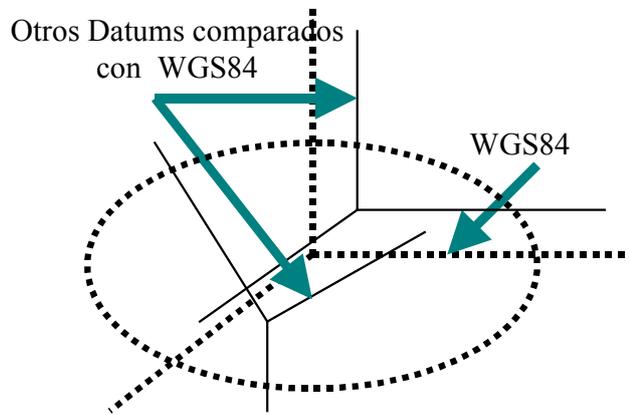


Figura 3.3.2 El Datum base WGS84

- Otro punto a tener en cuenta es que si bien con estos 7 parámetros alcanza, no hay que olvidarse de 2 parámetros más que son los del elipsoide: el eje mayor (a) y el achatamiento (f). Dichos parámetros van a entrar en juego cuando haya que realizar un cambio de datums con coordenadas geodésicas (las cuales dependen del elipsoide). Tal transformación será explicada en la sección de Sistemas de Coordenadas, ahora se enunciará nada más que la transformación de Datums con sistemas de coordenadas rectangulares, la cual pasa a ser una transformación entre marcos de referencia.

Transformación de Helmert

Este es el método más general de transformación de coordenadas entre diferentes Datums, utiliza los 7 parámetros mencionados anteriormente: traslaciones del origen, rotaciones de los ejes y un cambio de escala. Dicha fórmula es expresada como:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} \mu & \theta_z & -\theta_y \\ -\theta_z & \mu & \theta_x \\ \theta_y & \theta_x & \mu \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}$$

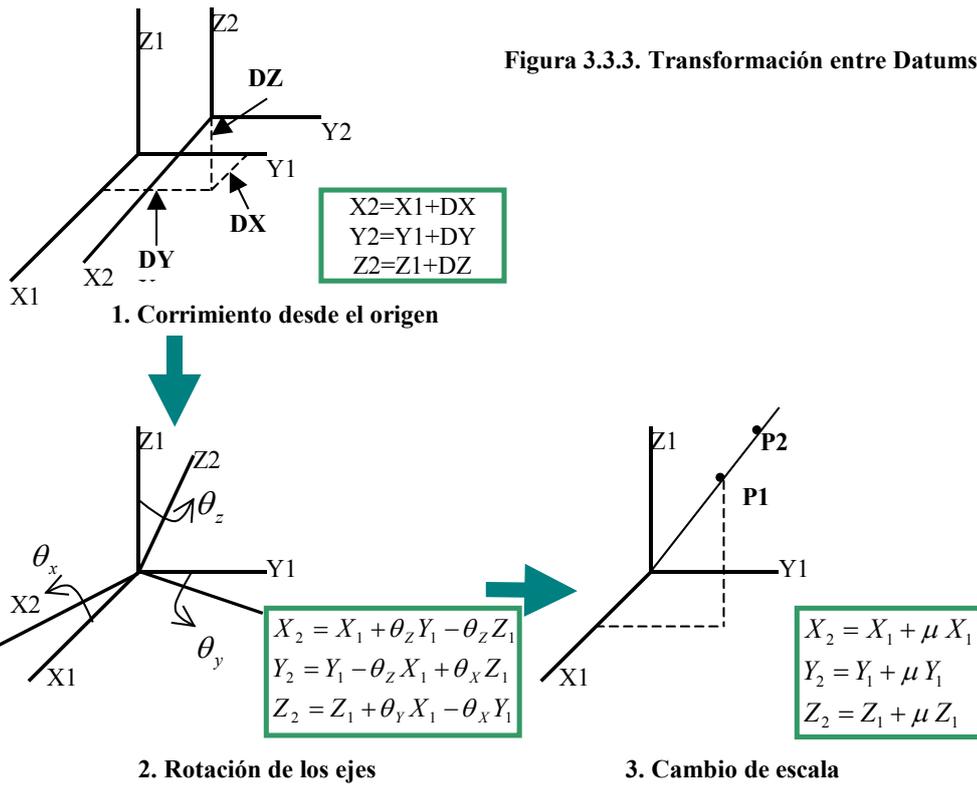
donde X_1, Y_1, Z_1 son las coordenadas del primer datum

X_2, Y_2, Z_2 son las coordenadas del segundo datum

$\theta_x, \theta_y, \theta_z$ son la rotación de los ángulos entre los datums

$\Delta X \Delta Y \Delta Z$ son la diferencia de origen entre los datums

Figura 3.3.3. Transformación entre Datums



3.4 PROYECCIONES CARTOGRÁFICAS

3.4.1 Introducción

Como se ha mencionado en las secciones previas, el espacio geográfico entendido como la superficie terrestre total es un espacio curvo, que puede ser materializado por un elipsoide de referencia que representa al geoide.

Es muy fácil trazar sobre este elipsoide o esfera un sistema de paralelos y meridianos, pero su representación en un plano requiere un estudio especial ya que la superficie esférica no puede desarrollarse sobre un plano, sin que se deforme o se rompa. Si se trata de representar en un mapa una pequeña parte de la superficie terrestre, por ejemplo, una zona de 250 a 300 kilómetros cuadrados, la deformación escasamente sobrepasa los límites del estiramiento del papel; pero, tratándose de mapas de mayor extensión, como de naciones enteras, hay que resolver el problema de manera diferente.

Son varios los métodos seguidos para vencer esta dificultad. El más sencillo consiste en rodear la esfera con un cilindro o con un cono, o en colocarla tangencialmente a un plano y proyectar una parte de la red de meridianos y paralelos desde el centro de la esfera o desde otro punto convenientemente elegido sobre el cilindro o el cono a lo largo de una generatriz y extendiéndolo sobre el plano se tiene un sistema de meridianos y paralelos resultado de una verdadera proyección.

En la práctica son pocas las proyecciones trazadas conforme a este método. La mayoría de las empleadas son modificaciones de proyecciones geométricas y en muchos casos lo son en tal grado que apenas si se conserva algo de las relaciones y proporciones originales. Por estas razones no debiera emplearse el término “proyección” para designar el reticulado plano de meridianos y paralelos, pero la costumbre ha consagrado su uso de modo universal, por lo cual seguiremos empleándolo en lo sucesivo.

Así, pues, puede definirse una proyección diciendo que es un sistema plano de meridianos y paralelos sobre el cual puede dibujarse un mapa.

De una manera más general, *una proyección o sistema de proyección establece una correspondencia entre los puntos del elipsoide y los del plano.*

3.4.2 Propiedades de las proyecciones

Se dice que una proyección es *equivalente* cuando una zona o extensión cualquiera, grande o pequeña, tiene la misma superficie en el plano que en una esfera a igual escala. No puede conseguirse esta igualdad de superficie sin que haya una deformación considerable que hace variar grandemente los ángulos reales, sobre todo en los bordes de la extensión representada. Los mapas equivalentes son buenos para poner de relieve la distribución de productos en los estudios económicos o industriales.

Las proyecciones *conformes* son aquellas en que cualquier parte de poca extensión tiene la misma forma en el plano que en la esfera; es decir, que un rectángulo en la esfera está representado por un rectángulo en el plano; la relación entre las longitudes de meridianos y paralelos en el plano es igual a la misma relación en la esfera. Aparentemente esta propiedad es de gran valor, pero en la práctica tales proyecciones entrañan

un cambio de superficies considerable. Las proyecciones conformes son especialmente convenientes en las aplicaciones en que la conservación de ángulos y direcciones es condición indispensable; su principal uso se tiene en la navegación. Evidentemente no hay una proyección que sea al mismo tiempo *equivalente y conforme*; únicamente en una esfera se cumple esta condición; pero no es posible desarrollar un globo sobre un plano sin deformación.

3.4.3 Elección de un sistema de proyección adecuado

“...Son centenares los medios de trazar o construir sistemas de proyecciones, pero no todos son igualmente aceptables, ya que unos se prestan bien para una cierta aplicación y otros para otras. Sin embargo, no puede decirse que para cada mapa exista una determinada proyección como la mejor de todas, por lo cual el cartógrafo tiene que saber escoger el sistema más apropiado para cada caso...” [Cartografía. Erwin Raiz].

La elección del sistema de proyección más conveniente para una mapa determinado es una decisión difícil, que depende de varias variables. Para grandes escalas y pequeñas extensiones, la elección de la proyección es bastante indiferente; basta generalmente con atenerse a los mapas locales, o a sus derivados. Pero el problema se presenta cuando se trata de mapas a pequeña escala, particularmente en mapas del mundo entero. En este caso, la elección está condicionada sobre todo por el uso que va a hacerse del mapa, y puede basarse en diferentes cualidades (ver cuadro 3.4.1).

Si sobre el mapa deben efectuarse medidas o comparaciones de superficies, se utilizará con preferencia una proyección equivalente, aunque las deformaciones marginales sean a menudo un inconveniente superior a las ventajas que esta proyección ofrece. Si el mapa debe servir para contrastar formas o medir ángulos, será mejor elegir una proyección conforme. La navegación marítima y la aérea plantean otras exigencias que, además, son diferentes para cada una. Una loxodrómica (línea que corta a los meridianos bajo un ángulo constante) es cómoda de trazar en la proyección de Mercator, lo mismo que una ortodrómica (arco de un círculo máximo) en proyección gnomónica; pero la navegación a la vista o la radionavegación, la navegación en regiones polares o en las ecuatoriales, no requieren los mismos documentos. Incluso la configuración de los países a representar, su alargamiento en el sentido de la longitud o de la latitud, y su posición sobre el elipsoide, son también factores a tener en cuenta, si se quiere llegar a una elección razonable.

“...El cartógrafo, aún cuando no intervenga él mismo en la construcción de la proyección, no puede dejar de preocuparse de este tema. Debe saber en todo caso que, entre mapas que hayan sido construidos en distintos sistemas de proyección, aunque tengan la misma escala, no es posible efectuar ninguna comparación, ninguna medida común, ninguna unión, ninguna superposición...”[La cartografía de Fernand Joly]

En estos últimos párrafos se destaca la importancia de brindar al usuario la capacidad de poder utilizar una proyección adecuada según sus necesidades, esta característica deberá estar presente en todo SIG. Además se menciona que el usuario SIG “debe tener conciencia de que no puede trabajar con 2 conjuntos de datos provenientes de 2 sistemas de proyección distintos”.

En esta tesis no solo se planteará una solución para poder incorporar de una manera fácil la propiedad de manejar y construir sus propios sistemas de proyección en todo SIG orientado a objetos, sino que se solucionará el segundo problema; “el usuario podrá trabajar con 2 conjuntos de datos distintos, puesto que el sistema realizará la conversión y realizará los cálculos apropiadamente”, esto eliminará el problema de incompatibilidad de datos y hará más fácil y amigable la utilización del SIG para usuarios no experimentados.

	Proyeccion	Características principales	Aplicaciones
C O N F O R M E S	Mapas planos	Coordenadas rectangulares equidistantes	Antigüedad. Planos
	Estereográfica	Todo círculo de la esfera se proyecta según otro círculo. Escala constante sobre todos los círculos que tienen por centro el centro de proyección	Regiones polares. Mapamundis. Cielo
	Mercator	Coordenadas rectangulares. Paralelos más espaciados, para latitudes crecientes. Escala variable con la latitud. Las loxodrómicas se proyectan según rectas.	Cartas náuticas. Planisferios. Bajas latitudes.
	Mercator transversa. UTM	Coordenadas en curvas trascendentes ortogonales. El Ecuador y el meridiano central son rectas perpendiculares,	Latitudes medias
	Cónica de Lambert	Meridianos rectilíneos concurrentes. Paralelos concéntricos. Escala invariable sobre el paralelo tangente	Mapas a gran escala. Latitudes medias
	Acimutal de Lambert	Meridianos rectilíneos, concurrentes en el polo.	Regiones polares. Hemisferio norte
	Cónica de Lambert	Meridianos rectilíneos concurrentes. Paralelos concéntricos.	Mapas regionales
E Q U I V A L E N T E S	Bonne	Paralelos concéntricos equidistantes. Los meridianos se proyectan según curvas trascendentes	Mapa de Francia a 1:80.000
	Mollweide	El Ecuador y el meridiano central son rectas perpendiculares. Los paralelos son rectas paralelas al Ecuador. Los meridianos se proyectan según elipses.	Planisferios

Cuadro 3.4.1 Principales sistemas de proyección

3.4.4. Clasificación de las proyecciones

La superficie sobre la cual se efectúa la proyección puede ser un plano o una superficie desarrollable como un cilindro o cono, que luego se corta a lo largo de una generatriz y se desarrolla sobre un plano. Podemos distinguir entonces:

1. **Proyecciones acimutales o cenitales:** proyectar sobre un plano tangente a la superficie en un punto, que es el centro de proyección (ver fig.3.4.1).

Las más utilizadas son:

- *Proyección Central o Gnomónica:* Construida a partir de un punto de vista situado en el centro de la tierra. No conserva ni los ángulos ni las superficies pero todo círculo máximo de la esfera se proyecta según una recta.
- *Proyección Estereográfica:* El punto de vista es diametralmente opuesto al centro de proyección. En ella la escala se mantiene constante alrededor de un punto y a lo largo de cualquier círculo que tenga por centro al centro de proyección.
- *Proyección Ortográfica:* el punto de vista se encuentra en el infinito. Desgraciadamente se obtienen grandes deformaciones por lo cual solo es utilizada para cartografía del sol y de los planetas.

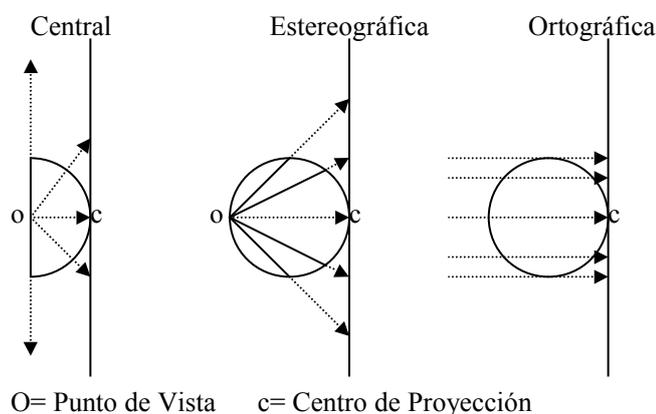


Figura 3.4.1. Proyecciones Acimutales

2. **Proyecciones Cilíndricas:** proyectar cada punto sobre una superficie cilíndrica que rodea la superficie terrestre .

Las más utilizadas son:

- *Proyección cilíndrica perspectiva o Central:* el cilindro es tangente al Ecuador con punto de vista en el centro de la esfera. Carece de interés práctico por su gran deformación pero se utiliza para la comprensión de características comunes de estas proyecciones.
- *Proyección de Mercator:* es una proyección conforme que se caracteriza por representar rectilíneamente las rutas trazadas mediante un compás, lo

cual la hace muy útil en navegación. Las escalas se mantienen constantes a lo largo de cada paralelo, pero la deformación aumenta rápidamente a medida que aumentan o disminuyen las latitudes (cambiando así de paralelo).

- *Proyección Mercator Transversa, de Gauss o cilíndrica de Lambert*: tiene por eje a un meridiano central, que se proyecta según una recta en el plano; el Ecuador es otra recta y está perpendicular al meridiano central, los demás paralelos y meridianos son curvas trascendentes y ortogonales entre sí.
- *Proyección Cilíndrica de Lambert*: los meridianos son rectas paralelas equidistantes y los paralelos son rectas perpendiculares cuya separación es decreciente hacia los bordes. Es por esto que solo es válida cerca del centro de proyección.

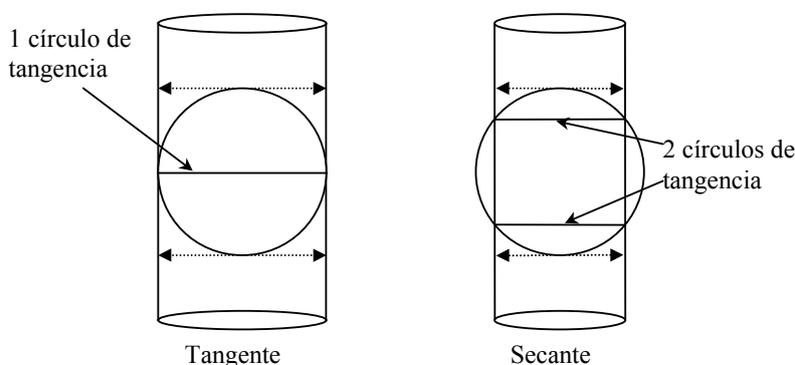


Figura 3.4.2 Proyecciones

3. **Proyecciones Cónicas**: proyectar cada punto sobre una superficie cónica que rodea la superficie terrestre. Esta superficie puede ser tangente (intersecta en un círculo) o secante (intersecta en 2 círculos) con respecto a la tierra. En ambos casos la escala se conserva sobre los círculos. Las deformaciones aumentan rápidamente a medida que se proyectan puntos lejanos de los círculos de tangencia.

Las más utilizadas son:

- *Proyección Cónica Central*: el punto de vista se encuentra en el centro de la tierra. Los meridianos se proyectan según rectas concurrentes y los paralelos son arcos de círculos concéntricos cuyo radio es función de la latitud.
- *Proyección Cónica Simple*: los meridianos se proyectan según rectas concurrentes y equidistantes. Los paralelos son arcos de círculo concéntricos también equidistantes. De esto resulta una proyección equidistante, aunque aumenta la escala en los círculos que no sean los de contacto.
- *Proyección Cónica Conforme de Lambert*: puede ser tangente o secante y conduce a meridianos rectilíneos concurrentes y paralelos circulares concéntricos.

- *Proyección Cónica Equivalente de Lambert*: los meridianos son rectas concurrentes y los paralelos son arcos concéntricos con radios que aseguran la condición de equivalencia.
- *Proyección de Bonne*: no es exactamente cónica, ya que los meridianos no son rectas concurrentes, pero se aproxima a ella ya que los paralelos son arcos de círculos concéntricos. El centro de proyección está en la intersección del meridiano central y de un paralelo L, tomado como origen. Las latitudes y longitudes se mantienen proporcionales a los de la esfera, y la proyección es equivalente. Las deformaciones de ángulos y distancias aumentan rápidamente al alejarse del meridiano central o del paralelo de origen.

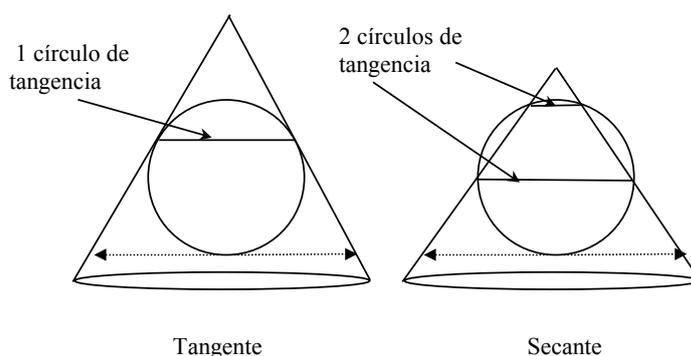


Figura 3.4.3 Proyecciones Cónicas

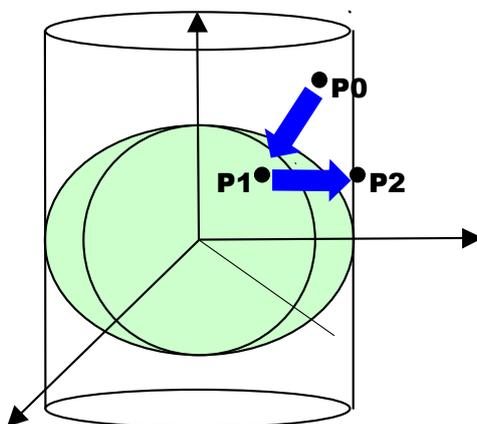
4. **Proyecciones de Mundo Entero**: Ninguna de las proyecciones precedentes permite una representación de la tierra en un solo mapa. O bien faltan los polos o existen latitudes y longitudes irrepresentables por sus excesivas deformaciones. Por ello se idearon otros sistemas que responden a esta necesidad:

- *Proyección Sinusoidal de Sanson-Flamsteed*: versión meridiana de la proyección de Bonne paralelos rectilíneos y equidistantes, divididos en partes iguales por los meridianos.
- *Proyección de Mollweide*: el ecuador y el meridiano central, que mide la mitad del anterior, son rectas perpendiculares entre si. Los meridianos son semielipses y los paralelos son rectilíneos y paralelos al Ecuador.

Sea cual sea la proyección de mundo entero seleccionada, siempre se tiene inconvenientes al alejarse del centro o de ciertas regiones privilegiadas. Existen técnicas específicas para reducir los inconvenientes al mínimo.

Todos los tipos de proyecciones mencionados convierten coordenadas expresadas en latitud y longitud en coordenadas planares x y, en consecuencia, lo que se proyecta en realidad no es la ubicación exacta del punto, pues se toma como origen la proyección de éste en el elipsoide de referencia sin considerar la altura.

La siguiente figura ilustra este concepto.



- P0. Punto sobre la superficie terrestre
- P1. Punto sobre el elipsoide
- P2. Punto proyectado

Figura 3.4.4 . Proyección de un punto

De esta manera las proyecciones no se basan en la altura elipsoidal (h), sólo consideran la posición del punto en el elipsoide de referencia. Teniendo en cuenta estas consideraciones arribamos al hecho que luego de realizar la proyección se pierde la altura del punto terrestre con respecto al elipsoide de referencia.

3.4.5 La proyección utilizada en Argentina: Gauss-Kruger [Brunini]

Nuestro país adoptó la proyección Gauss Kruger para la cartografía oficial. Se trata de una proyección conforme, cilíndrica y transversal.

Con excepción del meridiano central y del ecuador, cuyas representaciones sobre la carta son líneas rectas, todos los meridianos y paralelos quedan representados por líneas suavemente curvadas. Por la naturaleza conforme de la proyección, cualquier cruce entre un meridiano y un paralelo se produce en ángulo recto.

La proyección Gauss-Kruger conserva las formas pero no conserva las distancias ni las superficies. La distorsión en la distancia se cuantifica por medio del módulo de deformación, definido a través de la relación:

$$m = \frac{\text{distancia sobre la carta}}{\text{distancia sobre el elipsoide}}$$

El módulo de deformación crece rápidamente medida que el punto se aleja hacia el este o el oeste del meridiano central de la proyección. Su valor es siempre mayor que la unidad, salvo a lo largo del meridiano central, donde toma el valor 1. Lo dicho significa que la distancia sobre la carta es siempre mayor a lo sumo igual que sobre el elipsoide.

Para la cartografía oficial el país se dividió en siete fajas de tres grados de ancho, a cada una de las cuales les corresponde un meridiano central. Esta convención se adoptó basándose en el criterio de que en el borde de faja el módulo de deformación produjera en la cartografía un efecto imperceptible.

Es usual designar a las coordenadas Gauss Kruger con las letras X e Y. Estas coordenadas no deben ser confundidas con las cartesianas geocéntricas x,y,z . Las coordenadas Gauss Kruger X,Y se obtienen a partir de las geodésicas ϕ,λ utilizando fórmulas matemáticas en las que intervienen los parámetros del elipsoide a y b y la longitud del meridiano central λ_0 . Análogamente, las coordenadas geodésicas pueden obtenerse a partir de las Gauss Kruger mediante el uso de las fórmulas inversas.

3.5. SISTEMAS DE COORDENADAS

3.5.1. Introducción

Hemos visto que para referenciar la posición de un punto sobre la tierra existen diversas alternativas, si bien algunas son más precisas que otras, todas obtienen la ubicación en el espacio o en el plano para este punto.

La forma de expresar dicha ubicación se hace a través de algún sistema de coordenadas, por ejemplo, si elijo el marco de referencia ECEF, con un determinado elipsoide podré utilizar un sistema de coordenadas cartesianas (x,y,z) ó un sistema de coordenadas elipsóidicas (λ,ϕ,h) . Estos 2 tipos de coordenadas: cartesianas y geodésicas (o elipsóidicas) son **dos formas distintas pero equivalentes** para expresar la posición de un punto respecto a un datum. El usuario elige con cual prefiere trabajar.

Sistemas de coordenadas cartesianas

Son las coordenadas (x,y,z) de un punto según un marco de referencia que define un conjunto de ejes orientados y posicionados según se desee, pudiendo ser espacial ó planar, este último como consecuencia de algún método de proyección.

Las coordenadas cartesianas tienen la virtud de simplificar los cálculos matemáticos y por ello son utilizados por la mayoría de los programas de procesamiento de observaciones GPS. Su principal desventaja es la de no permitir una clara visualización espacial. Por ejemplo, es fácil calcular la distancia entre dos puntos, pero es difícil advertir qué punto está más al oeste o cuál está más alto.

Sistemas de coordenadas geodésicas o elipsóidicas

Un sistema de coordenadas geodésico es un sistema de coordenadas tridimensional definido por un elipsoide, el plano ecuatorial del elipsoide y un plano meridiano.

Las coordenadas geodésicas, por el contrario, facilitan la visualización espacial y, a pesar de que con ellas los cálculos matemáticos resulten más complicados, su uso está mucho más difundido que el de las cartesianas.

Para que estas coordenadas tengan validez, es necesario definir antes un marco de referencia y un elipsoide, lo que antes se definió como Datum, sin él dichas coordenadas carecen de sentido. Las coordenadas geodésicas se definen de la siguiente manera:¹

ϕ = Latitud = ángulo entre el plano X, Y y la normal al elipsoide que pasa por P

λ = Longitud = ángulo entre el plano X, Z y el plano meridiano de P

h = Altura = es la distancia entre P y el elipsoide

¹ Nota: No confundir la altura geodésica h , a veces llamada altura elipsóidica o altura sobre el elipsoide, con la altura sobre el geoido o altura sobre el nivel del mar, H , de la que se hablará luego.

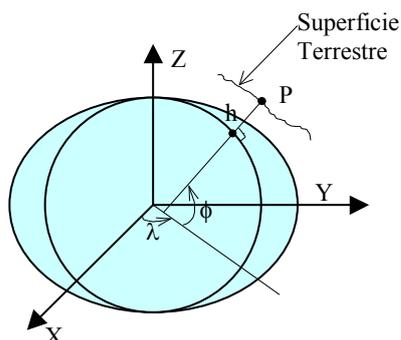


Figura 3.4.1 Sistemas de Coordenadas

Otros sistemas de coordenadas existentes son: el sistema de coordenadas esférico y el polar, entre otros.

3.5.2. Transformación entre sistemas de coordenadas

La conversión de las coordenadas cartesianas en geodésicas puede realizarse con total rigurosidad y precisión utilizando un conjunto de fórmulas matemáticas relativamente sencillas.

Es importante tener presente que las fórmulas de conversión utilizan los parámetros del elipsoide, *por lo que es imprescindible tener en claro qué elipsoide se está usando*. La tabla 3.5.1 ilustra los errores que se pueden cometer al convertir las coordenadas utilizando un elipsoide equivocado. Puede verse que confundir el elipsoide WGS84 con el Internacional 1980 no produce errores significativos (un décimo de milímetro), pero confundirlo con el Internacional 1924 produce errores mayores que 200 m.

	Internacional de 1924	WGS84	Internacional 1980
X[m]	1959438.62	1959352.38	1959352.38
Y[m]	-4849780.78	-4849567.31	-4849567.3188
Z[m]	-3637923.42	-3637866.90	-3637866.90

Tabla 3.5.1. Coordenadas cartesianas geocéntricas obtenidas al transformar las coordenadas geodésicas lat = 35 long = 68 y h = 0, utilizando diferentes elipsoides.

La fórmula para la transformación desde coordenadas geodésicas a cartesianas es:

$$X = (N + h) \cos \phi \cos \lambda$$

$$Y = (N + h) \cos \phi \operatorname{sen} \lambda$$

$$Z = (N(1 - e^2) + h) \operatorname{sen} \phi$$

donde N es el radio de curvatura de la sección normal que se puede calcular a partir de :

$$N = a / (1 - e^2 \operatorname{sen}^2 \phi)^{\frac{1}{2}}$$

Para poder realizar esta transformación es necesario conocer los parámetros a y e que definen al elipsoide en el cual se expresan las coordenadas. La transformación inversa es más compleja.

4. Modelo de Sistemas de Referencia

4.1. MODELO BASE

4.1.1. *Introducción al modelo base*

El modelo Orientado a Objetos para diseño de Aplicaciones GIS propone una arquitectura que une Objetos geográficos y no geográficos bajo una clase abstracta que agrupa el comportamiento geográfico común de todas sus subclases. Esta arquitectura desacopla las características de ubicación de los objetos para delegarlos a una clase que utiliza el Sistema de Referencia para el manejo de este aspecto. Esta tesis se basa en la ampliación de esta parte del modelo y su implementación.

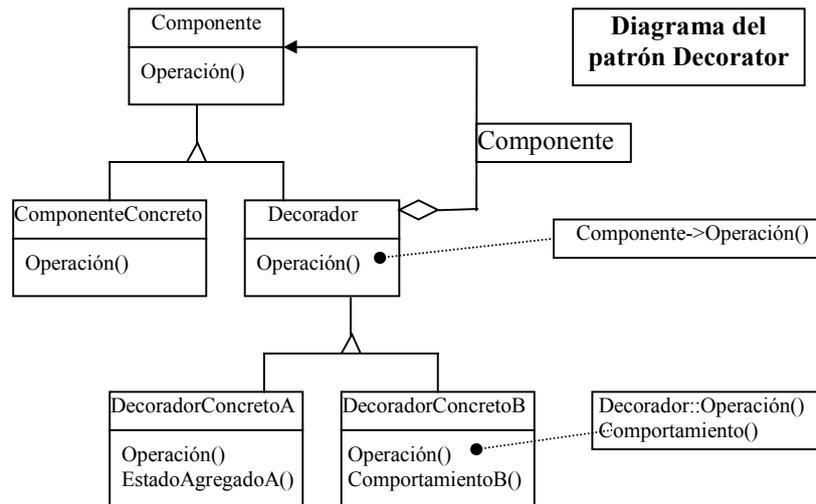
El modelo propone, básicamente, unir bajo una clase abstracta (AbstractGeoObject) los objetos que tienen características espaciales, utilizando el patrón de diseño Decorator para darle estas características a los objetos que no las tengan.

El patrón de diseño Decorator:

Decorator es un patrón de diseño definido en [6] cuya intención es “agregar dinámicamente responsabilidades adicionales a un objeto”, proporcionando una alternativa más flexible que la subclasificación para extender la funcionalidad de las clases. La diferencia sustancial entre decorar y el subclasificar es que mientras la primera es dinámica, la última es estática; usando decorators podemos agregar funcionalidad a algunos objetos sin re-definir la jerarquía conceptual.

Un decorator reproduce la interfaz del objeto decorado y también define conducta adicional. De esta manera, remite los mensajes al componente y puede realizar acciones adicionales antes o después de remitir [6].

Pueden decorarse objetos con conductas diferentes, por ejemplo la conducta espacial, y referirse al objeto original o a cualquiera de sus decoradores. De esta manera, dado que los decoradores imitan el protocolo del objeto conceptual, ellos delegan la implementación del protocolo no-espacial a ese objeto.



El uso de este patrón de diseño permite construir el modelo geográfico básico mediante el agregado de rasgos espaciales a cada objeto de una manera dinámica y transparente.

La siguiente figura ilustra la arquitectura general del patrón:

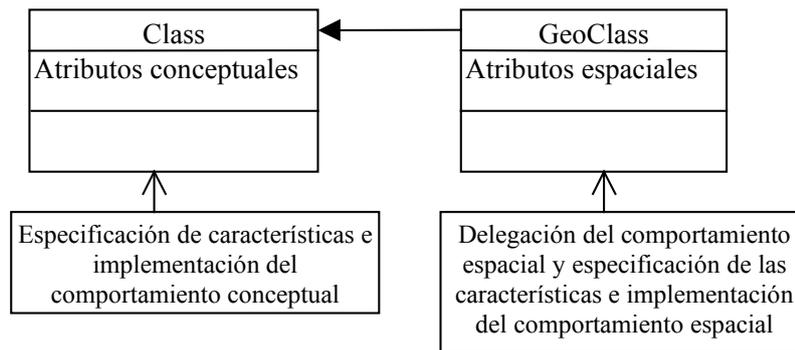


Figura 4.1.1. Especificación de una clase con un decorador

El modelo base se divide en dos capas: La Capa *Conceptual* y la Capa *Geográfica*.

La capa Conceptual maneja los aspectos convencionales del objeto, sin encargarse de sus aspectos geográficos, aún si los tuviere. Esto último es tarea de la Capa Geográfica del modelo.

Así, los objetos de la capa conceptual que contengan (además de las propias) características geográficas serán “envueltos” por objetos de la capa geográfica para que tengan ahora características geográficas como ubicación, área, distancia a otros objetos, etc. La resolución de los aspectos no geográficos de los objetos se realiza mediante la delegación de la solicitud al Objeto Conceptual.

Además de decoradores que “envuelven” objetos conceptuales, los objetos completamente geográficos pueden aparecer también; ellos no representan una vista de

cualquier objeto conceptual sino que se caracterizan por su relevancia Geográfica. Estos objetos pertenecen a las subclases de GeographicObject.

La arquitectura provee una clase que agrupa la conducta geográfica de los objetos que “decoran” a los objetos conceptuales y los objetos puramente geográficos.

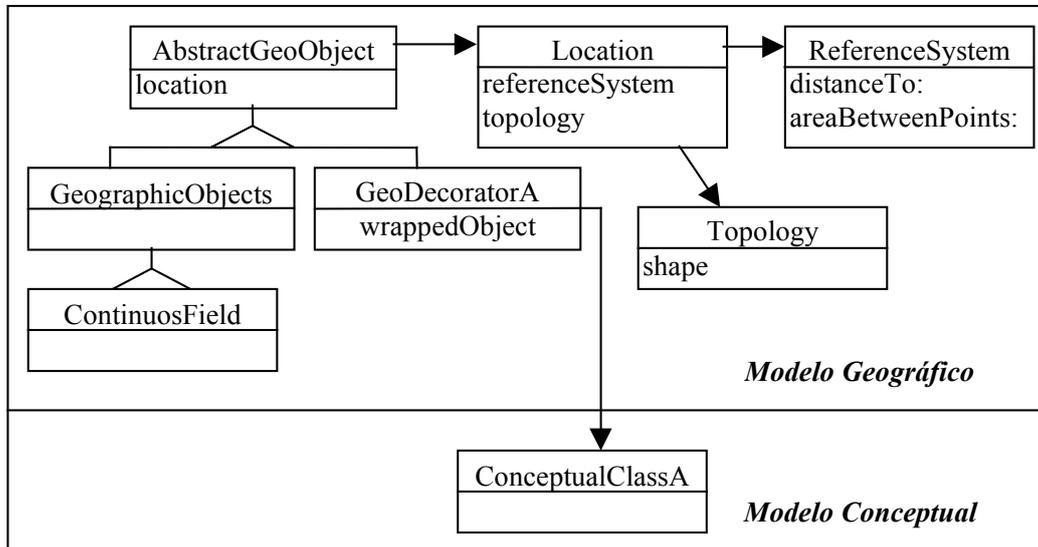


Figura 4.1.2. El esquema básico para construir aplicaciones geográficas

En el modelo base presentado ya se introduce el concepto de sistema de referencia, el cual se encuentra relacionado con una ubicación geográfica y es parte del modelo geográfico planteado.

En esta tesis nos concentraremos en las clases Location y ReferenceSystem.

Tomando el diseño del modelo base incorporaremos una pequeña adecuación subdividiendo la clase Location en un conjunto de objetos SingleLocation, y cada objeto de este tipo es quien posee el sistema de referencia.

Con esta modificación relacionaríamos el sistema de referencia con una simple ubicación geográfica y al juntar todas estas posiciones obtendríamos la ubicación de un objeto la cual está relacionada con una topología.

En secciones posteriores ampliaremos y explicaremos todo el diseño obtenido para los Sistemas de Referencia.

4.2. ANALISIS DEL MODELO

4.2.1 Dominio

El objetivo de esta tesis es encontrar una manera de poder referenciar objetos en un sistema de información geográfico de forma tal que el usuario se independice del sistema de referencia utilizado. El usuario puede mencionar una ubicación con el sistema que quiera y luego poder realizar operaciones entre los datos aunque éstos se hayan cargado con diferentes sistemas.

Para lograr este objetivo es necesario acotar el dominio de sistemas de referencia, pues en realidad pueden existir infinitos sistemas definidos a gusto del usuario co-

mo los sistemas de referencia *por nombre*, en los cuales se menciona una ubicación relacionándola con un grupo de palabras, como por ejemplo la definición de un domicilio; otro tipo de sistemas es el *lineal* el cual indica una ubicación relacionándola con una distancia a partir de un punto determinado.

Si bien, todos los sistemas mencionados y otros cualquiera inventados por el usuario son válidos, no van a ser considerados como parte del dominio de esta tesis, simplemente porque con ellos no se puede lograr alcanzar los objetivos planteados, no hay manera de poder hacer que el sistema automáticamente pueda transformar una ubicación definida en alguno de ellos a otro sistema de referencia cualquiera, la única manera para que esto suceda es que el usuario determine una manera de poder pasar a otro sistema de referencia y así sea posible el traspaso de uno a otro sin inconvenientes, pero esto no es lógico que suceda, por consiguiente se van a considerar como válidos para esta tesis aquellos sistemas de referencia en 2 y 3 dimensiones los cuales son utilizados comúnmente en los SIG.

Con este dominio de sistemas de referencia, es posible (mediante cálculos matemáticos y relaciones entre ellos) poder integrarlos y trabajar con todos a la vez haciendo más fácil la tarea del usuario sobre todo de los menos experimentados en SIG, ya que se obtendrá independencia en la manera de almacenar los datos y la transformación interna será transparente para él.

4.2.2. Análisis de los elementos del modelo

Primeramente se han de distinguir los elementos básicos que todo sistema de referencia necesita, éstos son: marco de referencia, figura de referencia y sistema de coordenada empleado. Cada uno de estos elementos debe estar presente en todo sistema de referencia y deben ser combinados para poder determinar las características del mismo.

La *figura de referencia* puede ser una esfera, un elipsoide de revolución, etc. Dicha figura nos provee una aproximación matemática de la esfera terrestre y cada sistema de referencia utilizará la figura que mejor se adecue a la zona que se desea direccionar.

El *marco de referencia* nos ubica espacialmente, es decir, nos provee una orientación y desplazamiento en el espacio tridimensional. Al igual que la figura de referencia, depende de la zona a referenciar.

Una combinación adecuada del marco de referencia y figura de referencia es fundamental para lograr una buena aproximación de la tierra (teniendo en cuenta la zona con la que se desee trabajar) y de esta manera lograr minimizar el error de referencia obtenido.

En general a la figura de referencia junto con el marco de referencia se lo llama *Datum*, existiendo diferentes datums según la región elegida. Según distintos autores la figura de referencia puede o no considerarse como parte del Datum, en esta tesis se considerará a la figura de referencia como un dato característico de ciertos sistemas de coordenadas.

El *sistema de coordenadas* puede ser rectangular tridimensional, polar, geodésico, planar, etc. Algunos sistemas de coordenadas no dependen de la figura de referencia, como por ejemplo el sistema rectangular, en cambio otros necesitan de una figura para poder expresar las coordenadas, como el sistema geodésico. Por esto se ha considerado la necesidad de tener a la figura de referencia como un dato propio de algunos sistemas

de coordenadas y no de todo sistema de referencia (dicho concepto es ampliado en análisis posteriores).

Por último, un sistema de referencia también puede involucrar una proyección cartográfica, de esta forma en algunos casos será necesario un *sistema de proyección* adecuado. En realidad la necesidad de una proyección aparece cuando se requiere pasar a 2 dimensiones, por ejemplo para la confección de mapas.

Es necesario aclarar que, si bien al utilizar una representación planarizada podría alcanzarse a trabajar con 2 coordenadas, puede necesitarse el dato de la altura por diversos motivos, entre ellos la realización de mapas en relieve, es por esto que en el caso de las ubicaciones expresadas en sistemas de 2 dimensiones se incorporará la altura elipsoidal con respecto a alguna figura de referencia para las coordenadas planas.

4.2.3 Análisis del comportamiento y colaboraciones entre los elementos

Todos los objetos mencionados, los cuales son parte de un sistema de referencia interactúan entre sí y colaboran mutuamente para poder determinar la posición de un objeto en un SIG.

Como se expuso en el modelo base, cada objeto en un SIG posee una ubicación geográfica la cual se encuentra expresada mediante un sistema de referencia cualquiera, cada objeto puede informar, además de su posición, la distancia a otros objetos o el volumen que posee. Para obtener esta información el objeto geográfico debe interactuar con su posición para que ésta le retorne la respuesta a la operación solicitada. De la misma manera la posición (location) se relaciona con su sistema de referencia para que éste realice el cálculo correspondiente.

Así, se establece una serie de colaboraciones entre los objetos mencionados para obtener el resultado deseado.

Por ejemplo, supongamos que deseamos obtener la distancia entre una ubicación geográfica llamémosla *a* y otra ubicación *b* las cuales poseen 2 sistemas de referencia distintos *A* y *B* respectivamente. La ubicación *a* va a requerir a su sistema *A* que calcule la distancia entre las 2 ubicaciones. El sistema *A* primeramente va a pedirle a la ubicación *b* que transforme sus coordenadas para que éstas sean expresadas en el sistema *A* y así obtener coordenadas compatibles. Una vez que las 2 ubicaciones estén expresadas en el mismo sistema de referencia se procede a realizar el cálculo de distancia (dicha operación es realizada por el sistema de referencia *A*).

Para realizar toda operación que involucre 2 o más ubicaciones se procederá de manera similar a lo expuesto en el párrafo anterior. Para poder lograr eso, toda ubicación debe poder cambiar su sistema de referencia sin importar el que tenga actualmente.

Para cambiar de sistema de referencia la ubicación geográfica va a tener que interactuar con su propio sistema para que éste realice el cambio de coordenadas a otro sistema de referencia apropiado. Si bien, vemos que el sistema de referencia colabora con la ubicación para realizar la transformación, el propio sistema de referencia, el cual debe realizar los cálculos, también necesita colaboración de los objetos que lo conforman para obtener la respuesta.

De esta forma el sistema de referencia requiere colaboración del sistema de coordenadas origen y destino como así también del datum origen y destino para poder realizar la conversión.

Para que quede más claro esta colaboración, se expondrá un ejemplo el cual muestra como una ubicación cambia su sistema de referencia. Supongamos que la ubicación posee el sistema de referencia A, el cual está conformado por el sistema de coordenadas Mercator y el datum NAD97 y va a cambiar al sistema de referencia B con el sistema de coordenadas ECEF y el datum Inchauspe.

En primer lugar, se cambia el sistema de coordenadas desde Mercator a ECEF, dicha transformación la realiza el sistema de coordenadas origen, es decir, el sistema de coordenadas Mercator. Luego se cambia el Datum de la misma manera desde NAD97 a Inchauspe.

Observando el ejemplo, se deduce que el sistema de coordenadas Mercator debe conocer las fórmulas y el procedimiento de pasaje hacia ECEF y el Datum NAD97 debe conocer como pasar a Inchauspe.

Generalizando estos conceptos todo sistema de referencia debe tener conocimiento de cómo pasar a cualquier otro sistema y de la misma forma, todo sistema de coordenadas debe conocer como pasar a cualquier otro sistema de coordenadas y todo Datum debe también conocer la transformación a cualquier otro Datum.

Para poder obtener un resultado automatizado de todo este comportamiento, sería necesario que todo sistema de referencia, sistema de coordenada y Datum, conozcan los cálculos y métodos para poder realizar la transformación a cualquier otro que exista. Esto es prácticamente imposible de desarrollar, y sería muy complicado incorporar un sistema nuevo, puesto que habría que indicar el método para pasar de éste a todos los existentes y a todos los existentes incorporarles además el conocimiento de cómo pasar a este nuevo.

Para solucionar esto se va a introducir el concepto de *Sistema de Referencia Base*.

Un *sistema de referencia base* es un sistema de referencia el cual conoce como pasar a cualquier sistema de referencia existente en la aplicación y cada sistema de referencia conoce como transformar las coordenadas para pasar a éste. De esta manera al incorporar un sistema nuevo solo serán necesario incorporar las transformaciones de éste hacia el sistema base y desde el sistema base a este nuevo.

El sistema de referencia base es un concepto que se introduce para poder realizar de una manera más fácil las transformaciones internas entre los sistemas. Como es de esperar este sistema estará conformado por un sistema de coordenadas base y un datum base.

4.3. ARQUITECTURA DEL MODELO DE SISTEMAS DE REFERENCIA

4.3.1. Descripción y presentación del modelo

Como se expuso en la explicación del modelo base, se debe mencionar primeramente la clase Location.

Esta clase indica la posición geográfica de un GeoObjeto del SIG, es decir, es quien almacena valores expresados en un sistema de referencia para determinar donde se encuentra dicho GeoObjeto en la esfera terrestre.

Dicha clase Location está compuesta por un conjunto de coordenadas, las que son interpretadas por un sistema de referencia.

El modelo base plantea que para cada objeto Location exista un único sistema de referencia, es decir, cada coordenada que forma parte de la Location se encuentra expresada en el mismo sistema. En esta tesis se incluirá el sistema de referencia como dato propio de cada coordenada pudiendo existir dentro de una misma Location 2 coordenadas expresadas en 2 sistemas de referencia distintos.

El motivo de modelarlo de esa forma es para darle flexibilidad al modelo y quitarle restricciones al usuario al momento de definir la ubicación de un objeto. Si bien es cierto que en la mayoría de los casos se empleará el mismo sistema de referencia para dar la posición de un objeto, puede no ser así, por ejemplo, supongamos que deseamos obtener la ubicación de América del Sur, es probable que haya que integrar información proveniente de diferentes países y por consiguiente con diferentes sistemas, por lo cual es más cómodo poder tener la posibilidad de que se detalle una posición con un conjunto de coordenadas expresadas cada una en un sistema de referencia diferente.

El único inconveniente es que para ciertas operaciones debe especificarse en que sistema de referencia realizar los cálculos, a diferencia que en el planteado en el modelo base pues como contaba con un único sistema de referencia se realizan los cálculos en base a éste.

Una vez que se ha definido la clase Location nos queda concentrarnos en el sistema de referencia. El sistema de referencia representará el marco de referencia que el usuario utiliza para referirse a su ubicación geográfica, es decir, determina la orientación y desplazamiento en 3 dimensiones y fija el marco a partir del cual se van a obtener las coordenadas correspondientes.

Ahora bien, si supusiéramos un marco de referencia fijo, podríamos hacer cambiar el sistema de coordenadas manteniéndonos en el mismo marco de referencia, es decir, sin rotar ni desplazar el origen.

De esto se deduce que el sistema de coordenadas es independiente del marco de referencia utilizado, por ejemplo para el cambio de un sistema de coordenadas a otro solo se necesitan los datos de la figura de referencia y el mecanismo propio de cada sistema de coordenadas.

Si en cambio, cambiáramos el marco de referencia se rotaría y desplazaría el centro lo cual produciría un cambio en el referenciamiento actual.

Hasta este punto hemos distinguido dos cosas: primero hay que contar con sistemas de coordenadas que tengan la capacidad de pasar de uno a otro y de un mecanismo que permita intercambiar los marcos de referencia.

Para el cálculo de distancias, áreas, etc., debemos llevar todos los puntos involucrados a un mismo marco de referencia y a un mismo sistema de coordenadas y a partir de ese momento hacer el cálculo. Convengamos que el cálculo de distancia podría ser distancia sobre un elipsoide, sobre el geoide o en el espacio de dos o tres dimensiones. Lo mismo ocurre en el caso de el cálculo de áreas.

Por consiguiente por un lado está el sistema de referencia y por otro el sistema de coordenadas.

Los sistemas de coordenadas empleados en un sistema de referencia geodésico no solo son los estándares de la matemática, también aparecen otros tipos y aquí es donde entran en juego la figura de referencia y las proyecciones cartográficas, teniendo sistemas de coordenadas geodésicos, geográficos y proyectados.

Estos últimos sistemas de coordenadas (proyectados) dependen no solo de una proyección específica sino también de la figura de referencia, puesto que lo que se proyecta es la figura de referencia (en general elipsoide) porque sino sería matemáticamente imposible realizar los cálculos.

4.3.2. Descripción de Clases de Modelo

1. Location

Un objeto de la clase Location expresa la posición de un Geo-Objeto, tal posición está conformada por una serie de puntos simples más una topología.

La topología indica la forma en que los puntos se relacionan para describir la ubicación, es decir, la topología puede ser un polígono, un círculo, u otra forma más compleja, de esta manera los puntos se unen de acuerdo a la topología para indicar la forma de dicho objeto Location.

Cada punto simple perteneciente a un objeto Location va a ser un objeto de una clase nueva la cual llamaremos SingleLocation (figura 4.3.1).

A continuación se explicará la clase SingleLocation, la clase Topology no será enunciada por no ser de relevancia para el cumplimiento del objetivo de la tesis.

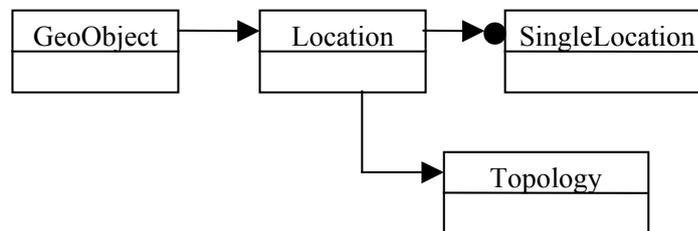


Figura 4.3.1. Location

2. SingleLocation

Un objeto de la clase SingleLocation es un punto de la tierra cuya ubicación se expresa con una coordenada y con un sistema de referencia.

En relación con la clase Location puede destacarse que para un mismo objeto Location podría existir más de un sistema de referencia (como fue mencionado en el análisis previo) pues cada objeto SingleLocation que lo forma es independiente uno de otro y cada uno posee su propio sistema.

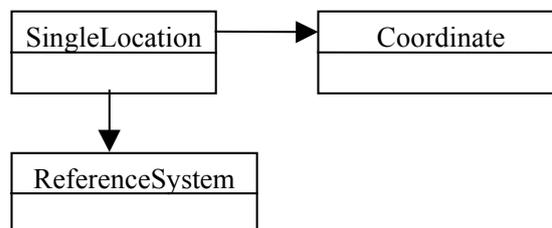


Figura 4.3.2. SingleLocation

Un objeto SingleLocation puede cambiar su sistema de referencia en cualquier momento, dicho cambio provoca que el objeto recalculé su coordenada dependiendo del sistema de referencia que tenía originalmente y el nuevo al que se desea cambiar.

3. ReferenceSystem

Determina el marco de referencia tridimensional que será la base a partir de la cual se describen y calculan las coordenadas.

Se configura teniendo en cuenta un desplazamiento compuesto de 3 valores y una orientación compuesta por 3 ángulos. Dicho desplazamiento y orientación deben calcularse de acuerdo a un sistema de referencia de partida o base.

De esta manera se debe determinar de antemano el sistema de referencia base para el cual sus valores en desplazamiento y orientación son 0 y a partir de éste los demás sistemas de referencia calculan sus valores.

La presencia de dicho sistema es fundamental pues esto determina el punto de partida o punto común para poder comparar la posición de un sistema con respecto a otro.

Todo sistema de referencia tiene la capacidad de realizar operaciones entre objetos del tipo `SingleLocation`, como así también de cambiar a otro sistema realizando los cálculos correspondientes.

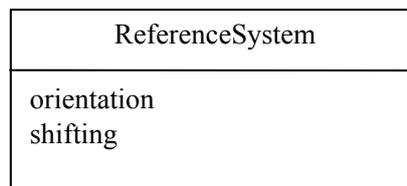


Figura 4.3.3. ReferenceSystem

4. Coordinate

Un objeto de la clase `Coordinate` representa una coordenada expresada en un sistema de coordenadas. Una coordenada puede ser de 2 o 3 dimensiones según el sistema asociado.

Podríamos contar con objetos del tipo `Coordinate` en aplicaciones que realicen gráficos, en las cuales las coordenadas nos ubican en la pantalla para dibujar objetos o en cualquier otra aplicación que las necesite, es decir, esta clase es independiente del sistema de referencia.

Dentro del contexto de sistemas de referencia, una coordenada determina la posición según un marco de referencia impuesto por el objeto `SingleLocation` que la posee, este marco es útil para el objeto `SingleLocation` pero para la coordenada no lo es porque es independiente de dicho marco. Una coordenada trabaja suponiendo un marco fijo cualquiera sin importarle la posición de éste en la esfera terrestre, esto último es importante solo para el objeto `SingleLocation`.

El usuario puede cambiar el sistema de coordenadas en cualquier momento y la coordenada recalcula sus valores dependiendo del sistema origen y destino.

Cuando el usuario pide el valor a una coordenada, ésta retorna valores expresados en el sistema de coordenadas actual, los cuales son un conjunto de magnitudes (angulares o de longitud), por ejemplo si el sistema de coordenadas es el Geodésico, la co-

ordenada a retornar va a estar formada por 2 unidades angulares y una unidad de longitud, si en cambio es el sistema de coordenadas rectangular en 3 dimensiones va a retornar 3 unidades de longitud, etc.

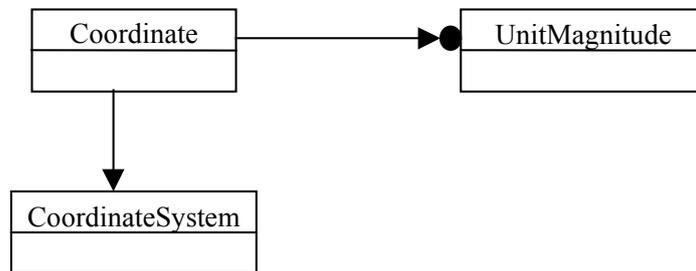


Figura 4.3.4 Coordinate

5.CoordinateSystem

El sistema de coordenadas provee una manera de referenciar un punto dentro de un sistema de ejes rectangulares.

En matemáticas, los sistemas de coordenadas más conocidos son: el polar y el planar, siendo éstos los sistemas que mencionaremos a modo de ejemplo en esta tesis, aunque todos los conceptos sirven para los demás sistemas.

Además de contar con estos sistemas matemáticos existen otros tipos que son los relacionados con la geodesia, los cuales involucran un elipsoide para realizar la referencia.

Debido a esto en el modelo se contará con 2 subclases debajo de la clase CoordinateSystem las cuales son las clases GeographicCoordinateSystem y MathematicalCoordinateSystem.

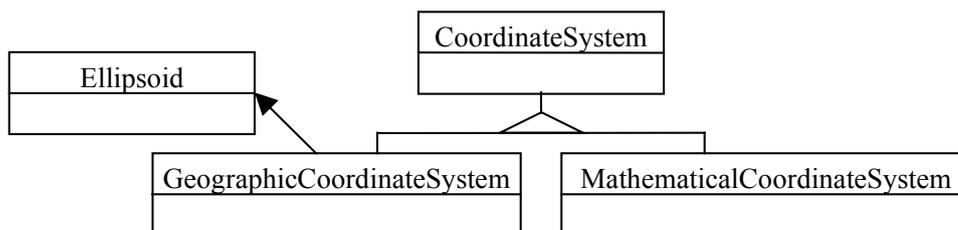


Figura 4.3.5. CoordinateSystem

5.1.GeographicCoordinateSystem

El sistema de coordenada geográfico depende de una figura de referencia (elipsoide) y determina la posición en el espacio de 3 dimensiones.

La coordenada se interpreta mediante 2 valores, que son Longitud y Latitud cuya definición es la siguiente:

Latitud(ϕ): el ángulo que forma la recta normal al elipsoide que pasa por el punto P y el plano ecuatorial (XY) del sistema de coordenadas rectangular.

Longitud(λ): el ángulo diedro entre el plano ecuatorial y el plano meridiano de el punto P (plano que contiene al eje z y al punto P).

Además de este sistema relacionado con la geodesia también existe otro semejante que incorpora un tercer valor a la coordenada, este último valor indica la altura, con la siguiente definición:

Altura (h): Distancia desde el punto P al elipsoide, medida sobre la normal a este.

A este sistema de coordenadas se lo llama Geodésico (GeodeticCoordinateSystem) y se incluirá en el modelo como una especialización del anterior.

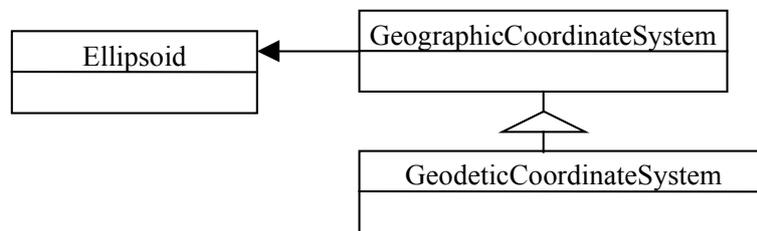


Figura 4.3.6. GeographicCoordinateSystem

5.2. MathematicalCoordinateSystem

Los sistemas de coordenadas matemáticos son los utilizados más comúnmente existiendo el sistema de coordenadas polar, el rectangular, etc.

Dichos sistemas matemáticos tienen además una dimensión asociada pudiendo ser de 2 o 3 dimensiones.

En el caso que el sistema sea de 2 dimensiones puede llegar a tener una proyección asociada, la cual será la función que se utilizó para obtener las coordenadas planares partiendo de un espacio tridimensional.

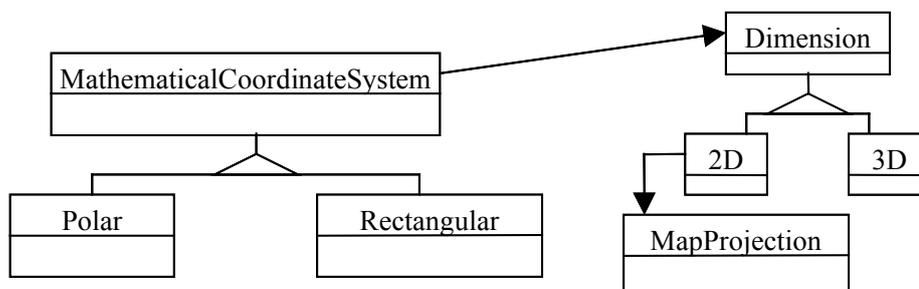


Figura 4.3.7. MathematicalCoordinateSystem

Cada sistema de coordenadas implementa la transformación de coordenadas expresadas en su propio sistema a otras en otro sistema diferente.

6.Map Projection

La superficie terrestre es no desarrollable, ya que si se cortara y extendiera en un plano, ésta se vería deformada y con errores de distancias entre puntos y ángulos.

Por esta razón para pasar de una representación en 3 dimensiones a una representación planarizada se necesita una correspondencia entre los puntos de la superficie y los del plano de proyección, a esta correspondencia se la denomina Sistema de Proyección.

De esta manera todo sistema de referencia de 2 dimensiones posee un sistema de proyección asociado.

Cada sistema de proyección (MapProjection) tendrá funciones para obtener las coordenadas (x,y) en el plano, a partir de sus coordenadas geográficas (latitud/longitud) y de las dimensiones del elipsoide de referencia (a y e). También se definen las funciones inversas.

Si por ejemplo se desea proyectar una ubicación expresada en algún sistema de referencia de 3 dimensiones, la proyección es la que realiza las conversiones adecuadas.

Es muy importante recordar que todo sistema de proyección convierte coordenadas expresadas en latitud y longitud, es decir, lo que se proyecta en realidad no es la ubicación exacta del punto sino que se toma como origen la proyección de éste en el elipsoide de referencia sin considerar la altura (ver figura 3.4.4 del capítulo 3).

Así, las proyecciones no utilizan la altura elipsoidal (h), sólo se basan en la posición del punto en el elipsoide de referencia.

Como consecuencia de lo mencionado anteriormente, al proyectar se pierde el valor que corresponde a la altura (h). Este dato es de suma importancia, no solo para poder realizar las transformaciones inversas y retornar al espacio tridimensional, sino también para la confección de mapas en relieve o para el cálculo de curvas de nivel; por esto una coordenada expresada en 2 dimensiones y que posea una proyección, va a guardar esta información solucionando de esta manera el problema presentado.

Otro punto a considerar en el tema de proyecciones es el sistema geodésico de origen. Tal como se expuso en capítulos previos, una proyección, tal como su nombre lo indica, refleja (o proyecta) cada punto del elipsoide en un plano, un cilindro o un cono, y luego corta y extiende estas figuras mostrando el resultado; este elipsoide de origen podría ser cualquiera, intentando ajustar lo mejor posible la zona de la tierra que se desea proyectar.

Como es de observar el elipsoide de origen es un dato fundamental para realizar la proyección, obteniendo diferentes resultados si aproximamos la región con 2 elipsoides diferentes, es decir, si aplico por ejemplo la proyección Mercator partiendo de un determinado elipsoide y un marco de referencia obtenemos una determinada proyección, pero si en cambio se parte de otro elipsoide y/o marco el resultado cambia. Esto implica que no se obtienen los mismos resultados al proyectar un mismo punto partiendo de 2 sistemas de referencia geodésicos distintos.

Por esto la clase MapProjection tiene asociado el elipsoide de referencia de origen, al contar con esa información es posible realizar las transformaciones inversas la cual sería imposible de otra manera.

6.1. Diagrama de Clases de los Sistemas de Proyección

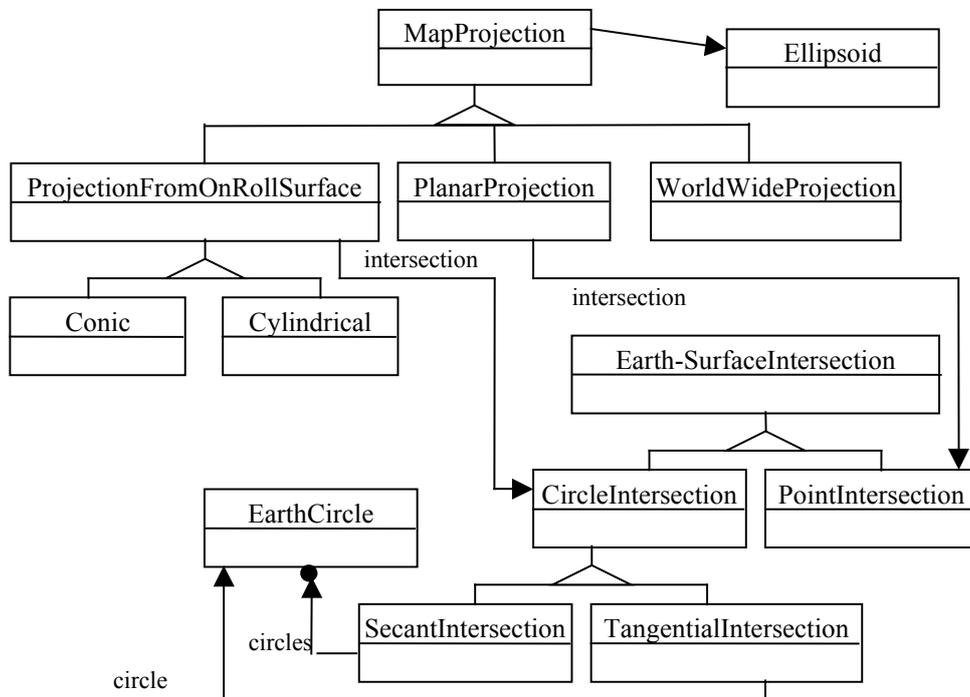


Figura 4.3.8. Modelo de sistemas de proyección

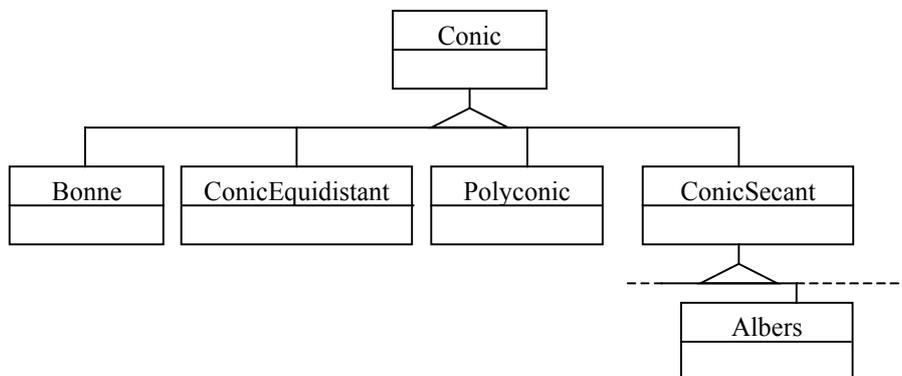


Figura 4.3.9. Jerarquía de las proyecciones cónicas

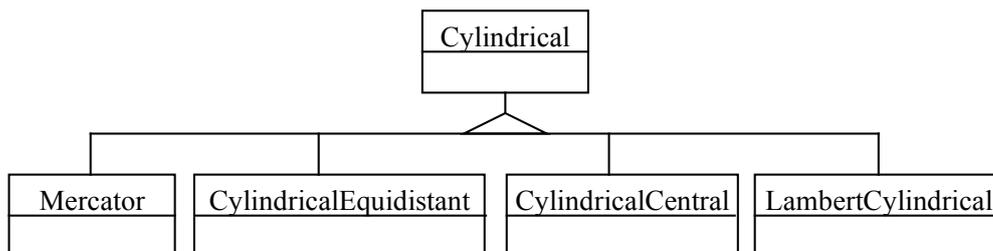


Figura 4.3.10. Jerarquía de las proyecciones cilíndricas

6.2. Participantes

- *MapProjection*:

Clase abstracta que define el comportamiento de todas las proyecciones. Cada subclase toma una coordenada y retorna una nueva la cual es el resultado de la proyección

- *ProjectionFromOnRollSurface*:

Clase abstracta que define la estructura básica de una proyección sobre una superficie desarrollable: cilindro o cono. Las mismas poseen información que indica el centro de proyección y el tipo de intersección con el elipsoide de referencia.

- *PlanarProjection*:

Define la estructura básica de todas las proyecciones planares, como la utilización de una instancia de *PointIntersection* representando el punto de tangencia entre el plano de proyección y la esfera.

- *WorldWide Projection*:

Define la estructura básica de todas las proyecciones de mundo entero. Estas proyecciones son especiales ya que están diseñadas para minimizar los errores que producirían las demás proyecciones si se quisiera proyectar el mundo entero.

- *Conic*:

Define la estructura básica y comportamiento de todas las proyecciones cónicas.

- *Cylindrical*:

Define la estructura básica y comportamiento de todas las proyecciones cilíndricas.

- *Earth-SurfaceIntersection*:

Clase Abstracta que representa la intersección entre la superficie de proyección y la esfera terrestre.

- *PointIntersection*:

Representa la intersección del plano de tangencia y la esfera a través de un único punto.

- *CircleIntersection*:

Clase abstracta que representa la intersección entre el cono o cilindro y la esfera terrestre a través de círculos, por ej. meridianos, paralelos y círculos oblicuos.

- *TangentialIntersection*:

Representa la intersección del plano de tangencia y la esfera a través de un solo círculo, por ej. un meridiano, un paralelo o un círculo oblicuo.

- *SecantIntersection*:

Representa la intersección del plano de tangencia y la esfera a través de dos círculos.

- *EarthCircle*:

Clase Abstracta que representa todos los círculos representables sobre la esfera terrestre (meridianos, paralelos y círculos oblicuos).

7.UnitMagnitude

La clase UnitMagnitude contiene un valor que expresa una magnitud, la cual podrá ser una distancia, un eje de alguna coordenada rectangular, el valor de un ángulo, etc. La clase almacena un valor junto con una unidad de magnitud (cm., km., m., grados, radianes, etc.), a un objeto de esta clase se le puede pedir que cambie de unidad y éste automáticamente recalcula el valor equivalente para la nueva unidad y lo almacena.

Entre objetos del tipo UnitMagnitude se pueden realizar operaciones tales como

suma, resta, etc., dichas operaciones se realizan unificando las unidades de las magnitudes involucradas por ejemplo, si tenemos 3 cm. + 1 m. primeramente se realizan las transformaciones de unidades correspondientes y luego se procede a calcular la suma.

Las magnitudes angulares tiene características particulares que no poseen las demás magnitudes, como ser cálculo de funciones trigonométricas, por esto se crea una especialización de la clase UnitMagnitude la cual llamaremos AngularUnitMagnitude (figura 4.3.11).

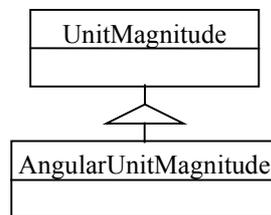


Figura 4.3.11. UnitMagnitude

Para poder realizar conversiones apropiadas se deben agrupar aquellas unidades pertenecientes a la misma familia, es decir, todas las unidades de longitud se unen en una misma familia como metros, centímetros, etc., y es posible poder cambiar entre éstas unidades, no siendo posible, por ejemplo, cambiar de metros a grados ya que éstas pertenecen a 2 familias de unidades diferentes.

Para el modelo de la clase UnitMagnitude se empleó el patrón “Quantity” de M. Fowler el cual será explicado en la sección final del capítulo.

4.3.3. Diagrama de Clases de Sistemas de Referencia

Integrando todas las clases involucradas, se obtiene el modelo final para los sistemas de referencia.

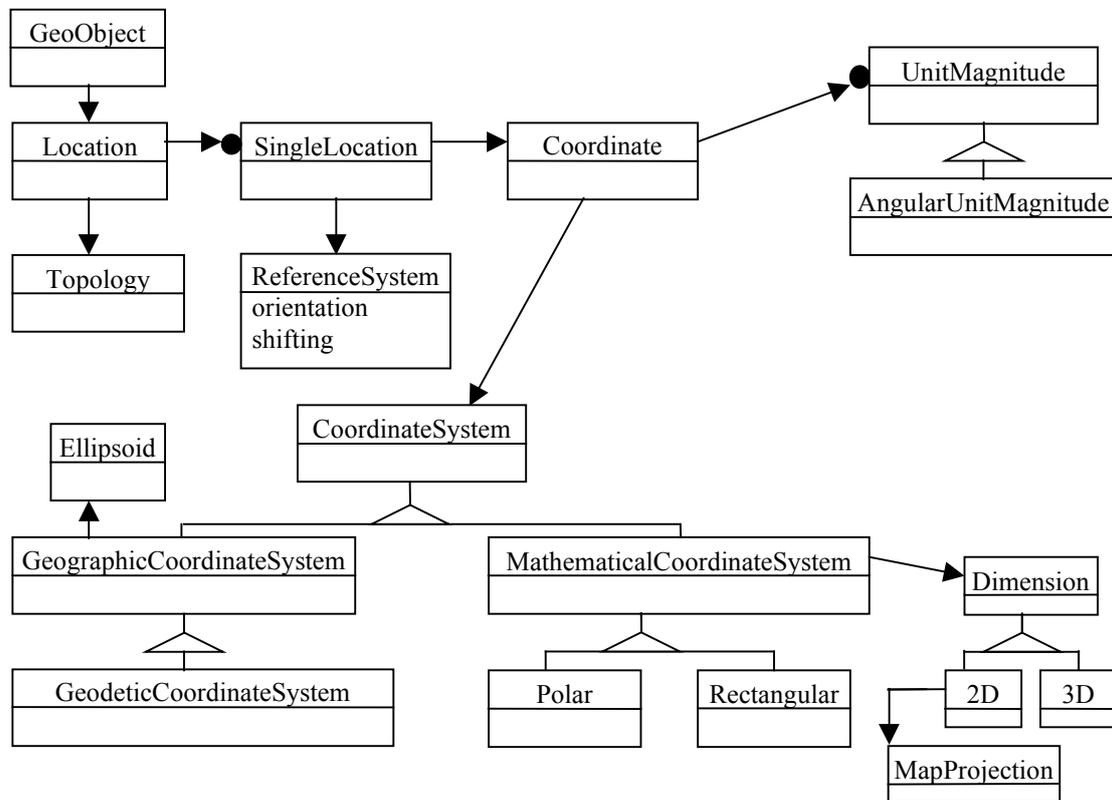


Figura 4.3.12. Modelo de Sistemas de Referencia

4.3.4. Comportamiento y colaboraciones entre las clases

4.3.4.1. TRANSFORMACIÓN ENTRE SISTEMAS DE REFERENCIA

El principal comportamiento del modelo de sistemas de referencia es el de permitir la transformación de cualquier ubicación expresada en un sistema de referencia a cualquier otro sistema de referencia.

Tal método es la base del modelo y es empleado en todas las demás operaciones que realiza pues siempre que sea necesario aplicar una operación que involucre más de una SingleLocation se procederá a pasar todas a un sistema de referencia común y luego realizar las operaciones, de esta manera es posible poder interrelacionar y operar con datos provenientes de cualquier sistema.

Por consiguiente la clase SingleLocation contendrá un método llamado *expressIn: aReferenceSystem* que recibe como parámetro el sistema de referencia que se quiere seleccionar como actual para ese objeto (figura 4.3.13 . Referencia 1)

Frente a esta invocación un objeto del tipo SingleLocation delega la responsabilidad al nuevo sistema de referencia, pidiéndole que cree un nuevo objeto SingleLocation para ese sistema partiendo de él mismo.

El método será *createSingleLocationFrom: aOriginalSingleLocation*, la respuesta a tal método será un objeto de la clase SingleLocation resultante de cambiar el sistema de referencia al objeto SingleLocation inicial (figura 4.3.13. Referencia 2).

Finalmente el objeto SingleLocation original toma los valores del nuevo. (figura 4.3.13. Referencia 3).

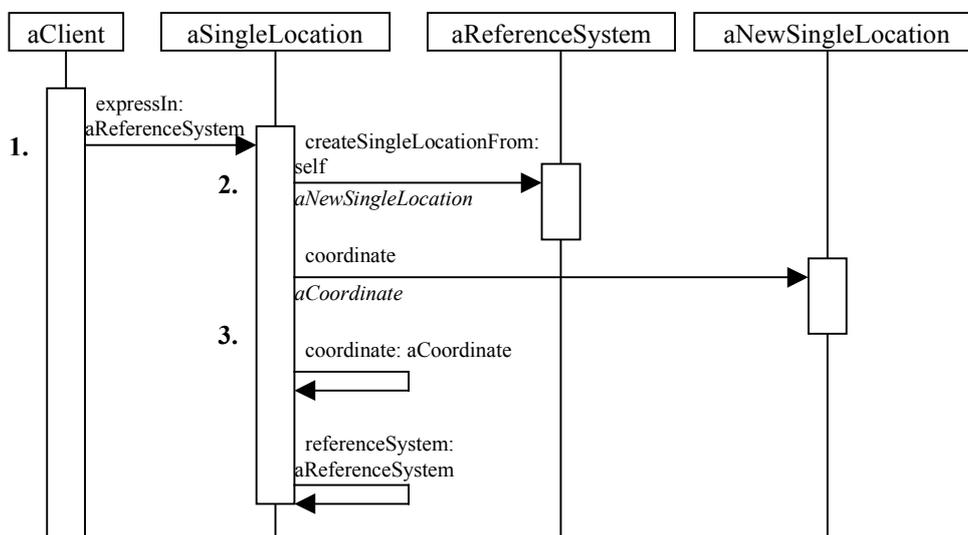


Figura 4.3.13 . Diagrama de interacción mostrando el cambio de sistema de referencia en un objeto de la clase SingleLocation

El método *createSingleLocationFrom: aOriginalSingleLocation*, de la clase ReferenceSystem cambia el marco de referencia, es decir, produce un desplazamiento y una rotación en la coordenada del objeto SingleLocation adecuándose al nuevo marco o sistema de referencia.

Todo sistema de referencia posee 6 parámetros que informan el desplazamiento y la orientación que tiene con respecto al sistema de referencia base. En análisis previos ya hemos mencionado la presencia del sistema base y porqué es necesario contar con él.

Para la implementación de esta tesis se tomó como sistema de referencia base “**el sistema WGS84**”, pues es por convención el que más se aproxima al marco de referencia terrestre verdadero y es el utilizado para el posicionamiento satelitario GPS mediante el cual se desea unificar las mediciones del mundo. El sistema de referencia base puede ser cualquiera pero hay que tener en cuenta que al momento de crear los demás sistemas se debe conocer el desplazamiento y orientación de éstos con respecto al base, por ejemplo, al crear el sistema Inchauspe se debe indicar cuan desplazado se encuentra su centro en relación con el sistema WGS84 (base) y además la rotación de sus ejes con relación a los ejes del sistema base.

Por esto, cada sistema de referencia obligatoriamente implementa 2 métodos; uno de ellos tomará una coordenada expresada en el sistema de referencia base y retornará una nueva coordenada en el sistema de referencia actual y el segundo método realizará la relación inversa.

Con estos métodos alcanza para poder pasar de cualquier sistema de referencia a otro.

Una vez explicado el cambio entre marcos de referencia, procederemos a explicar el cambio de sistemas de coordenadas.

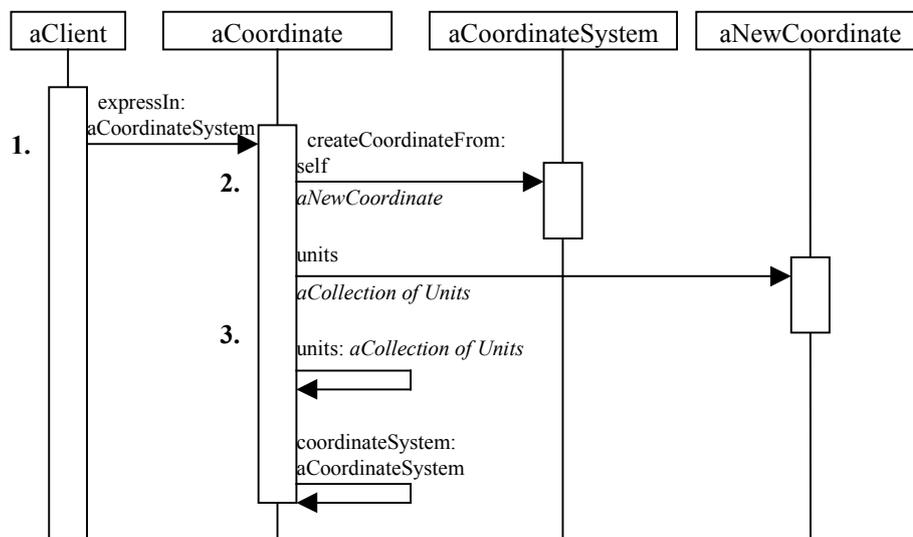


Figura 4.3.14. Diagrama de interacción mostrando el cambio de sistema de coordenadas en un objeto de la clase Coordinate

Para cambiar el sistema a una coordenada se invoca el método *expressIn: aNewCoordinateSystem* igual que en la clase *SingleLocation*. La implementación de este método será similar a la otra pero retornará una nueva coordenada expresada en el sistema de coordenadas que se desea (figura 4.3.14. Referencia 1).

Como muestra la figura (figura 4.3.14. Referencia 2) cada sistema de coordenadas implementa el traspaso de una coordenada expresada en cualquier sistema.

En comparación con los sistemas de referencia también es necesario especificar un sistema de coordenadas base al cual todos saben transformarse. En esta tesis se tomará como sistema de coordenadas base **“el Sistema Rectangular Tridimensional”**.

Similarmente al sistema de referencia, todo sistema de coordenadas implementará 2 métodos, en donde el primero retorna una colección de unidades en el sistema de coordenadas actual partiendo de una colección expresada en el sistema base y el segundo retorna un conjunto de unidades en el sistema de coordenadas base partiendo de un conjunto en cualquier sistema.

El cambio de sistema de coordenadas además de ser un cambio de sistema común podría implicar otras modificaciones adicionales como proyectar el punto, cambiar el elipsoide de referencia, la dimensión, etc.

En este punto es de destacar el comportamiento similar de los sistemas de referencia y de los sistemas de coordenadas, en la sección final de este capítulo se explica el patrón “Quantity” de Fowler, el cual abstrae este comportamiento similar y lo resuelve. También se verá que la clase *UnitMagnitude* se comporta de manera semejante.

Para finalizar falta el cambio de escala, tal cambio es efectuado desde la unidad de magnitud, cada objeto de esta clase almacena la *unidad* y eso hace que en cualquier momento se pueda cambiar a otra diferente.

En resumen,

1. La clase ReferenceSystem cambia la orientación y desplazamiento
2. La clase CoordinateSystem cambia la proyección, la figura de referencia y/o el sistema de coordenadas.
3. La clase UnitMagnitude cambia la escala.

4.3.4.2. OPERACIONES ENTRE OBJETOS CON DIFERENTES SISTEMAS DE REFERENCIA

Gracias a la arquitectura lograda es posible operar entre objetos aunque éstos se encuentren en cualquier sistema de referencia. Básicamente las operaciones antes de comenzar unificarán el sistema de todos los objetos involucrados y luego procederán comúnmente.

Por ejemplo, para calcular la distancia desde 2 objetos de la clase SingleLocation, llamésmole aSingleLocation1 y aSingleLocation2 se le pedirá al objeto aSingleLocation1 la distancia con respecto al otro mediante el método distanceTo: aSingleLocation2, tal método como primera instancia expresará el objeto aSingleLocation2 en un sistema de referencia igual al del objeto aSingleLocation1 y luego delegará la responsabilidad a las 2 coordenadas aCoordinate1 y aCoordinate2 pertenecientes a los 2 objetos del tipo SingleLocation respectivamente, de esta manera se invocará desde aCoordinate1 el método distanceTo: aCoordinate2 el cual cambiará el sistema de coordenadas de aCoordinate2 al sistema de coordenadas que posea aCoordinate1 y luego hará el cálculo.

El cálculo de distancia dependerá del sistema de coordenadas que lo efectúe, pues variará si tenemos un sistema matemático o uno geodésico, este último efectúa la distancia sobre el elipsoide.

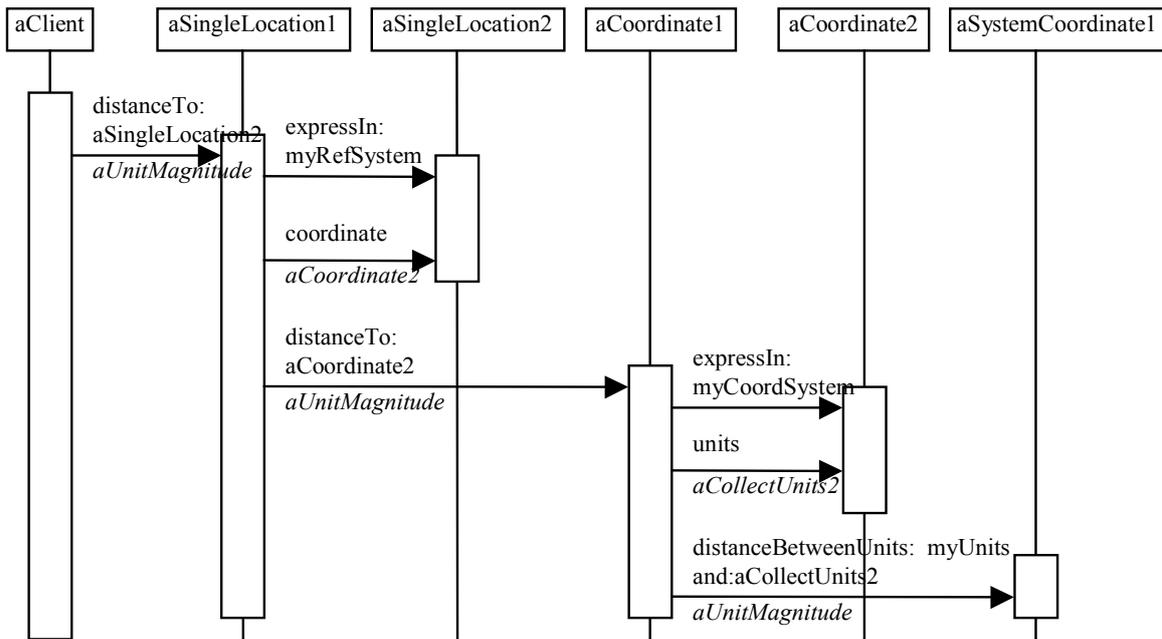


Figura 4.3.14a. Diagrama de interacción mostrando el cálculo para distancia

El ejemplo muestra el comportamiento frente al cálculo de distancia, para las demás operaciones el procedimiento es similar, la idea es ir delegando responsabilidades hasta llegar al punto en el que se efectúa la operación, en ese momento se puede operar con tranquilidad pues el mecanismo ya unificó todos los objetos a un mismo sistema.

4.3.5. Patrones de diseño utilizados

En esta sección se presentarán los patrones de diseño conocidos en la literatura actual que fueron utilizados en el modelo de sistemas de referencia, también se replantean muchos de los diseños vistos anteriormente, pero ahora con el propósito de precisar la *interacción* entre los componentes del modelo y los patrones empleados.

El empleo de patrones no solo mejora el modelo sino que brinda una mayor confiabilidad por ser soluciones probadas y conocidas como flexibles, mantenibles y correctas.

4.3.5.1. PATRÓN STRATEGY (GAMMA)

Intención

“Define una familia de algoritmos, encapsula cada uno de ellos y los hace intercambiables. La estrategia logra que el algoritmo varíe independientemente del cliente que lo usa.”

Aplicación del patrón Strategy en el modelo de Sistemas de Referencia

En la clase `MathematicalCoordinateSystem` se definió una dimensión, tal dimensión se modeló como una estrategia para este tipo de sistemas de coordenadas.

Si no se hubiera empleado el patrón la jerarquía en lugar de ser la presentada en la que solo se cuenta con las subclases `Polar` y `Rectangular`, habría que haber incorporado más subclases teniendo `Rectangular 2D`, `Rectangular 3D`, `Polar 2D`, `Polar 3D`, etc., y así con todas los demás sistemas que no hemos detallado por simplicidad. Esto no sería correcto porque agrega redundancia al modelo y además no abstrae las características comunes de todos los sistemas de 2 dimensiones como ser la presencia de una proyección y de los de 3 dimensiones.

Al incorporar esta estrategia, cada una tiene la responsabilidad de realizar las conversiones adecuadas comportándose en el espacio tridimensional o bidimensional, y además en el caso de éste último implementar los métodos adecuados para interactuar con la proyección para el cálculo de operaciones de transformación.

De esta manera cuando se le requiere al sistema de coordenadas que se transforme a otro, éste le pide a su estrategia que cambie para que ésta aplique o no la proyección adecuada.

4.3.5.2 PATRÓN “QUANTITY” (‘OBSERVATIONS AND MEASUREMENTS’ DE FOWLER)

Intención

“Encapsula el concepto de medida combinando un valor con una unidad.”

Motivación

Muchos sistemas guardan información de objetos del mundo real como atributos simples y la manera más simple de hacerlo es mediante un valor.

Como ejemplos podríamos mencionar, la altura, el peso, un valor monetario, etc. Tales valores carecen de sentido si no van acompañados de una unidad, de esta manera diremos 1.70 metros, 58 kilogramos y 50 pesos. La unidad clarifica el significado del número.

El patrón `Quantity` encapsula el valor junto con su unidad y los trata como un nuevo objeto, definiendo operaciones tanto aritméticas y comparativas.

Además las unidades deben permitir convertir valores de una unidad a otra. Para esto Fowler propone una nueva clase denominada **`ConversionRatio`**. Esta clase conoce como transformar un valor a otro nuevo en otra unidad.

Aplicación del patrón Quantity en el modelo de Sistemas de Referencia

Tal como fue definido, el patrón describe el concepto de unidad y abstrae todo su comportamiento general.

Dicho patrón es aplicable a todo objeto que se represente como un valor sumado a un sistema que lo interprete, ya sea un sistema de medida, un sistema monetario, etc. Además, se cumple la propiedad de poder realizar conversiones entre diferentes sistemas.

En el modelo de sistemas de referencia nos encontramos que tal característica se encuentra presente y mucho más aún, se repite en varios objetos diseñados hasta el momento.

Así, es el caso del objeto *SingleLocation*, *Coordinate* y *UnitMagnitude*. Observando detenidamente su comportamiento se llegó a la conclusión que no sólo eran similares sino que encuadraban perfectamente con el patrón descripto.

Considerando la importancia que presenta la aplicación del patrón en el modelo debido a que define el comportamiento de los principales objetos, se explica en la sección previa incorporando todos los detalles del impacto que produce y la ampliación del diseño hasta ahora presentado.

4.3.6. Aplicación del Patrón Quantity al modelo

4.3.6.1. INTRODUCCIÓN

En esta sección se presentará como se realizó el diseño basándose en el patrón utilizado. Primeramente se detallan los motivos que llevaron a introducir el patrón.

El patrón se aplicó porque se encontró un comportamiento similar en determinados objetos del modelo que encuadraban perfectamente con el del patrón *Quantity*.

Uno de los objetos a los que se aplicó es a la clase *UnitMagnitude* la cual representa una magnitud independientemente de la unidad en la que está expresada. Hay muchas unidades pero la magnitud de un objeto es única, a pesar de que existan muchas maneras de indicarla. Por ejemplo cuando se menciona una medida de algún objeto, es equivalente decir, 100.000 cm., 1.000 m., 1 km., etc. En consecuencia la clase *UnitMagnitude* conoce como transformar sus valores cambiando la unidad y realiza operaciones entre magnitudes con diferentes unidades.

Observando las características de esta clase se puede relacionar con los sistemas de referencia, una ubicación puede ser expresada de diferentes formas, con diferentes sistemas de referencia pero siempre está indicando un mismo lugar, un mismo punto en la tierra, simplemente que algunos usuarios conocerán por ejemplo el valor en el sistema Mercator, otros en el sistema Geodésico Inchauspe, etc., obteniendo de esta manera infinitas expresiones para indicar una “*misma ubicación geográfica*”.

Los sistemas de referencia expresan un idioma, expresan un mecanismo para que un usuario que lo conozca sepa en donde encontrar una ubicación terrestre.

Así, el patrón se aplica en nuestro modelo en la clase SingleLocation, dicha clase indica un punto único en la esfera terrestre, también el patrón se aplica a la clase UnitMagnitude.

Cada objeto del tipo SingleLocation es configurable cambiándole el sistema de referencia, análogamente al objeto UnitMagnitude al cual se le cambia la unidad métrica.

Frente a este cambio de unidad o de sistema de referencia el objeto en cuestión realiza transformaciones internas para pasar sus valores a valores nuevos expresados en la nueva unidad o sistema.

Como se definió en la introducción a este modelo, el sistema de referencia define el marco de referencia, es decir, provee una orientación y desplazamiento para poder ubicar un objeto terrestre, pero luego es necesario determinar el sistema de coordenadas.

El sistema de coordenadas también es un ejemplo ideal para aplicar el patrón nuevamente, aplicándolo de esta manera a la clase Coordinate.

La clase Coordinate encapsula el valor de una coordenada suponiendo un marco de referencia fijo y puede expresarse mediante muchos sistemas de coordenadas, como el rectangular, el geodésico, etc., y en todos los casos al cambiar el sistema de coordenadas se hacen cálculos para obtener los nuevos valores.

Siguiendo con el análisis, cada coordenada expresada en algún sistema correspondiente tendrá un conjunto de valores los cuales podrán ser longitudes, ángulos, etc., las cuales también se modelan en base al patrón Quantity.

En todos los casos se encontró una característica común, el objeto tiene muchas maneras equivalentes de expresarse, cada forma de expresión es equivalente a otra y puede transformarse de una a otra cambiando el valor pero siempre diciendo lo mismo, indicando lo mismo. Además cada objeto es capaz de operar con otro similar pero con unidades diferentes, realizando las conversiones adecuadas.

De esta manera aplicando sucesivamente el patrón obtenemos un resultado totalmente flexible para el usuario y se logra independencia de los sistemas utilizados.

El usuario puede en todo momento pedirle a cualquier unidad que cambie su sistema actual y le retorne el nuevo valor.

En conclusión, cuando el usuario pida a un objeto del tipo SingleLocation que retorne su valor, éste retornará una coordenada la cual se encuentra definida en base al sistema de referencia que posea el objeto SingleLocation.

A dicha coordenada es posible también pedirle que retorne su valor, el cual será un conjunto de unidades de magnitud (UnitMagnitude), dichos valores dependerán del sistema de coordenadas que posea actualmente la coordenada.

Por último a cada objeto UnitMagnitude también se le puede pedir su valor, el cual será un número que dependerá de la unidad que posea el objeto, la cual podría ser metros, centímetros, grados, radianes, etc.

De esta manera el usuario puede cambiar la magnitud sin cambiar el sistema de coordenadas ó cambiar el sistema de coordenadas sin cambiar el sistema de referencia y los puede intercambiar según desee obteniendo muchas maneras diferentes aunque todas expresen la misma ubicación geográfica.

“ Cambiamos el idioma pero el contenido de la información es el mismo”

4.3.6.2. CLASES NUEVAS DERIVADAS DEL PATRÓN

En la sección previa ya se describieron las clases a las cuales se les aplicó el patrón. Como el patrón Quantity se encontraba en varias partes del modelo y para una mejor modelización, fue necesaria la creación de clases abstractas, las cuales definen todo el comportamiento general del patrón y representan las clases a partir de las cuales derivarán las que tenemos hasta el momento.

1. Unit

Representa un valor junto con un sistema de unidad. Cada unidad puede expresarse en cualquier sistema diferente siempre que éste sea compatible con el que tenga actualmente, al cambiar el sistema recalcula su valor.

Además de este comportamiento cada subclase de unidad tendrá definidas las operaciones particulares de cada una.

Subclases de la clase Unit serán las clases UnitMagnitude, UnitCoordinate y UnitLocation². Los sistemas de unidades asociadas a cada una son el sistema de longitud (metros, centímetros, etc.), el sistema de coordenadas (rectangular, geodésica, etc.) y el sistema de referencia (WGS84, Inchauspe, etc.) respectivamente.

2. UnitFamilySystem

Representa una familia de sistemas de unidades.

Cada familia está formada por un conjunto de sistemas de unidades de los cuales uno es el sistema de unidad base. Tal sistema de unidad base posee la característica que todos los demás sistemas de la familia conocen como transformar un valor hacia este sistema base y el sistema base conoce a su vez como transformar cualquier valor a cualquier otro sistema perteneciente a la misma familia.

Dicho de otro modo el sistema base establece un mecanismo para poder realizar conversiones entre los distintos sistemas.

3. UnitSystem

Especifica el comportamiento general de todo sistema de unidad.

En el modelo tenemos diferentes sistemas de unidades, por ejemplo, los sistemas de unidades de longitud, sistemas de unidades de coordenadas, sistemas de referencia, etc.

Cada sistema de unidad pertenece a una familia y cuenta con una función de conversión. La función de conversión es configurable y se puede cambiar en cualquier momento dependiendo del sistema de unidad base que posea la familia, pues esta función indica como convertir un determinado valor desde el sistema actual al sistema base y su relación inversa.

4.3.6.3. DISEÑO DEL PATRÓN QUANTITY

² Notar que se cambiaron los nombres de determinadas clases ya presentadas, la clase Coordinate pasó a llamarse UnitCoordinate y la clase SingleLocation pasó a nombrarse como UnitLocation, el motivo de tal decisión es para hacer más legible y comprensiva la lectura del modelo resultante una vez que el patrón ya fue aplicado.

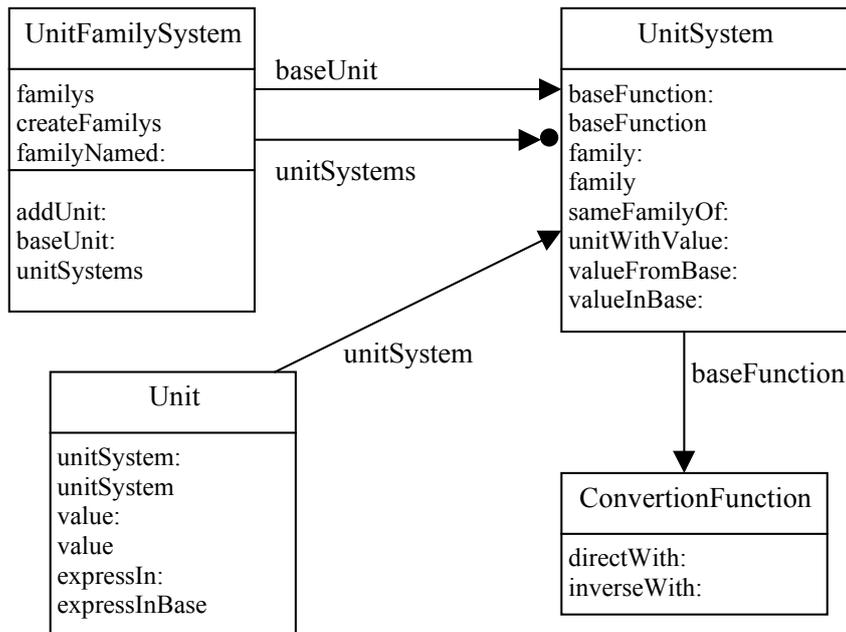


Figura 4.3.15. Diseño del pattern Quantity

En el diseño se ilustran las clases descritas anteriormente y la relación entre ellas.

Es importante notar que tal como el patrón lo expresa, el comportamiento de un objeto de la clase Unit se basa en la capacidad de poder expresar la unidad en cualquier sistema, esto se logra mediante el método *expressIn:* de dicha clase.

Para poder resolver tal método la unidad se interrelacionará con su sistema para poder realizar la conversión, la cual lo hará mediante los métodos *valueFromBase:* y *valueInBase:* que realizan los cálculos en base a la función de conversión que posea.

Como se puede apreciar la función de conversión cuenta con 2 métodos *directWith:* e *inverseWith:*, es decir, se modela una función matemática. Para el modelo, tal función se utilizará para poder convertir un valor cualquiera a un valor expresado en la unidad base y viceversa; la necesidad de implementar tales funciones en forma desacoplada es por el hecho que el sistema de unidad base puede cambiar, por lo que también cambiará dicha función. En secciones posteriores se mencionarán ejemplos de funciones utilizadas para la implementación de los sistemas de referencia, lo cual aclarará más estos conceptos.

Por último se ilustra el diagrama de interacción que expresa el comportamiento mencionado. (figura 4.3.16)

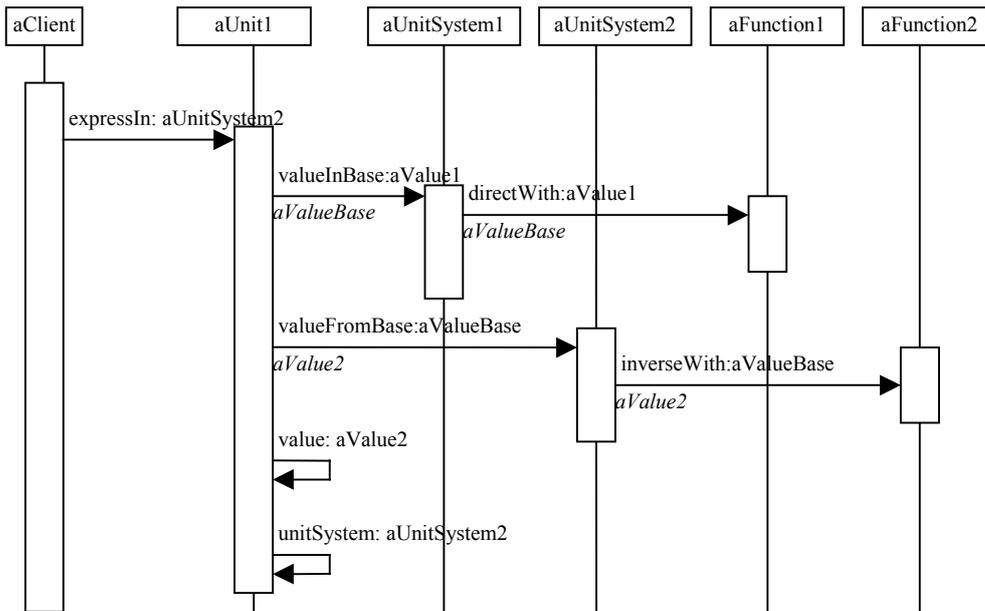


Figura 4.3.16. Diagrama de interacción mostrando el comportamiento del pattern Quantity

4.3.6.4. DISEÑO DE CLASES DEL MODELO LUEGO DE APLICAR EL PATRÓN

Una vez definido el modelo del patrón Quantity con las clases abstractas se procede a determinar la jerarquía resultante de relacionar el modelo del patrón con el modelo de sistemas de referencia obtenido hasta el momento.

En el modelo de sistemas de referencia se destacan 3 clases que son unidades, es decir, que están conformadas por un valor sumado con un sistema de unidad, estas clases son: UnitMagnitude, UnitCoordinate y UnitLocation (figura 4.3.17).

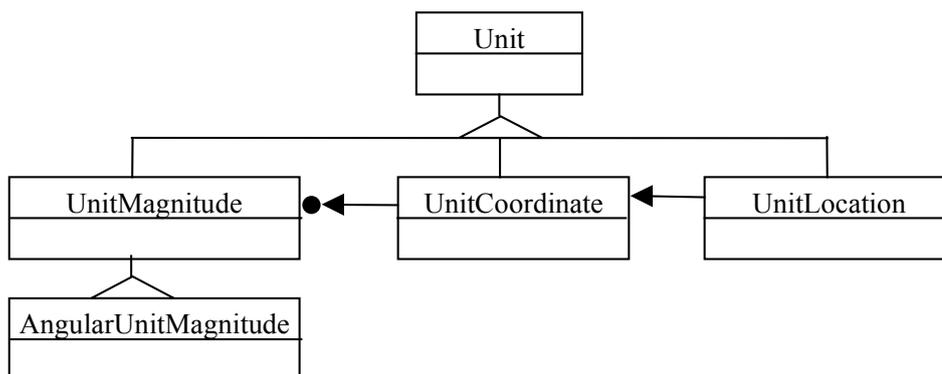


Figura 4.3.17. Jerarquía de la clase Unit

Cada tipo de unidad se corresponde con un tipo de sistema de unidad, de esta forma obtenemos la jerarquía para la clase de sistemas de unidades (UnitSystem).

Tal jerarquía está formada por el sistema angular, el sistema de longitud, el sistema de coordenadas y el sistema de referencia (figura 4.3.18)

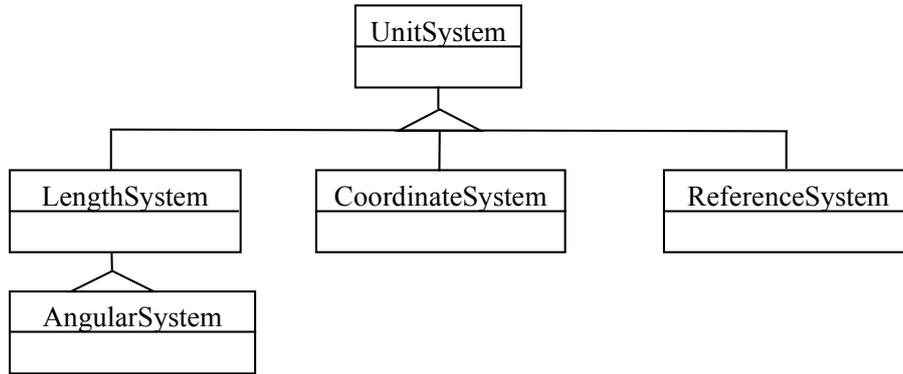


Figura 4.3.18. Jerarquía de la clase UnitSystem

Cada sistema de unidad defina un conjunto de operaciones propio de cada uno, tal conjunto determinará el comportamiento de la clase de unidad correspondiente.

Para ilustrar lo anterior se mencionarán solo unas pocas operaciones de cada sistema como ejemplo.

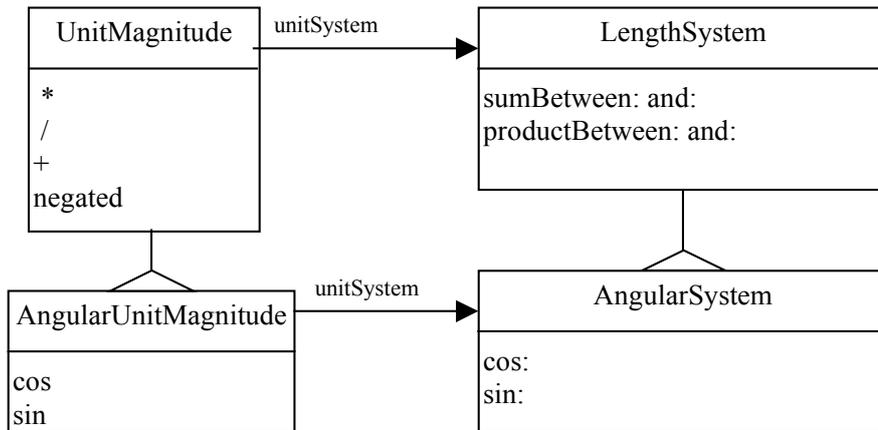


Figura 4.3.19. Comportamiento de magnitudes.

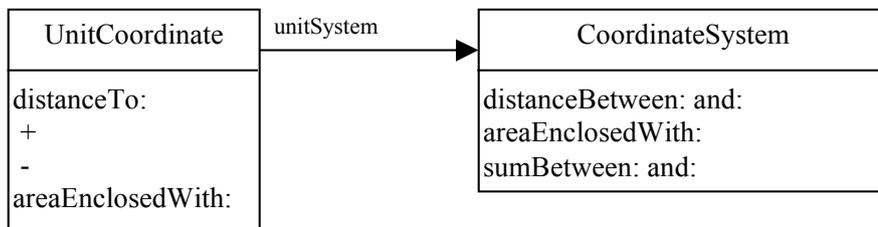


Figura 4.3.20. Comportamiento de coordenadas

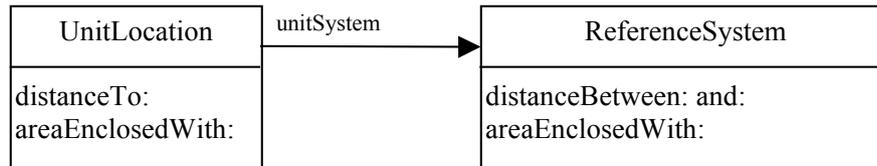


Figura 4.3.21. Comportamiento de las posiciones

Como se puede observar cada unidad realiza sus operaciones delegando la responsabilidad al sistema de unidad correspondiente, el cual realiza finalmente los cálculos.

En conclusión, el sistema establece el comportamiento de la unidad.

4.3.7. Modelo Final de Sistemas de Referencia

Para concluir el análisis del modelo mostramos el diagrama de clases y el diagrama de objetos de acuerdo a todo lo mencionado previamente.

4.3.7.1. DIAGRAMA DE CLASES

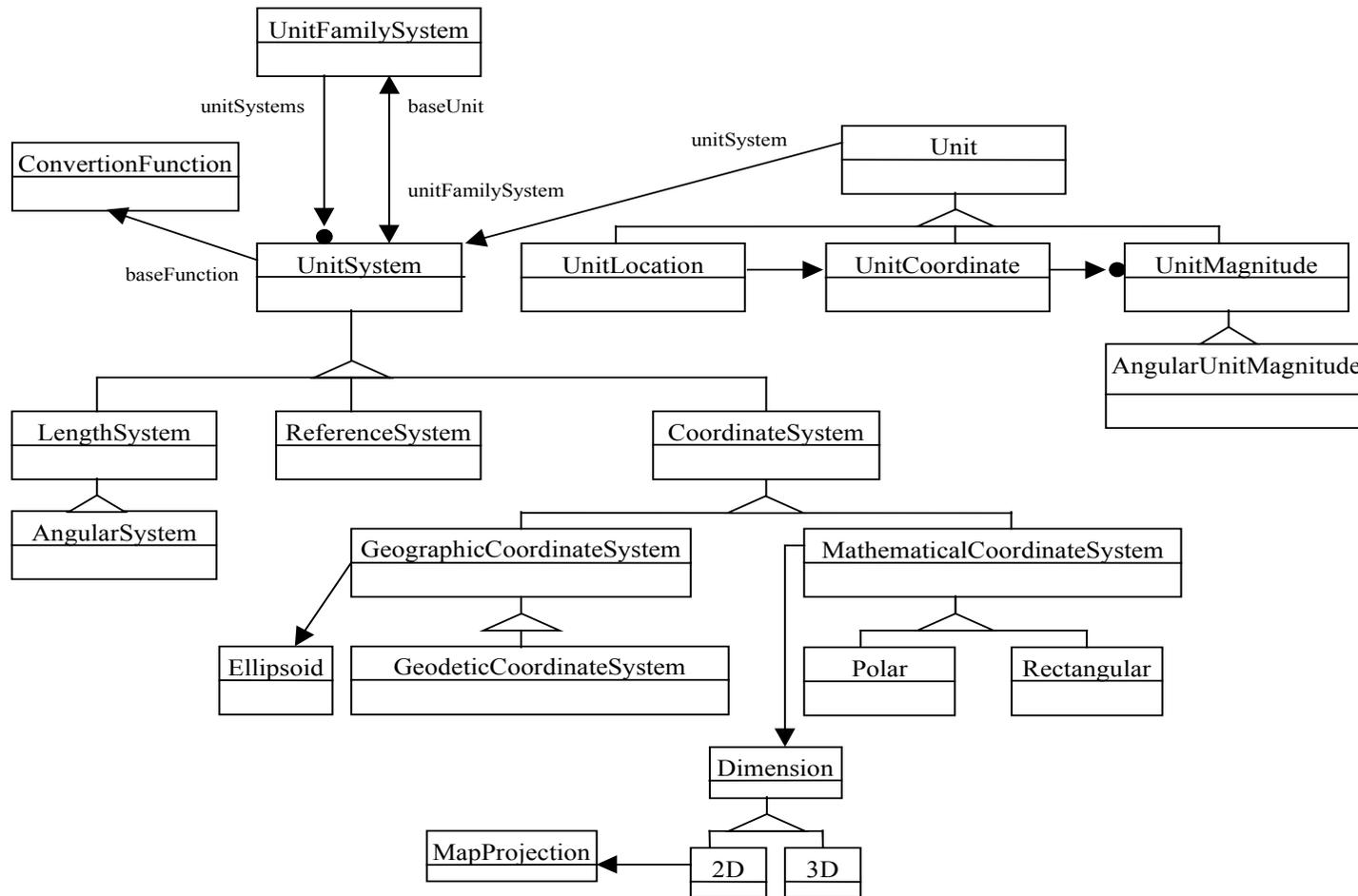


Figura 4.3.22. Modelo Final de Sistemas de Referencia

4.3.7.2. DIAGRAMA DE OBJETOS

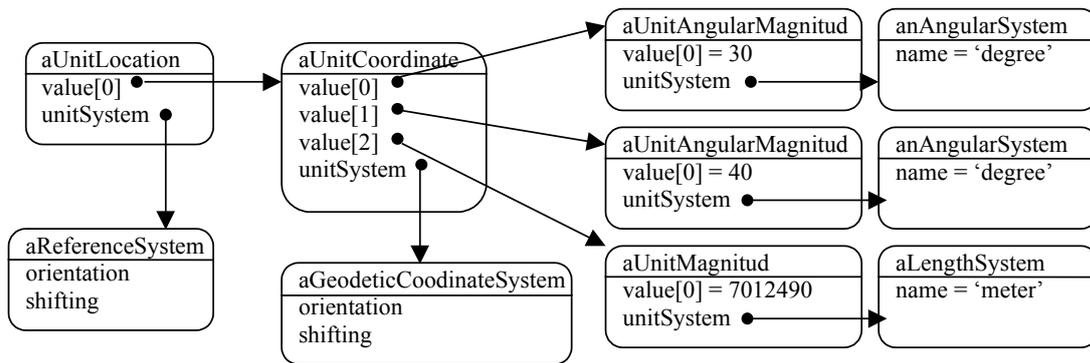


Figura 4.3.23. Diagrama de Objetos

5. Implementación

5.1 IMPLEMENTACION DEL MODELO DE SISTEMAS DE REFERENCIA

5.1.1 Introducción

La implementación del modelo de sistemas de referencia se divide en dos partes: la primera abarca el comportamiento general del sistema y la segunda la instanciación de algunos sistemas de referencia para mostrar el funcionamiento del modelo.

Durante la primera parte se implementó todo lo necesario para poder instanciar y crear cualquier tipo de sistemas de referencia mencionado en el dominio del análisis. Esto abarca la implementación del patrón Quantity, el cual es fundamental para la implementación, pues determina la base y las clases abstractas de las cuales derivarán las clases específicas de los sistemas de referencia. En la misma etapa se procedió a implementar las clases principales del modelo integrándola con el patrón como se explicó en el capítulo anterior.

Una vez que fue finalizada esta primer etapa se seleccionaron algunos tipos de sistemas con el objetivo de mostrar el funcionamiento del modelo, la elección incluyó unidades de magnitud, sistemas de coordenadas, elipsoides y sistemas de referencia.

5.1.2. Implementación del patrón Quantity

Las clases que abarcan la implementación del patrón son: UnitFamilySystem, UnitSystem, Unit y ConversionFunction. Estas clases implementan el comportamiento del patrón mencionado.

Se describirán las tarjetas de clase ³ de cada una de ellas, los métodos remarcados están implementados al final de la sección.

³ Formato General de Tarjetas de Clase.

Clase:	
Subclase de:	
{Atributos de la clase}	
{Métodos de clase}	{Colaboradores}
{Responsabilidades}	{Colaboraciones}

Tarjetas de clase

Clase: "UnitFamilySystem"	
Subclase de: Object	
name unitSystems baseUnit	
createFamilys newWithName: familyNamed: unitNamed:	UnitSystem
baseUnit baseUnit: name name: unitSystem unitSystem: addUnit: unitNamed:	

Clase: "UnitSystem"	
Subclase de: Object	
family baseFunction name	
newWithName: withBaseFunction:	
family family: baseFunction baseFunction: name name: sameFamilyOf: unitWithValue: vaueFromBase: valueInBase:	Unit ConversionFcuntion ConversionFunction

Clase: "Unit"	
Subclase de: Object	
value unitSystem	
newWithValue: withUnitSystem:	
Value value: unitSystem unitSystem: expressIn: expressInBase	UnitSystem UnitSystem

Clase: "ConversionFunction"	
Subclase de: Object	
directWith: inverseWith:	

• **Descripción e implementación de los métodos más relevantes**

UnitFamilySystem:: createFamilies

Este método crea e instancia los sistemas de coordenadas, unidades de magnitud y sistemas de referencia.

Tal método será explicado con detalle en la sección final, pues especifica el alcance de la implementación.

Unit:: expressInBase

Cambia la unidad de sistema a la unidad de sistema base.

En este método la unidad toma el valor que posee actualmente y pide a su unidad de sistema que transforme el valor al valor base equivalente, obteniendo un nuevo valor expresado en la unidad de sistema base.

De esta manera la unidad cambia su sistema actual al sistema base y por consiguiente cambia también el valor adecuándose al nuevo sistema.

"Cambio el valor al base"

value := self unitSystem valueInBase: self value.

"Cambio la unidad de sistema a la unidad de sistema base"

unitSystem := self unitSystem family baseUnit.

^self

Unit:: expressIn: aNewUnitSystem

Cambia la unidad de sistema a la unidad pasada como parámetro.

Este método es similar al anterior salvo que el sistema nuevo es un sistema cualquiera en lugar del sistema base.

Para lograr la transformación se pasa primeramente al sistema base y luego desde el sistema base se pasa al nuevo.

```
| aBaseUnit |
“Verifica que las 2 unidades de sistemas sean compatibles”
(self unitSystem sameFamilyOf: aNewUnitSystem)
ifFalse: [^self error: 'Unidades Incompatibles']
ifTrue:
[“Paso primero a la Unidad de Sistema Base”
aBaseUnit := self expressInBase.
“Paso desde la Unidad de Sistema Base a la nueva”
value := aNewUnitSystem valueFromBase: aBaseUnit value.
unitSystem := aNewUnitSystem.
^self]
```

UnitSystem:: valueFromBase: aBaseValue

Transforma el valor pasado como parámetro aBaseValue el cual es un valor expresado en la unidad de sistema base de la familia a un nuevo valor en el sistema de unidad actual.

Este método delega la responsabilidad a la función de conversión que el sistema de unidad tiene.

```
^ self baseFunction inverseWith: aBaseValue
```

UnitSystem:: valueInBase: aValue

```
^ self baseFunction directWith: aValue
```

Transforma el valor aValue el cual es un valor expresado en la unidad de sistema actual a un nuevo valor expresado en el sistema de unidad base de la familia.

Al igual que el método anterior delega la responsabilidad a la función de conversión que el sistema de unidad tiene.

ConversionFunction:: directWith: aValue

ConversionFunction:: inverseWith: aValue

Estos métodos son abstractos y son redefinidos en cada subclase de la clase `ConversionFunction`.

Esta clase modela el comportamiento de una función matemática, para esta implementación las subclases que derivarán de `ConversionFunction` representarán funciones para poder realizar las conversiones entre un sistema de unidad y el sistema base correspondiente.

Para que quede más claro el concepto mencionaremos un ejemplo para el traspaso de unidades de longitud.

Ejemplo:

Crearemos una clase llamada `ConstantFunction`.

Clase: "ConstantFunction"	
Subclase de: <code>ConversionFunction</code>	
constant	
newWithConstant:	
directWith:	
inverseWith:	

Los métodos serán:

directWith: aValue

"Retorna el resultado de aplicar la función directa"

\wedge self constant * aValue

inverseWith: aValue

"Retorna el resultado de aplicar la función inversa"

\wedge aValue / self constant

Con esta función definida ya podemos crear unidades de longitud. Tomemos la unidad "Meter" como unidad de sistema base, así para crear el sistema de longitud "Centimeter" lo crearemos de la siguiente manera:

```
centimeter := UnitSystem
    newWithName: 'Centimeter'
    withBaseFunction: (ConstantFunction newWithConstant: 1/100).
```

En el ejemplo se observa que la función de conversión sirve para poder pasar desde centímetros a metros y a la inversa.

5.1.3. Implementación de las clases principales del modelo de sistemas de referencia

En esta sección mencionaremos las clases principales del modelo. Una vez implementadas estas clases solo falta realizar las instancias correspondientes para configurar los sistemas con los que el usuario desee trabajar.

Clase: "SingleLocation"	
Subclase de: Unit	
topology	
newWithValue: withReferenceSystem:	
distanceTo:	ReferenceSystem
areaEnclosedWith:	ReferenceSystem
topology	
topology:	

Clase: "Coordinate"	
Subclase de: Unit	
newWithValue: withCoordinateSystem:	
distanceTo:	CoordinateSystem
areaEnclosedWith:	CoordinateSystem
+	CoordinateSystem
-	CoordinateSystem

Clase: "UnitMagnitude"	
Subclase de: Unit	
newWithValue: withLengthSystem:	
*	LengthSystem
**	LengthSystem
+	LengthSystem
-	LengthSystem
/	LengthSystem
=	LengthSystem
>	LengthSystem
arcTan	LengthSystem
ln	LengthSystem
sqrt	LengthSystem

Clase: "AngularUnitMagnitude"	
Subclase de: Unit	
newWithValue: withAngularSystem:	
cos	AngularSystem
sin	AngularSystem
cot	AngularSystem
sec	AngularSystem
tan	AngularSystem

Clase: "LengthSystem"	
Subclase de: UnitSystem	
unitWithValue:	
sumBetween: and:	ConversionFunction
productBetween: and:	ConversionFunction

Clase: "CoordinateSystem"	
Subclase de: UnitSystem	
unitWithValue:	
distanceBetween: and:	
areaEnclosedWith:	
sumBetween: and:	

Clase: "GeographicCoordinateSystem"	
Subclase de: CoordinateSystem	
ellipsoid	
newWithName: withEllipsoid:	
ellipsoid	
ellipsoid:	
distanceBetween: and:	
areaEnclosedWith:	
sumBetween: and:	

Clase: "MathematicalCoordinateSystem"	
Subclase de: CoordinateSystem	
Dimension	
newWithName: withDimension:	
Dimension distanceBetween: and: areaEnclosedWith: sumBetween: and:	

Clase: "ReferenceSystem"	
Subclase de: UnitSystem	
orientation shifting	
newWithName: withOrientation: withShifting:	
orientation orientation: shifting shifting: unitWithValue: distanceBetween: and: areaEnclosedWith: sumBetween: and:	

Observando los métodos que realizan cálculos de distancia, área, etc., vemos que aparecen en varias clases, esto es así porque cada subclase lo implementa teniendo en cuenta el sistema de unidad correspondiente.

5.1.4. Implementación de las Funciones de Conversión

Se desarrollaron diferentes funciones que servirán para poder realizar las transformaciones entre los diferentes sistemas.

Por simplicidad solo mencionaremos la jerarquía y descripción de las clases desarrolladas en este trabajo, la implementación de tales funciones se corresponden con las funciones matemáticas encontradas en la bibliografía investigada.

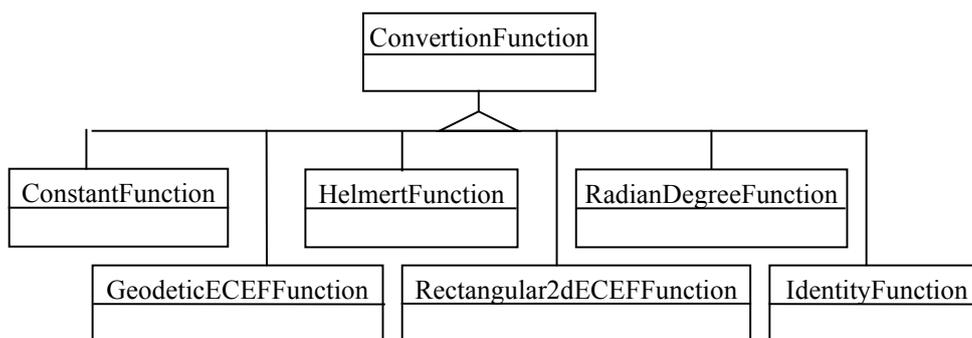


Figura 5.1.1. Jerarquía de las funciones de conversión implementadas

ConstantFunction

Esta función se instancia con una constante.

La función toma un valor y retorna ese valor multiplicado por la constante. La inversa realiza la división.

Es utilizada por los sistemas de longitud, en párrafos anteriores fue mencionada y explicada a través de un ejemplo (ver sección 5.1.2).

GeodeticECEFFunction

Esta función se instancia con un elipsoide.

La función toma un valor expresado en coordenadas geodésicas (latitud, longitud y altura) y retorna el valor equivalente expresado en coordenadas rectangulares (x,y,z). El elipsoide es necesario ya que la altura es con relación a éste.

HelmertFunction

Esta función se instancia con un sistema de referencia origen.

La función toma una coordenada y retorna una nueva como resultado de aplicar la orientación y desplazamiento del sistema de referencia origen.

Esta función es utilizada para la conversión entre los sistemas de referencia.

Cabe aclarar que hay muchas otras funciones que también podrían implementarse y utilizarse con el mismo fin, como la función de Molodensky, en realidad la función de conversión puede seleccionarse por el usuario, pues es configurable desde el sistema de referencia.

IdentityFunction

Esta función tal como su nombre lo indica retorna el mismo valor de origen.

Se utilizará sobre todo en los sistemas de unidades seleccionados como sistemas bases.

RadianDegreeFunction

Esta función transforma un valor en radianes al valor equivalente en grados.

Es utilizada por los sistemas angulares.

Rectangular2dECEFFunction

Esta función se instancia con una proyección determinada.

Toma un valor expresado en coordenadas rectangulares de 2 dimensiones que representa un punto proyectado y retorna una coordenada rectangular (x,y,z) que representa el punto de original lo cual se obtiene aplicando la inversa de la proyección.

5.1.4. Instanciación del modelo

Finalmente se procederá a instanciar las clases para crear algunos sistemas y así poder mostrar el funcionamiento del modelo.

Cabe aclarar que la instanciación presentada es un estado inicial para que el usuario pueda comenzar a trabajar, si el usuario detecta en tiempo de ejecución que necesita algún otro tipo de sistema que no hemos instanciado al comienzo, puede agregarlo y éste se incorpora a los demás quedando disponible para futuras ejecuciones.

La instanciación del modelo se encuentra en el método de clase *createFamilys* de la clase *UnitFamilySystem*, aquí mencionaremos algunos de los sistemas que se encuentran en ese método y mostraremos como están instanciados.

Sistemas de longitud

Sistema	Función de conversión
Megameter	ConstantFunction 1000000
Kilometer	ConstantFunction 1000
Hectometer	ConstantFunction 100
Decameter	ConstantFunction 10
Meter	IdentityFunction
Decimeter	ConstantFunction 1/10
Centimeter	ConstantFunction 1/100
Milimeter	ConstantFunction 1/1000

El sistema Meter fue el elegido para ser el sistema base dentro de los sistemas de longitud.

Sistemas Angulares

Sistema	Función de conversión
Degree	IdentityFunction
Radian	RadianDegreeFunction
Minute	ConstantFunction 1/60
Second	ConstantFunction 1/3600

El sistema Degree fue el elegido para ser el sistema base dentro de los sistemas angulares.

Sistemas de Coordenadas

Sistema	Función de conversión
WGS84Geodetic (sistema geodésico basado en el elipsoide WGS84) latitud-longitud-altura	GeodeticECEFFunction - Ellipsoid WGS84
NAD97Geodetic (sistema geodésico basado en el elipsoide NAD97) latitud-longitud-altura	GeodeticECEFFunction - Ellipsoid NAD97
Rectangular3d (coordenadas ECEF) X Y Z	IdentityFunction
Rectangular2dMercator (coordenadas rectangulares 2d proyectadas – proy. Mercator) X Y	Rectangular2dECEFFunction – Proyección Mercator
Rectangular2dPolyconic (coordenadas rectangulares 2d proyectadas – proy. Polyconic) X Y	Rectangular2dECEFFunction – Proyección Polyconic

El sistema Rectangular3d fue el elegido para ser el sistema base dentro de los sistemas de coordenadas.

Sistemas de Referencia

Sistema	Función de conversión
WGS84	IdentityFunction
Inchauspe	HelmertFunction – Orientation = (0,0,0) Shifting = (-148, 136, 90)

El sistema WGS84 fue el elegido para ser el sistema base dentro de los sistemas de referencia.

Notar que la orientación y desplazamiento de cada sistema es en relación al sistema seleccionado como base, en este caso el sistema WGS84, si se cambia la base

habría que ajustar tales parámetros en las funciones de conversión de los demás sistemas.

Como puede observarse gracias a la flexibilidad lograda pueden armarse muchos tipos de sistemas a gusto del usuario combinando las funciones de conversión, con elipsoides, proyecciones, etc.

5.2 IMPLEMENTACION DE LA APLICACION

- ✓ Nombre de la Aplicación: “Cambio de Sistemas de Referencia”
- ✓ Utilidad: Visualización gráfica y textual de transformaciones de Sistemas de Referencia.
- ✓ Paradigma de Desarrollo: Orientado a Objetos.
- ✓ Herramienta de Desarrollo: Visual Works 2.5.1
- ✓ Sistemas Operativos Soportados: Windows 9X, Windows NT.

5.2.1 Objetivo de la aplicación

El objetivo de la aplicación es utilizar el modelo desarrollado en esta tesis para visualizar de manera gráfica o textual la transformación del Sistema de Referencia de un conjunto de datos.

5.2.2 Características Generales

La aplicación presenta al usuario 2 conjuntos de datos (Locations) con sus correspondientes Sistemas de Referencia, Sistemas de Coordenadas y puntos que las conforman. Las mismas son:

- Perfil humano:
 - Conjunto de Coordenadas cuya topología se asemeja al perfil de un rostro humano.
 - Sistema de Referencia: WGS84.
 - Sistema de Coordenadas: Geodésico con Elipsoide WGS84.

- América:
 - Conjunto de Coordenadas cuya topología se asemeja a la forma del continente Americano.
 - Sistema de Referencia: Inchauspe.
 - Sistema de Coordenadas: ECEF con Elipsoide Inchauspe

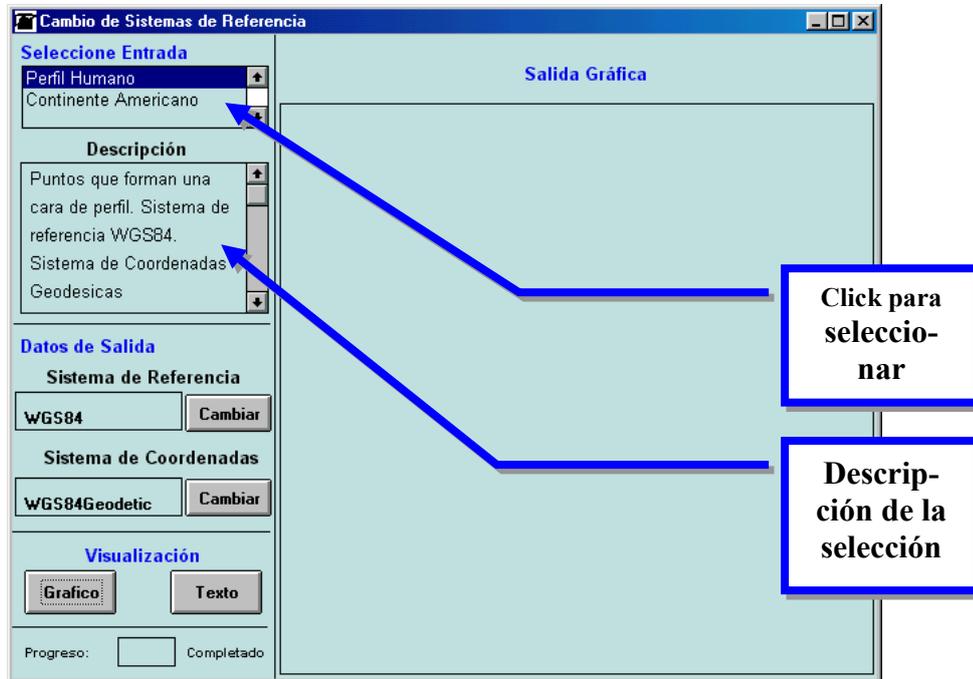
A partir de aquí se selecciona una de ellas; la aplicación mostrará la información de la Location. Seguidamente se debe indicar las características del sistema de referencia de salida, así como del sistema de coordenadas.

Si el resultado puede visualizarse, el usuario tendrá la posibilidad de optar por la salida gráfica o en forma de texto. Si no puede visualizarse (es tridimensional), el usuario solo tendrá la opción de la salida textual.

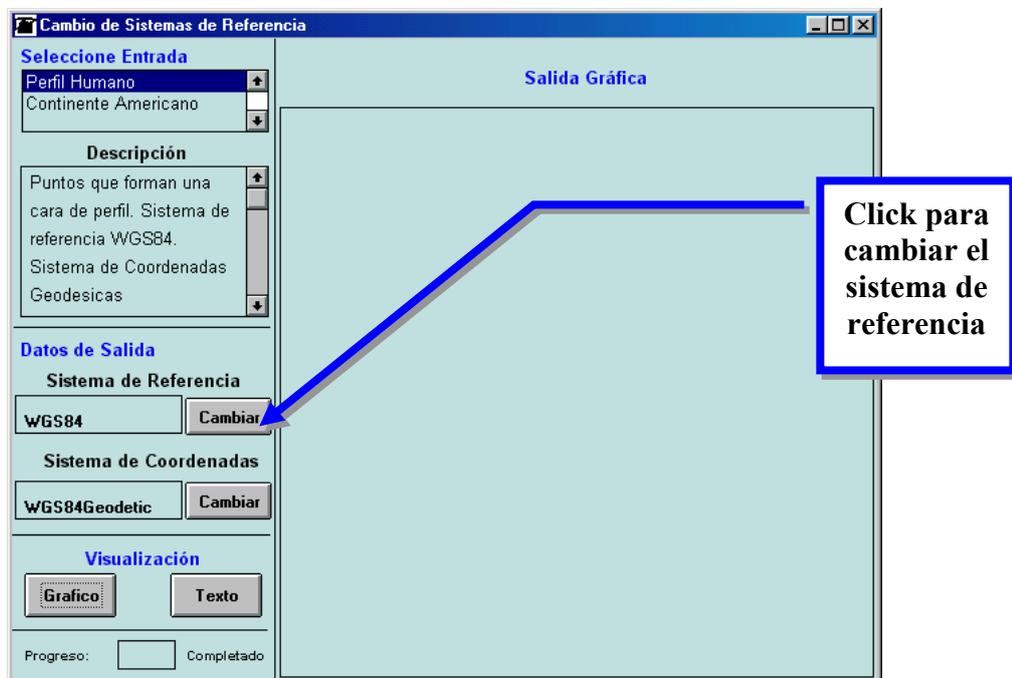
5.2.3 Ejemplo de uso de la Aplicación

En este ejemplo se mostrara la transformación y visualización gráfica de la Location “Perfil Humano”.

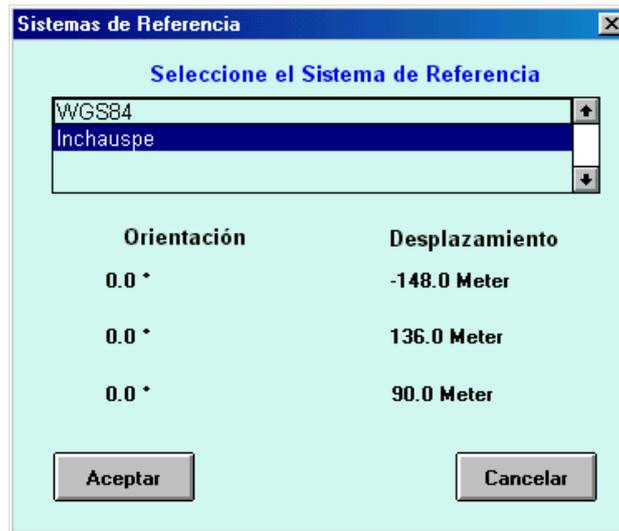
1. Selección de la Location “Perfil Humano”:



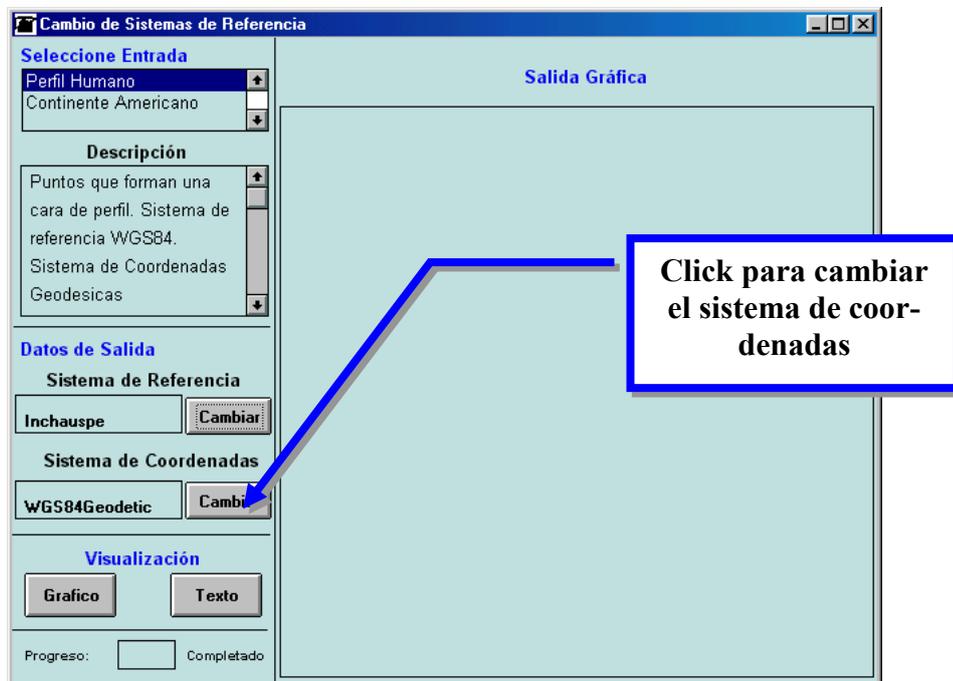
2. Selección del Sistema de Referencia de Salida:



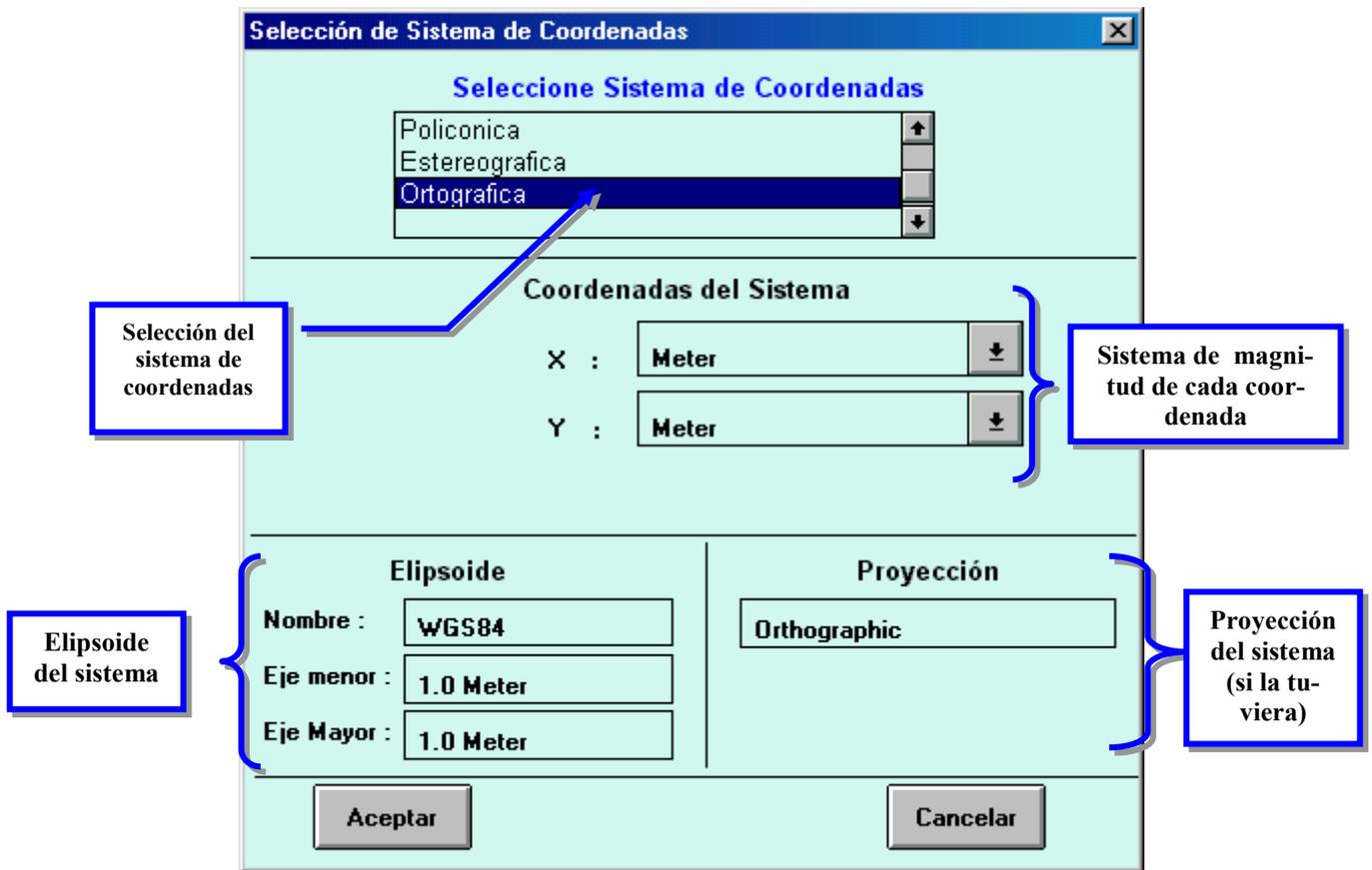
Al seleccionar el Sistema de referencia de la lista, se informan las características de lo elegido.



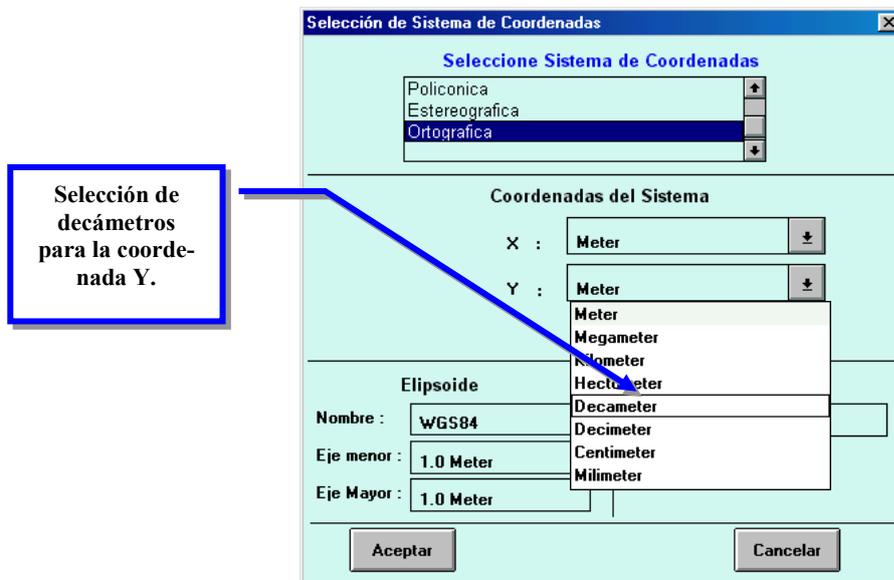
3. Selección del Sistema de Coordenadas de Salida:



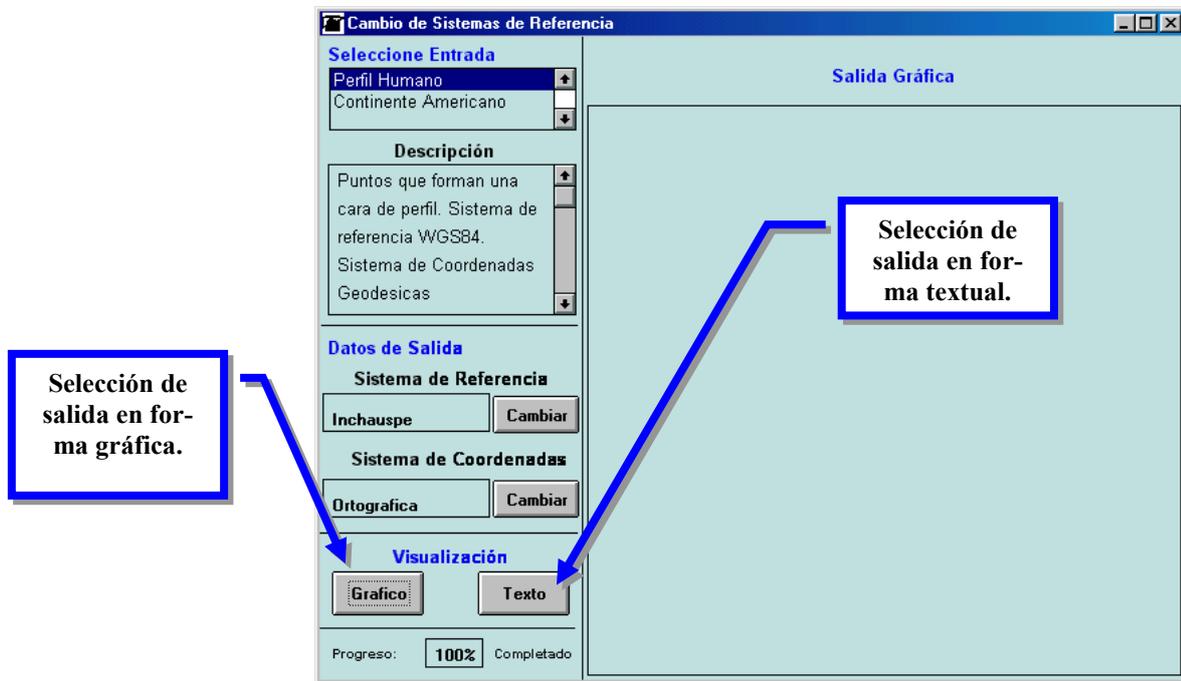
Al seleccionar el Sistema de Coordenadas de la lista, se informan las características de lo elegido.



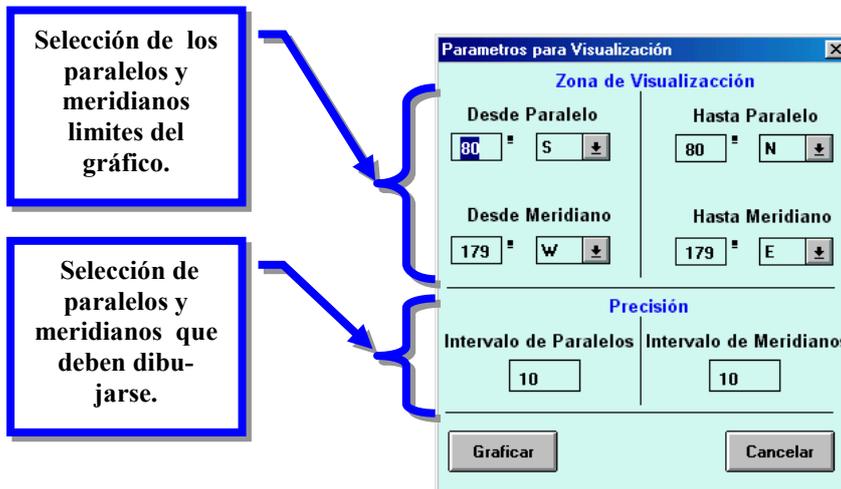
Puede cambiarse la magnitud de las Coordenadas para la salida:



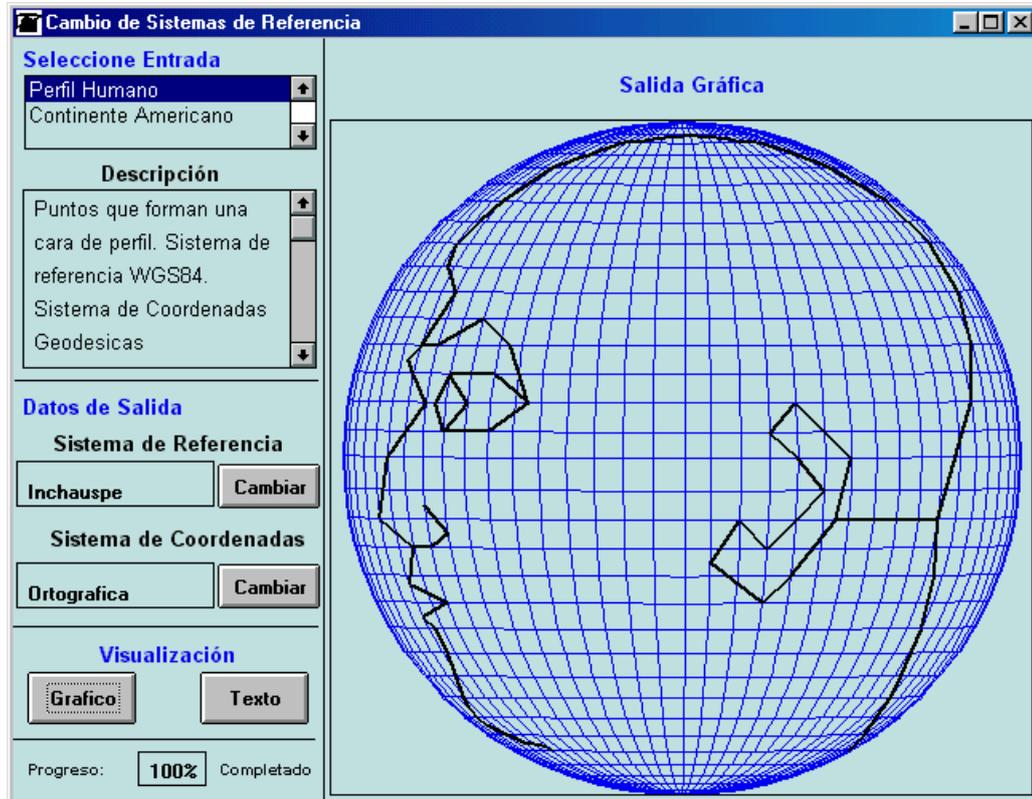
4. Seguidamente se selecciona la forma de salida para los resultados de la transformación.



Si la salida seleccionada es gráfica, se pide el ingreso de los parámetros de visualización:



5. Presionando el botón GRAFICAR, se obtiene la salida gráfica con la precisión y la zona seleccionada.



6. Conclusiones

6.1. CONCLUSIONES

Esta tesis demuestra la flexibilidad que un modelo puede alcanzar cuando el dominio y el paradigma seleccionados para obtenerlo son adecuados. El modelo resultante es, así, una estructura flexible y poderosa.

Cada uno de sus componentes fueron diseñados con el objetivo de hacer que el modelo sea fácilmente enriquecido e instanciado con distintos Sistemas de Referencia, Coordenadas o Magnitudes.

La combinación, transformación y funcionalidad general del modelo se realiza con el mismo criterio; sin importar la magnitud del objeto en cuestión, siendo el criterio general: “objeto exprésate con estas características”.

Dado que el fin de la utilización de un software SIG mas un conjunto de datos es encontrar nuevas relaciones entre los mismos y obtener la información “oculta” en ellos, cuanto mejor se modelen los datos, más es la información que podemos obtener.

Este modelo es un buen “*punto de partida*” o “*piedra fundamental*” para construir a partir de él una aplicación SIG Orientada a Objetos, ya que permite representar gran cantidad de información geográfica para cada entidad; y las conversiones y cálculos (bases para la búsqueda de información) están resueltas de antemano como consecuencia de su comportamiento, solo restaría agregar las características conceptuales específicas de la aplicación.

6.2. CONTRIBUCIONES

Las principales contribuciones de este trabajo son:

- ✓ Se ha definido un modelo unificado de conceptos de funcionalidad de datos geográficos.
- ✓ Se han aplicado los principios de la tecnología de objetos en su concepción actual al campo de referenciamiento terrestre, logrando los beneficios conocidos del paradigma OO, como reusabilidad, extensibilidad y facilidad de mantenimiento, a través de un modelo OO para las aplicaciones en el dominio.
- ✓ Se ha descubierto que la aplicación recurrente de ciertos patrones de diseño resultó en un modelo con comportamiento homogéneo y gran flexibilidad.
- ✓ Se ha construido un modelo que permite extender cualquier aplicación OO con la funcionalidad geográfica que sus entidades requieran.

6.3. EXTENSIONES

Esta tesis resuelve todo lo concerniente a referenciamiento espacial, pero para futuras investigaciones sería adecuado extender este comportamiento para darle características *espacio-temporales*.

Si bien es cierto que un dato geográfico conceptual posee una posición terrestre actual, la misma puede ser modificada con el transcurso del tiempo. Esto se debe a diversos factores, entre ellos se encuentran :

- ✓ el desplazamiento natural de los continentes;
- ✓ la modificación mínima pero constante de las dimensiones de la esfera terrestre;
- ✓ otros factores propios del objeto (podría moverse por si mismo).

La base de esta extensión sería poder conocer o proyectar la posición de cada objeto combinando las características espaciales y temporales del dato.

7. Bibliografía

1. Aronoff S.: "*Geographic Information Systems: A Management Perspective*". WDL Publications. Ottawa. Canada.1989.
2. Balaguer F., Gordillo S. and Das Neves F: "*Patterns for GIS Application Design*". In proceedings of PloP. Programming Language of Patterns, 1997.
3. Fowler, M : "*Analysis Patterns : Reusable Object Models*". Addison Wesley, 1996.
4. Gamma, R. Helm, R. Johnson, J. Vlissides: "*Design Patterns. Elements of reusable Object-Oriented Software*". Addison Wesley, 1995.
5. Gordillo, S., Balaguer, F. and Das Neves, F.: "*Generating the Architecture of GIS Applications with Design Patterns*". Proceedings of the ACM-GIS'97: Advances in Geographic Information Systems. Las Vegas, USA, november 13-14, 1997.
6. Kemp K.: "*Fields as a framework for integrating GIS and environmental process models. Part 1: Representing spatial continuity*". Transactions in GIS, vol. 1 nro. 3, p. 219, 1997.
7. Kemp K.: "*Fields as a framework for integrating GIS and environmental process models. Part 2: Specifying field variables*". Transactions in GIS, vol. 1 nro. 3, p. 325, 1997.
8. Laurini, R. and Thompson D.: "*Fundamentals of Spatial Information Systems*" . Academic Press, 1992.
9. Lopez de Oliveira, J., Pires F. and Medeiros C.: "*An environment for Modelling and Design of Geographic Applications*" . GeoInformatica. vol.1, number 1, p. 29-58, april 1997, Kluwer Academic Publishers
10. Medeiros, M. A. Casanova and G. Camara: "*The Domus project. Building an OODB GIS for environmental control*" Proceedings of IGIS'94. International Workshop on Advanced Research in GIS, Springer Verlag LNCS, N. 884, pp 45-54
11. Perdomo R., Del Cogliano D.: "*GPS del campo al Informe Final*" Facultad de Ciencias Astronómicas y Geofísicas. UNLP. 1994.
12. Open GIS Consortium (OGC), 1996B, *The Open GIS Guide-A Guide to Interoperable Geo-processing*, Available at <http://ogis.org/guide/guide1.htm>
13. Pariente D.: "*Estimation, Modelisation et Langage de Declaration et Manipulation de Champs Spatiaux Continus*". PhD Thesis. University of Lyon. France. 1994.
14. Rumbaugh, M. Blaha, M. Premerlani and W. Lorensen: "*Object-Oriented Modeling and Design*". Prentice Hall, Englewoods Cliff, New Jersey, 1991
15. Star J. and Estes J: "*Geographic Information Systems. An Introduction*". Prentice Hall. 1990.