

Índice general

1.1	
Introducción.....	¡Error! Marcador no definido.
1.2 Motivación	
.....	¡Error! Marcador no definido.
1.3 Estructura del Trabajo.....	2
1.4 Contribuciones de la Tesis.....	3
2.1 Sistemas de Información Geográfica	4
2.2 Introducción al Geoprocesamiento.....	4
2.3 ¿Qué es un SIG?.....	5
2.3.1 Características de un SIG.....	6
2.4 Bases de Datos Espaciales.....	9
2.4.1 Datos espaciales.....	10
2.4.2 Representación de datos espaciales.....	11
2.4.3 Relaciones entre objetos espaciales.....	12
2.4.4 Captura de la Información.....	12
2.4.4.1 Formato RASTER.....	13
2.4.4.2 Formato VECTORIAL.....	14
2.4.4.3 Ventajas y Desventajas del Formato RASTER y VECTORIAL..	16
3.1 Posicionamiento.....	17
3.2 Sistemas de Referencia Espacia.....	18
3.2.1 Sistemas de Coordenadas.....	19
3.2.1.1 Sistemas de Coordenadas Geográficas.....	19
3.2.1.2 Sistemas de Coordenadas Geocéntricas o Cartesianas.....	20
3.2.1.3 Sistemas de Coordenadas Planares.....	21
3.2.2 DATUM.....	22

3.2.3	Proyecciones Cartográficas.....	24
3.2.3.1	Sistema Universal Transversal Mercator.....	26
3.3	Localización Espacial.....	26
3.3.1	Servicios Basados en Posiciones.....	27
4.1	Sistemas de Posicionamiento.....	28
4.2	Clasificación de Sistemas de Posicionamiento.....	29
4.2.1	Sistema de Posicionamiento Satelital.....	29
4.2.1.1	Sistema de Posicionamiento Global NAVSTAR.....	30
4.2.1.2	Sistemas de Posicionamiento GLONASS.....	32
4.2.1.3	Sistemas de Posicionamiento GALILEO.....	32
4.2.1.4	Método de Triangulación.....	33
4.2.2	Sistema de posicionamiento Indoor.....	34
5.1	Arquitectura en Capas.....	38
5.2	Análisis del Modelo.....	39
5.2.1	Capa Geográfica.....	41
5.2.2	Capa de Movilidad.....	44
5.3	Movilidades.....	55
5.3.1	Movilidad Transitar Áreas Interés.....	56
5.3.2	Movilidad Transitar Áreas Lógicas	57
5.3.3	Movilidad Red de Circulación Interna.....	60
5.3.4	Movilidad Red de Calles.....	63
5.4	Reposicionamiento del Usuario.....	66
6.1	Casos de Uso.....	68
6.2	Diagramas de Secuencia.....	72
6.3	Diagrama de Clase.....	91
7.1	Conclusiones.....	94
7.2	Trabajos Futuros.....	95

Bibliografía	96
Anexo.....	99

Agradecimientos

A nuestra directora, la Dra. Silvia Gordillo, por habernos orientado en la elección del tema.

A la Lic. Alejandra Literas por su paciencia y dedicación.

A nuestras familias por el apoyo que nos han brindado todos estos años.

A nuestros amigos y compañeros por las experiencias compartidas.

A cada una de nosotras por haber logrado concluir este desafío.

Claudia Luján Ilarragorri 1698/5

Leticia Beatriz Durante 3157/7

Capítulo 1

En este capítulo se hace una pequeña introducción al desarrollo del presente trabajo, describiendo brevemente los temas abordados en los distintos capítulos y se mencionan las contribuciones del mismo.

1.1 Introducción

Las aplicaciones de Hipermedia Física (HF) extienden el paradigma de Hipermedia, brindando soporte para que los usuarios con dispositivos móviles puedan explorar tanto digitalmente como físicamente los objetos de interés.

En este paradigma se combinan los objetos digitales que se definen y utilizan como en las aplicaciones de Hipermedia convencionales, con objetos físicos frente a los cuales el usuario se posiciona (ya sea por su movilidad o por movilidad del objeto). De esta manera, cuando el usuario se encuentra ubicado físicamente frente a un objeto, el sistema es capaz de detectarlo y entonces puede proveer información adicional sobre el mismo.

1.2 Motivación

Vivimos en una época donde cada vez es más común que los usuarios tengan acceso a tecnologías de última generación y donde el avance tecnológico parece no tener límite. Como consecuencia del incremento en el uso de aplicaciones móviles y gracias a la expansión de las redes de telefonía celular, es que las empresas buscan innovar en los servicios que ofrecen para aumentar su competitividad.

Hoy en día existen numerosas aplicaciones basadas en posiciones que integran Sistemas de Información Geográfica (SIG) con sistemas de posicionamiento. Como por ejemplo

aplicaciones de tráfico en tiempo real, sistemas de seguimiento vehicular, “buscadores de amigos”, que permiten a los usuarios de dispositivos móviles disponer de aplicaciones que les brinden mayores y mejores servicios.

En este contexto vale la pena analizar la problemática descrita en [1], que plantea una solución a distintas situaciones que son interesantes para incorporarlas como servicios en aplicaciones de Hipermedia Física (HF). Algunos de estos escenarios son:

- Recalcular caminos cuando la persona se pierde o se desvía del camino propuesto.
- Tener caminos que involucren tramos que por diversos motivos no están habilitados para el usuario:
- Recorridos internos en edificio o parte del mismo
- Recorridos que involucran salir de un edificio.

Nuestro trabajo consiste en desarrollar una posible implementación que se desprende del modelo propuesto en [1]. Dicho modelo está dividido en dos capas, por un lado la capa Geográfica y por el otro la capa de Movilidad. Esta última establece la movilidad entre distintos puntos de interés (PI) de usuarios móviles.

Asumiendo que disponemos de la funcionalidad de la capa Geográfica que permite el georeferenciamiento de entidades conceptuales mediante el concepto de “Location”, centraremos nuestro trabajo en analizar e implementar la funcionalidad de la capa de Movilidad mencionada.

1.3 Estructura del trabajo

En el **capítulo 2** se hace una introducción al concepto de Sistemas de Información Geográfica, sus diferentes visiones y la forma de representación de la información.

En el **capítulo 3** se da una visión general de lo que es “Posicionamiento” y su utilización en los Servicios Basados en Posición (LBS). Junto con el concepto de Posición espacial, se presentan las diferentes maneras de expresar una posición espacial en un sistema de coordenadas y se explica el término DATUM. Además, se da una introducción a la proyección de la superficie terrestre en mapas.

En el **capítulo 4** se describen diferentes sistemas de Posicionamiento Indoor y Outdoor que se usan hoy en día para determinar la ubicación de un vehículo o una persona.

También se realiza una introducción al concepto de Movilidad.

En el **capítulo 5** se analiza y explica detalladamente la funcionalidad del modelo tomado como base, junto con las modificaciones que éste ha sufrido a la largo del desarrollo.

En el **capítulo 6** se explica la implementación desarrollada, mediante la utilización de diagramas de casos de uso y diagramas de Secuencia.

En el **capítulo 7** se establecen las conclusiones y trabajos futuros.

En el **anexo** se plantean ejemplos de la ejecución del código para las distintas movilidades que abarca el modelo, se expone un ejemplo que involucra el paso de un escenario Indoor a un escenario Outdoor y viceversa. Además se plantea un ejemplo que implica cambiar de movilidad dentro de un mismo escenario.

1.4 Contribuciones de la Tesis

El presente trabajo realiza el análisis exhaustivo del modelo Orientado a Objetos propuesto en [1], que establece una arquitectura modular en capas, en la que se asume definida la capa Geográfica. Sobre esta capa se define una capa de Movilidad que plantea la integración de escenarios Indoor y Outdoor, la cual constituye el objetivo de nuestro análisis.

Se propone la funcionalidad básica de la capa de Movilidad mencionada como forma de mostrar la sustentabilidad del modelo y su incorporación en aplicaciones de Hipermedia Física.

Se desarrollan algoritmos de búsquedas de caminos entre distintos puntos de interés, como implementación concreta de un servicio de asistencia al usuario, planteando recorridos que involucran escenarios Indoor, Outdoor o Indoor-Outdoor.

Capítulo 2

En este capítulo haremos una introducción al concepto de Sistema de Información Geográfica. Veremos las diferentes visiones que hay sobre los mismos y como se representa la información en ellos.

2.1 Sistemas de Información Geográfica

En los últimos años los Sistemas de Información Geográfica constituyen la herramienta metodológica adecuada para el análisis temporal, como por ejemplo dinámica de los usos del suelo, espacial y en las tareas de planificación ambiental y ordenación del territorio [22].

2.2 Introducción al Geoprocesamiento

El geoprocesamiento puede ser definido como el conjunto de tecnologías orientadas a la recopilación y tratamiento de información espacial (este término se ampliará en la sección 2.4.1) con un objetivo específico. Así, las actividades que envuelven el geoprocesamiento, son ejecutadas por sistemas específicos para cada aplicación. Estos sistemas son comúnmente llamados Sistemas de Información Geográfica (SIG).

Un sistema de geoprocesamiento está destinado al procesamiento de datos referenciados geográficamente o georreferenciados, desde su recopilación hasta la creación de salidas en forma de mapas convencionales, informes, archivos digitales, entre otras, debiendo poseer recursos para su almacenamiento, gerenciamiento, manipulación y análisis.

Con la evolución de la tecnología de geoprocesamiento y de softwares gráficos, varios términos surgieron para las diferentes especialidades. El nombre Sistemas de Información Geográfica o, en inglés, Geographic Information System (GIS) es muy utilizado y en muchos casos es confundido con geoprocesamiento. El geoprocesamiento es un concepto más abarcador y representa cualquier tipo de procesamiento de datos georreferenciados,

mientras que un SIG procesa datos gráficos y no gráficos, alfanuméricos, con énfasis en análisis espaciales y modelados de superficies [29].

2.3 ¿Qué es un SIG?

Un sistema de información geográfica puede ser descrito de varias formas. A través de su evolución surgieron distintas definiciones, algunas de las cuales citamos a continuación:

- Sistemas basados en computadoras usados para almacenar y manipular datos geográficamente referenciados [5].
- Un SIG es un conjunto de herramientas para almacenamiento, recuperación, transformación y la representación de datos espaciales relativos al mundo real [24].
- Un Sistema de Información Geográfica es un tipo especializado de base de datos, que se caracteriza por su capacidad de manejar datos geográficos, es decir, espacialmente referenciados, los cuales se pueden representar gráficamente como imágenes [25].
- Un conjunto de tecnologías de la información, datos y procedimientos, para recolección, almacenamiento, manipulación, análisis, y descripción de características que pueden ser representadas sobre mapas [26].
- Un Sistema de Información Geográfico (SIG), es un sistema computacional capaz de almacenar, recuperar, integrar, manipular y visualizar diferentes informaciones de la superficie terrestre. Es utilizado en estudios relacionados al medio ambiente, recursos naturales, investigación de la previsión de determinados fenómenos o en el apoyo a las decisiones de planificación, considerado el hecho que los datos almacenados representen un modelo del mundo real [24].
- Un Sistema de Información Geográfica se puede contemplar como un conjunto de mapas de la misma porción del territorio, donde un lugar concreto tiene la misma localización (las mismas coordenadas) en todos los mapas incluidos en el sistema de información. De este modo, resulta posible realizar análisis de sus características espaciales y temáticas para obtener un mejor conocimiento de esa zona [27].

Todas las definiciones de SIG de los diferentes autores que se han citado, tienen una característica en común: tratan con información geográfica (datos geográficos). Y se diferencian en la manera en la cual ven al SIG, algunos lo ven como un conjunto de mapas, otros como una base de datos y otros como un sistema para realizar análisis espacial. Estas diferentes maneras de ver a un SIG se ampliarán en la sección 2.3.1

2.3.1 Características de un SIG

El término Sistemas de Información Geográfica es aplicado para sistemas que realizan el tratamiento computacional de datos geográficos. Debido a su amplia gama de aplicaciones, que incluye temas como agricultura, bosques, cartografía, catastro urbano, redes de distribución, como por ejemplo, agua, energía, telefonía, entre otras, existen por lo menos tres grandes maneras de utilizar un SIG [29]:

- como una base de datos geográfica, con funciones de almacenamiento y recuperación de información espacial;
- como herramienta para la producción de mapas;
- como soporte para el análisis espacial;

A continuación explicaremos cada una de estas visiones.

1. Desde el punto de vista de Base de Datos Geográfica

Un SIG funciona como una base de datos con información geográfica, datos alfanuméricos, que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital. De esta forma, señalando un objeto, tanto del espacio como de la Tierra, se conocen sus atributos e, inversamente, preguntando por un registro de la base de datos se puede saber su localización en el mapa digital. Es decir, una base de datos espacial, basada principalmente en una base de datos estructurada que describe al mundo en términos geográficos, como se muestra en la Figura 2.3.1.



Figura 2.3.1 - Ejemplo de información espacial

En la Figura 2.3.1 se ve un ejemplo de cómo se almacena información espacial, en este caso se registran calles, en forma de direcciones, que pertenecen a una ciudad. La información espacial que se almacenará es: nombre de la calle, número de la calle, nombre de la ciudad, y demás datos que permitan ubicar un lugar de forma unívoca [13].

2. Desde el punto de vista de una herramienta para la producción de mapas

Esta visión, se centra en los aspectos de la cartografía manipulada por un SIG. Los defensores de esta visión ven a los SIG como sistemas de visualización o procesamiento de mapas. En el procesamiento de mapas, cada conjunto de información se representa como un mapa o layer ¹ de cobertura. La Figura 2.3.2 muestra un conjunto de layers de cobertura. Los mapas se manejan usualmente en formato raster o vectorial, (este tema se ampliará en la sección 2.4.4), constituyendo lo que se conoce como mapas digitales.

La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas o Layers y las almacena independientemente, permitiendo trabajar con ellas de manera rápida y sencilla, facilitando la posibilidad de relacionar la información existente a través de la topología² de los objetos.

¹ Capas temáticas (Layers): un layer o capa es donde se grafica sin afectar otras capas.

² Topología: En el contexto de los SIG, topología hace referencia a las propiedades de vecindad o contigüidad, inclusión, conectividad y proximidad es decir, propiedades no métricas y que permanecen invariables ante cambios morfológicos, de escala o de proyección.

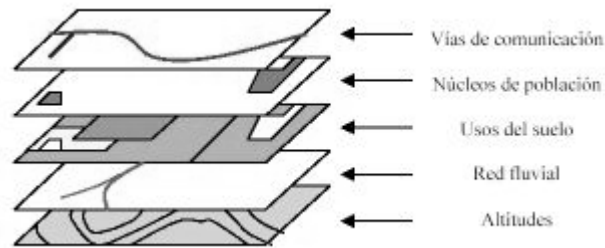


Figura. 2.3.2. Información del mundo real realizado en layers

Puesto que el conjunto de datos geográficos en un SIG está georeferenciado, las posiciones del mundo real se superponen entre si como se ve en la Figura 2.3.2.

Una definición precisa del conjunto de datos geográficos es fundamental para los sistemas de información geográfica, y la definición de las distintas capas temáticas de la información es fundamental para los conjuntos de datos SIG [12].

Esta visión incluye mapas interactivos y otras vistas que operan sobre un conjunto de datos geográficos. Los mapas interactivos proveen la principal interfase de usuario para la mayoría de las aplicaciones SIG, disponibles desde dispositivos móviles, mapas Web o aplicaciones de escritorio. [13]. Este concepto se muestra en la Figura 2.3.3.



Figura. 2.3.3. Distintas interfaces de aplicaciones SIG

En la Figura 2.3.3 se visualizan ejemplos de distintas interfaces donde se muestran SIG en forma de mapa.

3. Desde el punto de vista del análisis espacial

El análisis espacial hace referencia a las herramientas y procesos usados para generar conjunto de datos derivados de bases de datos existentes. En otras palabras, combinando

datos y aplicando algunas reglas analíticas, se puede crear un modelo para resolver distintas problemáticas.

Esta visión es la usada para diferenciar a los SIG del resto de los sistemas de información. [13].

2.4 Bases de datos Espaciales

Una base de datos espacial o geográfica, es una base de datos que maneja información existente en el espacio o datos espaciales (este concepto se ampliará en la sección 2.4.1). El espacio proporciona un marco de referencia para definir la localización y relación entre objetos. La recolección de datos espaciales implica un proceso de abstracción del mundo real. Este proceso tiene diversos niveles o capas que se definen según el propósito de la aplicación.

Un sistema de bases de datos espacial tiene las siguientes características:

a- Un sistema de base de datos espacial es un sistema de base de datos: un administrador de bases de datos espacial (SDBMS) debe proveer todos los servicios de los DBMS tradicionales más capacidades adicionales para manejar datos espaciales.

b- Ofrece tipos de datos espaciales (SDTS), un modelo de datos y lenguaje de consulta:

El principal aspecto de los datos tratados en un SIG es la naturaleza dual de la información, ya que se debe almacenar tanto los objetos geográficos como sus atributos. Un objeto geográfico posee una localización geográfica, expresada como coordenadas en un mapa, y los atributos descriptivos, pueden ser representados en una base de datos convencional.

Una característica básica y general en un SIG es su capacidad de tratar las relaciones espaciales entre los objetos geográficos.

Para cada objeto geográfico, el SIG necesita almacenar sus atributos y las diversas representaciones gráficas asociadas [29].

Los tipos de datos espaciales proveen la abstracción fundamental para modelar la estructura de entidades geométricas en el espacio y sus relaciones.

c- Soporta tipos de datos espaciales y su implementación, con capacidades para indexado espacial y algoritmos eficientes para recuperar la información: el SDBMS por lo menos debe tener capacidades para recuperar objetos desde una colección de datos almacenados sin necesidad de realizar una búsqueda secuencial. Además debería soportar posibilidades para conectar objetos diferentes por alguna relación espacial.

El principal objetivo del indexado espacial es poder soportar la selección espacial o join espacial, esto es, recuperar a partir de un conjunto grande de objetos espaciales aquellos que cumplen con una relación particular [14].

Las bases de datos espaciales no tienen un conjunto de operadores que sirvan como elementos básicos para la evaluación de consultas ya que estas manejan un volumen extremadamente grande de objetos complejos no ordenados en una dimensión [40]. Es por esto que existen algoritmos complejos para evaluar predicados espaciales. Las consultas son realizadas generalmente en SSQL o Spatial SQL, el cual introduce, mediante extensiones, los distintos conceptos del álgebra ROSE [39], dentro del lenguaje SQL estándar, es decir, utiliza las cláusulas SELECT-FROM-WHERE para las tres operaciones en el álgebra relacional (proyección algebraica, producto cartesiano y selección) [40].

2.4.1 Datos Espaciales

El dato espacial, como término, se refiere al dato o información que se puede ubicar en el espacio, ya sea con referencia a un sistema de coordenadas (este concepto se explica en el capítulo 3) o a un orden topológico.

El conjunto de datos espaciales y no espaciales, y los atributos del objeto, constituyen la base de datos, que es el principal componente sobre el que se basan los análisis y los resultados producidos por el SIG, que pueden ser tanto implícitos como explícitos [31].

Un dato espacial es una variable asociada a una posición del espacio. Normalmente se utilizan datos vectoriales, los cuales pueden ser expresados mediante los objetos espaciales punto, línea y polígono, estos términos se definen más adelante.

El término dato espacial, se refiere al dato o información que se puede ubicar en el espacio, sea con referencia a un sistema de coordenadas o a un orden topológico. El conjunto de datos espaciales y no espaciales, los atributos del objeto, constituyen la base de datos, que

es el principal componente sobre el que se basan los análisis y los resultados que pueden ser implícitos como explícitos, que son producidos con el SIG [31].

A todo objeto se asocian atributos que pueden ser:

- Gráficos
- No gráficos o alfanuméricos.

Atributos Gráficos

Son las representaciones de los objetos geográficos asociados con ubicaciones específicas en el mundo real. La representación de los objetos se puede hacer por medio de puntos, líneas o polígonos.

Atributos No Gráficos

También llamados atributos alfanuméricos. Corresponden a las descripciones, o características que nombran y determinan los objetos o elementos geográficos.

2.4.2 Representación de datos espaciales

Los sistemas de bases de datos espaciales, distinguen dos posibilidades de representación de objetos espaciales, de acuerdo a lo que se desee ver [15] [16]:

- a) Objetos Espaciales:** interesa poder distinguir, manejar y consultar entidades geográficas distribuidas en un espacio de representación. Cada una de estas entidades tiene su descripción geométrica propia, que permite modelar por ejemplo ciudades, rutas, ríos.
- b) Espacio:** se describe el espacio en sí mismo, esto es, se quiere decir algo sobre cada punto del espacio. Se usa para describir mapas temáticos como por ejemplo división política.

Los objetos geográficos son identificados por sus dimensiones. Los objetos que ocupan un área son denominados polígonos. Los objetos de una dimensión son denominados líneas. Los objetos de cero dimensiones son denominados puntos.

2.4.3 - Relaciones entre Objetos Espaciales

En lo que respecta a las consultas espaciales y para que éstas brinden la flexibilidad adecuada es necesaria una representación apropiada para los datos espaciales. Entre las operaciones ofrecidas por las álgebras espaciales, las más importantes son las *relaciones espaciales*. Las consultas espaciales pueden clasificarse en:

- **Relaciones topológicas**, como adyacencia, contenido, disjunto. Estas relaciones son invariantes bajo transformaciones topológicas como por ejemplo: traslación, escalado y notación.
- **Relaciones de dirección**, como por ejemplo arriba, abajo, al norte de.
- **Relaciones métricas**, como por ejemplo distancia.

De las relaciones anteriores, aquellas de mayor interés son las relaciones topológicas fundamentales para la mayoría de las aplicaciones SIG, ya que estas establecen propiedades de vecindad o contigüidad, inclusión, conectividad y proximidad de los objetos.

2.4.4 Captura de la información

Los atributos gráficos son guardados en archivos y manejados por el software de un sistema SIG. Los objetos geográficos son organizados por temas de información, o capas de información, llamadas también niveles. Aunque los puntos, líneas y polígonos pueden ser almacenados en niveles separados, lo que permite la agrupación de la información en temas son los atributos no gráficos. Los elementos simplemente son agrupados por lo que ellos representan. Así por ejemplo, ríos y carreteras aunque sean ambos objetos línea se almacenan en distintos niveles debido a que sus atributos son diferentes.

La información geográfica con la cual se trabaja en los SIG puede encontrarse en dos tipos de presentaciones o formatos: Celular o Raster y Vectorial. [17]. La Figura 2.4.4 muestra estos tipos de formatos.

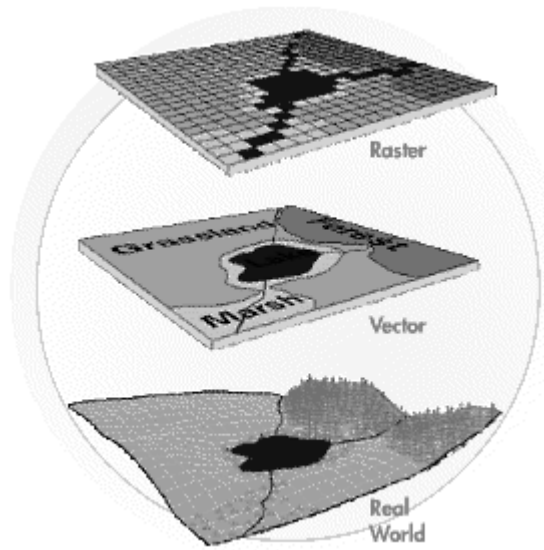


Figura 2.4.4 - Formato de información geográfica

En la Figura 2.4.4 se muestra como se representa la información del mundo real en los formatos Raster y Vectorial.

A continuación se explican cada uno de estos formatos.

2.4.4.1 Formato RASTER

El modelo raster es un método para el almacenamiento, el procesado y la visualización de datos geográficos. Cada superficie a representar se divide en filas y columnas, formando una matriz regular. Cada celda o píxeles de la matriz es rectangular, y guarda tanto las coordenadas de la localización como el valor temático. La localización de cada celda es implícita, dependiendo directamente del orden que ocupa en la matriz. En el modelo raster el espacio no es continuo sino que se divide en unidades discretas. Esto hace a este formato especialmente indicado para ciertas operaciones espaciales como por ejemplo las superposiciones de mapas o el cálculo de superficies [5].

Con la rasterización se asignan códigos a las celdas según tres tipos de muestreo:

- Modal (lo más representativo en la celda)
- Punto medio: (lo que represente el punto central de celda)
- Lógico (si una entidad en cuestión aparece o no).

En la Figura 2.4.4.1 se muestra un ejemplo de rasterización.

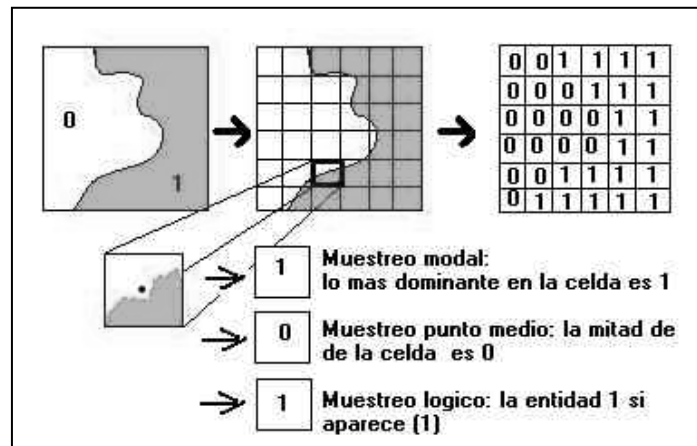


Figura 2.4.4.1 – Ejemplo de formato RASTER

En la Figura 2.4.4.1 se pueden observar los distintos tipos de muestreos para la codificación raster [23].

La representación de los elementos del mundo real se hace de la siguiente forma: un elemento puntual se representa mediante una celda, un elemento lineal mediante una secuencia de celdas alineadas y un elemento poligonal mediante una agrupación de celdas contiguas. Los límites entre elementos geográficos no quedan registrados de forma explícita, pero se pueden deducir a partir de los valores de las celdas.

Una capa constituye un conjunto de celdas y sus valores asociados. Dado que en cada celda se registra un único valor, si se quiere almacenar información relativa a distintas variables, se han de incluir tantas capas como variables se consideren, por ejemplo, altitud, precipitación, uso del suelo, entre otras.

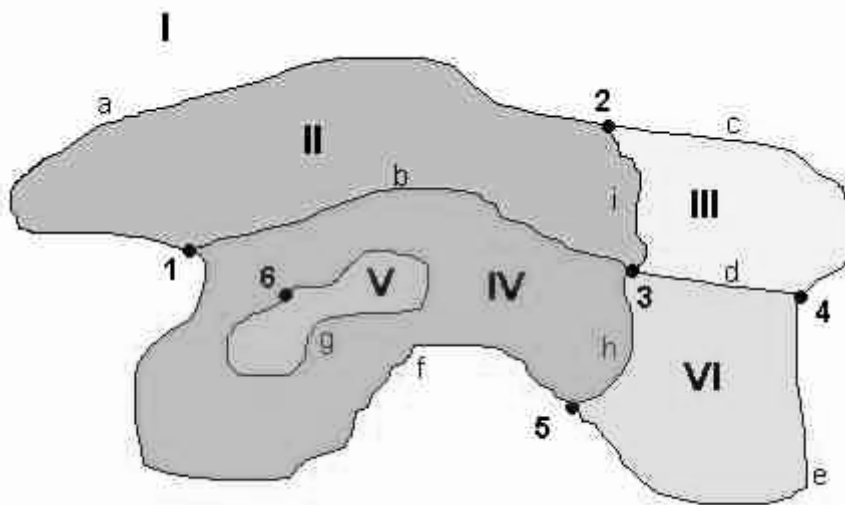
La captura de la información en este formato se hace mediante los siguientes medios: scanners, imágenes de satélite, fotografía aérea, cámaras de video, entre otros. En todos los casos se obtiene un archivo digital de esa información [5].

2.4.4.2 Formato VECTORIAL

Al contrario de lo que ocurre con el formato raster, el formato vectorial define objetos geométricos mediante la codificación explícita de sus coordenadas, estableciendo además las relaciones topológicas entre las mismas.

Una forma de representación vectorial es la estructura Arco-Nodo, en la cual el sistema puede identificar relaciones como la inclusión, adyacencia, entre otras, gracias a un conjunto de tablas topológicas, una para polígonos, una para arcos y finalmente una para nodos. Si los polígonos comparten un mismo arco entonces hay ADYACENCIA de polígonos. Si hay un nodo final o inicial repetido en dos arcos entonces hay CONECTIVIDAD de arcos. Si hay un registro negativo significa que hay INCLUSIÓN de una figura en otra [5].

En la Figura 2.4.4.2 se muestra un ejemplo de un modelo vectorial.



Topología de arcos						Topología de polígonos		Topología de nodos			
ARCO	Nodo inicial	Nodo final	Coord. vértices x,y...x,y	Polígono Izq.	Polígono der.	Polígono	Arcos	Nodo	arcos	Coord x	Coord y
a	1	2		I	II	III	i, c, d	1	a, b, f		
b	1	3		II	IV	IV	b, h, f, -g	3	b, i, d, h		
i	2	3		III	II	V	g	4	c, d, e		
g		6		V, IV		VI	d, e, h	6	g		
:	:	:	:	:	:	:	:	:	:	:	:

Figura 2.4.4.2 – Ejemplo del formato VECTORIAL

En la Figura 2.4.4.2, se observa como se representan las relaciones en las tablas topológicas. Por ejemplo, observando la tabla de polígonos, se puede determinar que los polígonos III y VI son adyacentes ya que comparten el arco denominado **d**, observando la tabla de arcos podemos determinar que el arco **a** y el arco **i** están conectados por medio del nodo llamado **2** [23].

En este formato, los puntos se codifican por un par de coordenadas en el espacio, las líneas como una sucesión de puntos conectados y los polígonos como líneas cerradas o como un conjunto de líneas que constituyen las diferentes fronteras del polígono.

La captura de la información en el formato vectorial se hace por medio de mesas digitalizadoras, convertidores de formato raster a formato vectorial, sistemas de geoposicionamiento global (GPS), entrada de datos alfanumérica, entre otros [5].

2.4.4.3 - Ventajas y desventajas del formato RASTER y VECTORIAL

A continuación se listan las ventajas y desventajas de los formatos Raster y Vectorial [5].

Ventajas del formato Raster:

- Estructura de datos simple.
- Representación eficiente de alta variabilidad espacial.
- Manipulación eficiente de imágenes digitales.
- Fácil implementación de operaciones de superposición.

Desventajas del formato Raster:

- Estructura de datos menos compacta.
- Dificultad para representar relaciones topológicas.
- La salida gráfica es menos precisa debido a que las fronteras tienen una apariencia de bloque.

Ventajas del formato Vectorial

- Estructura de datos más compacta.
- Codificación eficiente de Topología, por lo tanto implementación más eficiente de operaciones que requieran información Topológica.
- Más apropiado para el soporte de gráficos.

Desventajas del formato Vectorial

- Estructura de datos más compleja.
- Operaciones de superposición difíciles de implantar.
- Representación ineficiente de una alta variabilidad espacial.
- Manipulación ineficiente de imágenes digitales.

Capítulo 3

En este capítulo se da una visión general de lo que es “Posicionamiento”, se presentan las diferentes maneras de expresar una posición espacial en un sistema de coordenadas y se explica el termino DATUM que se utiliza para el modelado de la superficie de la Tierra. También se explica brevemente la proyección de la superficie terrestre en mapas. Se realiza una introducción al concepto de Localización Espacial y su utilización en los Servicios Basados en Posición (LBS).

3.1 Posicionamiento

Hoy en día, determinar la posición del usuario es de gran importancia, ya que, no solo permite la concepción de nuevos servicios, como por ejemplo, Sistemas de Localización Automática de Vehículos (AVL), sino que también tiene el potencial de hacer que los servicios de Internet móvil sean de mayor utilidad para los usuarios en la medida que la información se ajuste al contexto, como por ejemplo brindar información climática de la región en que se encuentra la persona.

Básicamente el término “Posición” esta asociado a un lugar específico en el mundo, el cual debe poder ser referenciado inequívocamente [4]. Por ejemplo: Cuando dos personas coordinan encontrarse deben establecer un lugar para hacerlo, como puede ser un restaurante o una oficina, este lugar debe ser identificado por ambas personas de manera que concurran al mismo sitio.

Mediante la Georreferenciación se puede determinar un lugar en forma unívoca. Se llama Georreferenciación al posicionamiento en el que se define la localización de un objeto

espacial en un sistema de coordenadas y DATUM determinado establecido en el sistema de referencia.

A continuación se explican estos conceptos.

3.2 Sistema de Referencia Espacial

Un sistema de referencia espacial consta de los siguientes elementos:

- un sistema de coordenadas
- un DATUM

Un sistema de referencia es un conjunto definido de un origen y de tres ejes perpendiculares que pasan por él, y que permite la localización de un punto sobre la superficie terrestre, como se observa en la Figura 3.2.2.

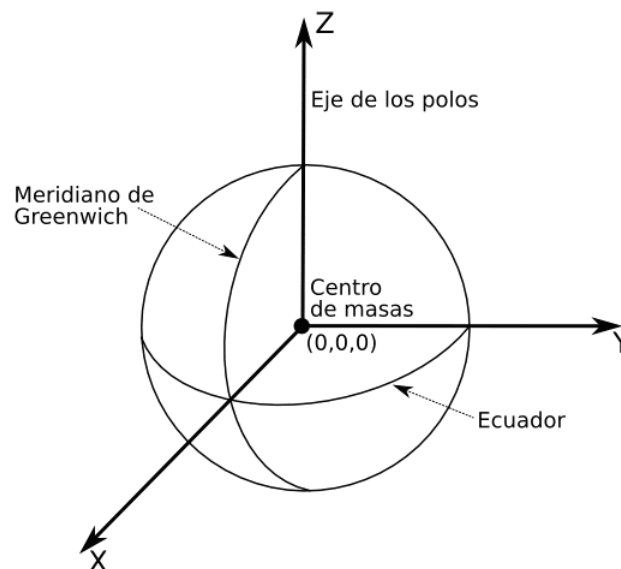


Figura 3.2.2 - Sistema de Referencia

Estos tres ejes componen una terna cartesiana geocéntrica (X, Y, Z), en la cual el eje X es la intersección del plano del Ecuador con el meridiano de Greenwich, el eje Y está ubicado también sobre el plano del Ecuador pero a 90° del eje X hacia el Este y el eje Z completa la terna sobre el eje de rotación terrestre.

3.2.1 Sistema de Coordenadas

Una coordenada es una tupla de valores que permiten definir inequívocamente la posición de cualquier punto en el espacio o en el plano. Es usada para referenciar una determinada ubicación espacial por medio de un vector de números [4]. Las coordenadas de un objeto se refieren a su posición, ya sea, medida por la distancia o por el ángulo respecto a dos o tres ejes del sistema de coordenadas, dependiendo de si la posición se fija sobre un plano o sobre el espacio.

Existen diferentes tipos de Sistemas de Coordenadas, a continuación explicaremos algunos de ellos:

- Sistemas de Coordenadas Geográficas
- Sistemas de Coordenadas Geocéntricas o Cartesianas
- Sistemas de Coordenadas Planares

3.2.1.1 Sistemas de Coordenadas Geográficas

Constituye el sistema más poderoso ya que permite una alta resolución espacial y análisis espacial incluyendo el cálculo de distancias entre puntos ubicados sobre la curvatura de la Tierra.

La manera mas conveniente de expresar una ubicación espacial es usando un sistema de coordenadas geográficas que modela la superficie de la Tierra como un elipsoide. Un elipsoide describe la forma de la Tierra mediante la fijación de un radio ecuatorial y polar. El origen en un sistema de referencia geográfico es dado por dos planos de referencia, ambos ortogonales entre si y que cruzan por el centro de la Tierra.

Una posición sobre la superficie terrestre es expresada en latitud, longitud.

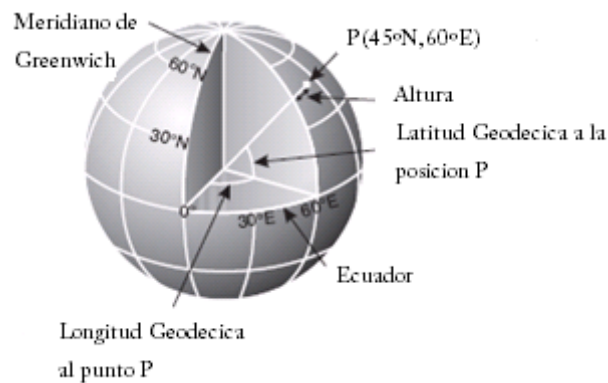


Figura 3.2.2.1.1 Latitud, longitud y altura

Como se muestra en la Figura 3.2.2.1.1, las líneas de latitud corren paralelamente al Ecuador y se denominan paralelos, mientras que las líneas de longitud, son perpendiculares a la línea del Ecuador, y son llamadas meridianos. El concepto de latitud/longitud permite expresar una posición situada sobre la superficie de la Tierra. Sin embargo, para muchas aplicaciones, tales como la aviación, también es importante saber la altura o la altitud de una posición, representada por el eje **Z**, respecto a la superficie de la Tierra.

La altura geodésica se utiliza para expresar la altitud, y se define como la distancia entre una posición y la referencia elíptica a lo largo de la línea perpendicular al elipsoide.

Si bien la altitud puede ser expresada por cualquier unidad de longitud, como metros o pies, la unidad de latitud y longitud se da en grados ($^{\circ}$). La latitud tiene un valor en el rango de -90° a $+90^{\circ}$ (o $N90^{\circ}$ a $S90^{\circ}$). La longitud varía entre -180° y $+180^{\circ}$ (o $E180^{\circ}$ a $O180^{\circ}$).

Debido a que las líneas de latitud se organizan como paralelos, tienen siempre una distancia constante a lo largo de la superficie de la Tierra, en cambio, las líneas de longitud convergen en los polos, por lo cual, la distancia entre ellas depende de la latitud [4].

3.2.1.2 Sistema de Coordenadas Geocéntricas o Cartesianas

Los Sistemas de Coordenadas Geocéntricas o Cartesianas son los más utilizados en los cálculos matemáticos. Se usan para expresar la ubicación espacial mediante la especificación de sus distancias a los ejes. Los ejes son etiquetados como X e Y en un

sistema de coordenadas bidimensional y como X, Y, y Z en sistemas de coordenadas tridimensional y están dispuestos entre si en forma ortogonal. Cada par de ejes define un plano, que es etiquetado de acuerdo a como los ejes se ubican en el mismo, por ejemplo, en un plano formado por los ejes X e Y, donde X es el eje vertical e Y es el eje horizontal, el plano se denomina plano XY. Los ejes se intersecan en el origen, el cual en un sistema de coordenadas bidimensional es por lo general (0, 0) y en un sistema de coordenadas tridimensional es por lo general (0, 0, 0).

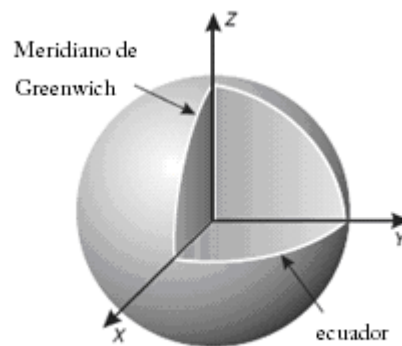


Figura 3.2.2.1.2 Centro de la Tierra, Tierra fija X,Y,Z

Como se muestra en la Figura 3.2.2.1.2, el eje Z se corresponde con el eje de rotación de la Tierra y, por lo tanto, es definido por la línea que va del centro de la Tierra hacia el Polo Norte. El eje X se cruza con el ecuador en un punto que se conoce como el meridiano de Greenwich. El eje Y se define en relación con el eje X en el sentido de que completa un sistema ortogonal. El plano XY es llamado también plano ecuatorial [4].

3.2.1.3 Sistema de Coordenadas Planares

Las coordenadas planares describen la distancia al origen (0,0) a lo largo de dos ejes separados, un eje horizontal X, representando la dirección Este – Oeste y un eje vertical Y representando la dirección Norte – Sur.

Debido a que la Tierra es un geoide y los mapas son planos, obtener información de la superficie curva a la plana requiere una fórmula matemática llamada proyección cartografía [32], (este término se explica en la sección 3.2.3).

3.2.2 DATUM

Durante mucho tiempo, la humanidad creyó que la Tierra era perfecta, de forma regular, hipótesis que fue postulada por el matemático y filósofo griego Pitágoras alrededor de 500 años AC. Por el año 1687, el científico Inglés Sir Isaac Newton llegó a la conclusión, después de extensas investigaciones, que los planetas deben tener la forma de un elipsoide achatado. Argumentó que los planetas serian esferas perfectas solo si no rotaran [4].

Debido a que la Tierra no es esférica, ni siquiera es un cuerpo regular achatado en los polos, es que cada país, o incluso cada región, debe elegir un modelo de cuerpo, definible matemáticamente, que más se ajuste a la forma de la tierra en su territorio. Generalmente este cuerpo suele ser un elipsoide.

Es por ello que los estudios de geodesia han creado muchos modelos de aproximación de la forma de la Tierra, que se conoce comúnmente como DATUM geodésico [4].

Un DATUM define el tamaño y la forma de la Tierra, así como el origen y la orientación del sistema de coordenadas que se utiliza para referenciar a una determinada posición [4].

Un DATUM se clasifica de la siguiente manera:

- **DATUM Horizontal:** Estos DATUM aproximan a la Tierra con la forma de un elipsoide que define el origen y la orientación del sistema de coordenadas. Los Datums horizontales son la condición esencial para referenciar una posición con respecto a la superficie de la Tierra y son también la base para la proyección de la superficie de la Tierra sobre un mapa.
- **DATUM Vertical:** Se basan en la determinación de un geoide, el cual reproduce el nivel medio del mar, con más o menos precisión. El DATUM Vertical sirve principalmente para determinar la altura de una cierta posición con respecto al nivel medio del mar, para lo cual el geoide actúa como un punto de referencia para el origen y la orientación.

Los diferentes elipsoides se definen matemáticamente de acuerdo a los siguientes parámetros:

- El radio mayor y menor del elipsoide, denominados con las letras a y b.
- El aplastamiento del elipsoide, denominado con la letra f, donde f se calcula con la siguiente formula: $f = a / (a-b)$

Tal como se muestra en la Figura 3.2.2.2.

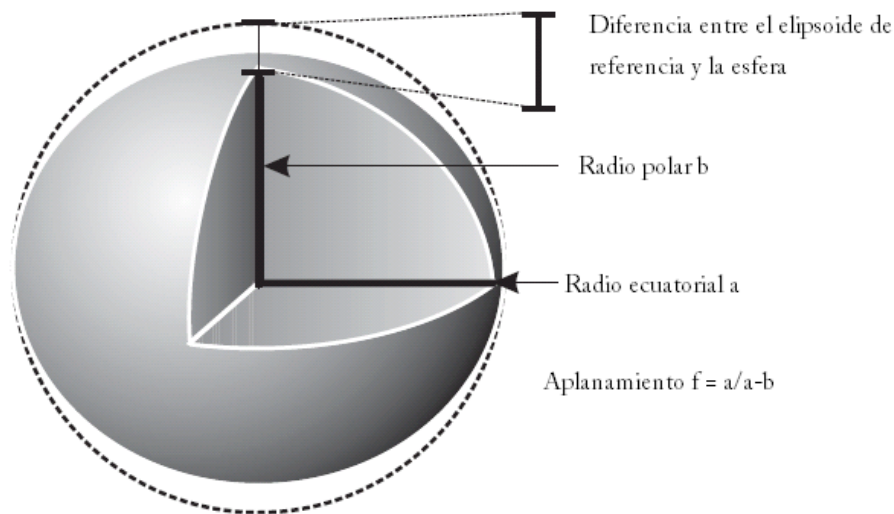


Figura 3.2.2.2 Esfera y elipsoide de referencia

En la Figura 3.2.2.2 se muestra la relación entre el aplanamiento de la forma de la Tierra con respecto a los radios ecuatorial y polar. [4]

Cada DATUM esta compuesto por:

- a) Un elipsoide, definido por el radio polar, el radio ecuatorial y el aplanamiento.
- b) Un punto Fundamental en el que el elipsoide y la tierra son tangentes. Este punto Fundamental se define por sus coordenadas geográficas latitud y longitud. Se trata de un lugar o sitio preciso que se usa como referencia u origen para definir el resto de los puntos de un mapa.

Se ha establecido como estándar internacional el DATUM llamado WGS84 (World Geodetic System), el cual posee las siguientes características:

- El origen 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 o ECEF: Earth) [35].

La selección correcta del DATUM es especialmente crítica cuando utilizemos sistemas de posicionamiento global (GPS), los cuales brindan sus resultados en latitud y longitud y utilizan el WGS84 como sistema de referencia [33].

3.2.3 Proyecciones Cartográficas

Una proyección cartográfica es una representación sistemática de los paralelos y meridianos de una superficie tridimensional en una superficie bidimensional, como se puede observar en la Figura 3.2.3.1.

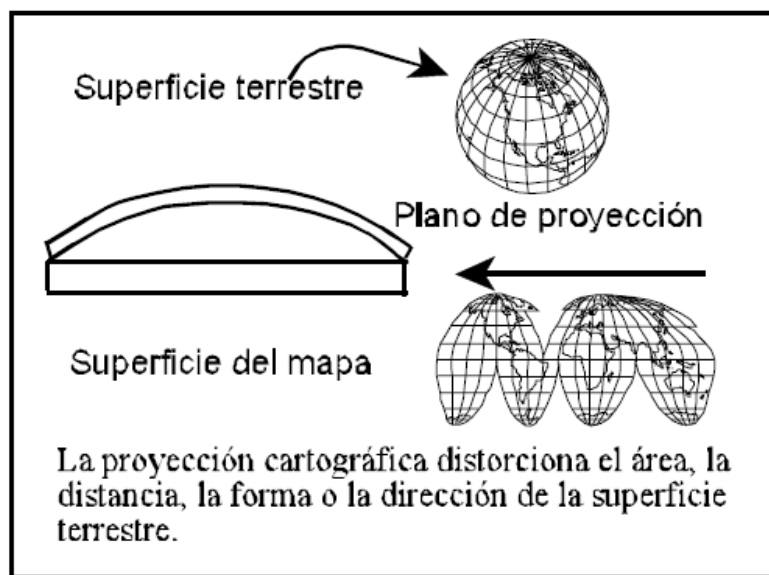


Figura 3.2.3.1 Concepto de proyección Cartográfica

La Figura 3.2.3.1 muestra que, dado que una superficie plana no puede ajustarse a una esfera sin estirarse o encogerse tampoco es posible representar atributos de la esfera terrestre (Ej. meridianos, paralelos, límites entre países) en un mapa sin causar distorsiones. Existen diversas proyecciones y cada una de ellas trata de minimizar las distorsiones. Por ejemplo, el cartógrafo puede diseñar una cuadrícula sobre la superficie terrestre de tal forma que una o más de sus propiedades geométricas se mantengan o de tal forma que las áreas de mayor distorsión se ubiquen en zonas de menor importancia para el uso que se le dará al mapa (Ej. mantener la geometría de los continentes a expensas de la geometría de los océanos) [7].

Las proyecciones que se utilizan en la actualidad se han derivado a partir de modelos matemáticos del globo terrestre y todas ellas comparten la misma característica: mostrar la

posición correcta de las líneas de longitud y latitud del Planeta. En otras palabras, cada proyección es solamente un reordenamiento de los meridianos y paralelos trasladados del Globo Terrestre a un mapa [33].

En la Argentina la proyección más utilizada es la transversa de Mercator, identificada como Gauss Krüger, como se observa en la Figura 3.2.3.2

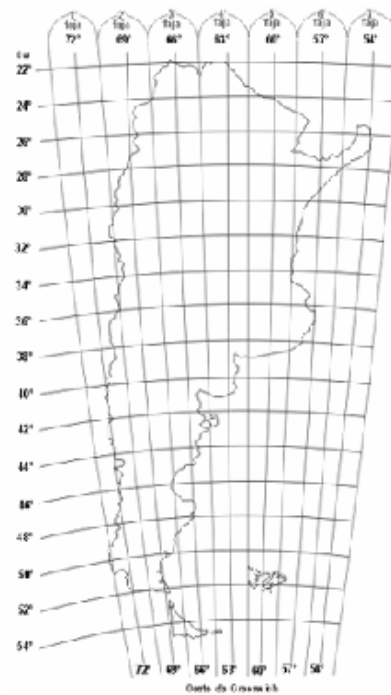


Figura 3.2.3.2 – Proyección Gauss Krüger

La Figura 3.2.3.2, muestra el sistema de proyección Gauss Krüger. Este sistema de proyección, empleado por el Instituto Geográfico Militar para la confección de todas las cartas topográficas nacionales, divide a la República Argentina (sector continental e Islas Malvinas) en 7 fajas meridianas numeradas de Oeste a Este. Cada faja de la grilla Gauss Krüger mide 3° de ancho (longitud) por 34° de largo (latitud) y tiene como propio origen la intersección del Polo Sur con el meridiano central de cada faja. Esta misma proyección es la empleada en el mundo bajo la denominación de UTM que presenta diferencias en su aplicación [7].

3.2.3.1 Sistema Universal Transversal de Mercator (UTM)

Esta no es una proyección en sí misma sino más bien un sistema de coordenadas planas basado en la proyección Transversal de Mercator (ESRI, 1994). Meridianos centrales son establecidos cada 6° de longitud en el ámbito 84° norte y -80° sur. Esto define 60 zonas que se extienden 3° de longitud a ambos lados del meridiano central. Cada zona UTM esta dividida en 20 bandas, designada con una letra del alfabeto de la C a la X etiquetando de sur a norte, como se ve en la Figura 3.2.3.1.1.

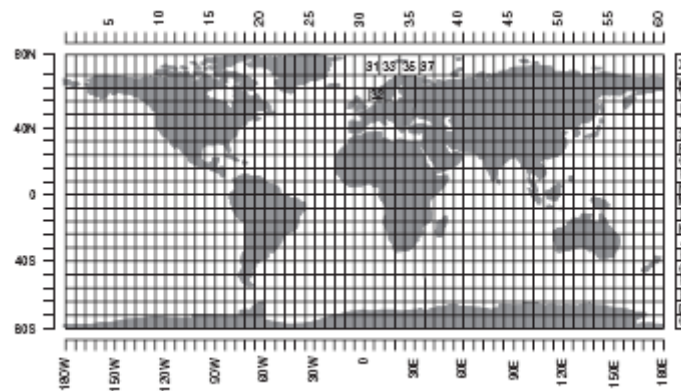


Figura 3.2.3.1.1 Mapa Mundial UTM

Como se observa en la Figura 3.2.3.1.1, UTM es un sistema de coordenadas rectangular, el cual permite obtener con mucha precisión la distancia entre dos puntos de un mapa, asumiendo que estos puntos caen en la misma zona [7].

3.3 Localización Espacial

La ubicación o posición Espacial se basa en un sistema de referencia bien definido que subdivide un área geográfica, en la mayoría de los casos la totalidad de la Tierra, en unidades de forma y tamaño común.

El principal interés sobre el concepto de ubicación espacial aparece para los desarrolladores y usuarios de Servicios basados en Posiciones en forma de coordenadas, un conjunto de valores numéricos, que deben ser interpretados en un contexto geográfico provisto por el

sistema de referencia y que está fuertemente relacionado con la figura de la Tierra, los sistemas de coordenadas y los DATUMs [4].

3.3.1 Servicios basados en Posiciones

Los Servicios basados en posiciones, LBS según su acrónimo en inglés, se refieren a aplicaciones que poseen funciones de manejo de datos espaciales, definidos en el capítulo 2, y a la integración de información georeferenciada con otros tipos de datos [8]. Ejemplos de estas aplicaciones son los Sistemas de Navegación y los Sistemas de Seguimiento Vehicular

El LBS combina dispositivos de hardware, redes de comunicación inalámbrica, información geográfica y programas informáticos, guiando a los usuarios con respecto a una ubicación. El LBS proporciona servicios orientado a aplicaciones específicas, basándose en una posición determinada. Un ejemplo de LBS podría ser un usuario buscando el restaurante más cercano a través de una aplicación WAP ¹ en su teléfono móvil. En este caso la aplicación LBS interactúa con otros componentes de la tecnología de localización para determinar la ubicación del usuario y proporcionar una lista de restaurantes dentro de una cierta proximidad con el usuario móvil [16].

¹ Wireless Application Protocol (WAP) Protocolo para aplicaciones sin hilos, estándar abierto que ofrece aplicaciones móviles avanzadas y acceso a Internet a los usuarios de teléfonos móviles.

Capítulo 4

En este capítulo se describen diferentes sistemas de Posicionamiento Indoor y Outdoor que se usan hoy en día para determinar la ubicación de un vehículo o una persona. También haremos una instrucción al concepto de movilidad.

4.1 Sistemas de Posicionamiento

En la antigüedad, los marineros se orientaban en alta mar gracias a las constelaciones. Luego fue la brújula la encargada de marcar el Norte, y el rumbo. Hoy, en plena era digital, la ayuda sigue viniendo del cielo, pero no de las estrellas, sino de los satélites.

En 1957 se supuso el comienzo de la era de los satélites artificiales y su posterior uso en aplicaciones para el interés de la comunidad mundial, la tecnología ha avanzado rápidamente en este aspecto. Uno de los campos en los cuales se ha manifestado dicho avance es en las aplicaciones que conciernen a las ciencias de la Tierra, y dentro de ellas, la Geodesia, que estudia su forma y dimensiones [19].

El origen de los métodos de posicionamiento se dio en 1978 con el lanzamiento de los primeros satélites de la constelación NAVSTAR (detallado en la sección 4.3.1.1). A principios de los años 80, se empezaron a utilizar estos métodos para aplicaciones civiles, tales como actividades de navegación aérea, marítima y terrestre.

4.2 Clasificación de Sistemas de Posicionamiento

Un sistema de posicionamiento, es un método para identificar y almacenar, generalmente en forma electrónica, la ubicación de un objeto como puede ser un vehículo o una persona. Este sistema puede ser usado para registrar el recorrido de un vehículo a través de la superficie terrestre, en el aire o en el espacio.

Existen distintos sistemas de posicionamiento:

- Sistemas de Posicionamiento Satelital o Espacial.
- Sistemas de Posicionamiento Indoor.

A continuación explicaremos cada uno de estos.

4.2.1 Sistemas de Posicionamiento Satelital

Todo sistema de posicionamiento satelital utiliza un conjunto de satélites artificiales, llamados constelaciones. Existen distintas constelaciones, entre ellas se destacan:

- La constelación NAVSTAR (Navegación por satélite en tiempo y distancia).
- La constelación GLONASS (Sistema Global de navegación por satélite).
- La constelación GALILEO GNSS (Sistema global de navegación satelital europea).

Este tipo de posicionamiento se produce sobre un sistema de referencia cartesiano, que en el caso de usar la constelación americana NAVSTAR corresponde al sistema WGS-84, y en el caso de usar la constelación rusa GLONASS corresponde al sistema PZ-90. En el área de la investigación han sido utilizados para fines científicos, pero las aplicaciones en las cuales estos sistemas han tenido mayor demanda es en la Geodesia ¹ y la Topografía², a partir del descubrimiento de que dichos sistemas de posicionamiento podían aportar las precisiones requeridas para el avance de estas ciencias y su aplicación [13].

A continuación explicaremos brevemente los distintos sistemas de posicionamiento satelitales.

¹ GEODESIA: ciencia que estudia el levantamiento y la representación de la forma y superficie de la Tierra con sus formas naturales y artificiales.

² TOPOGRAFÍA: ciencia que estudia el conjunto de procedimientos para determinar las posiciones de puntos sobre la superficie de la tierra.

4.2.1.1 Sistemas de Posicionamiento Global NAVSTAR

Es un sistema de navegación basado en satélites, creado y operado por el Departamento de Defensa de los Estados Unidos. Como ya se mencionó, comenzado a principios de los '80, este sistema fue declarado completamente operacional el 27 de Abril de 1995. Completamente operacional significa que el sistema puede ser usado para determinar la posición de un receptor las 24 horas del día, en cualquier parte de la Tierra. El sistema fue concebido originalmente como un auxiliar para la navegación de las fuerzas militares de los Estados Unidos, pero hoy en día el GPS sirve también para fines industriales, comerciales y civiles.

El sistema GPS está conformado por tres sectores fundamentales y dependientes entre sí:

- I. Segmento espacial
- II. Segmento de control
- III. Segmento de usuario

I. Segmento espacial

Este sector lo forman los satélites de la constelación NAVSTAR, la cual consiste de una constelación de 24 satélites. Con una órbita de 20200 km de altura (10900 millas) sobre la superficie terrestre, cada satélite orbita la tierra 2 veces al día, o sea una vez cada 12 horas. Los 24 satélites se dividen en 6 órbitas con 4 satélites cada una., como se muestra en la Figura 4.2.1.1.

CONSTELACIÓN DE SATÉLITES GPS

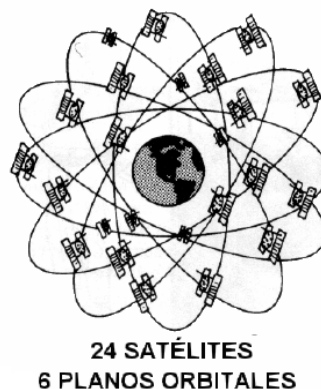


Figura 4.2.1.1: constelación de satélites NAVSTAR.

Esta distribución particular garantiza que por lo menos 4 satélites estarán visibles por un receptor de GPS, por encima de un ángulo de elevación de 15° en cualquier punto de la superficie terrestre, durante las 24 horas del día.

El receptor GPS mide su distancia con los satélites calculando el tiempo que la señal tarda en llegar desde el satélite hasta receptor, y así determina su posición. Con tomar las señales de tres satélites es suficiente para que un aparato GPS pueda triangular las señales y así calcular su posición exacta, este concepto se explicará en la sección 4.4. Sin embargo, los GPS toman señales de cuatro satélites; tres para determinar el lugar y el cuarto, para calcular el tiempo exacto de viaje que tienen estas [13].

Por una cuestión de seguridad, el Departamento de Defensa de Estados Unidos intencionalmente interfiere en la exactitud de los datos proporcionados, como una forma de asegurarse que ninguna fuerza terrorista se valga del GPS. El margen de error que puede tener la información recibida varía entre los 3 y los 15 metros [18].

II. Segmento de Control

Cada satélite GPS lleva a bordo varios relojes atómicos muy precisos. Estos relojes operan en una frecuencia fundamental de 10.23 MHz, la cual se emplea para generar las señales transmitidas por el satélite, para navegación y mapeo.

Los satélites son seguidos y monitoreados por varias estaciones ubicadas estratégicamente alrededor del mundo. Esta red de estaciones de monitoreo se denomina generalmente segmento de control del GPS, y consta de 4 estaciones de monitoreo y una estación de control principal ubicada en EEUU.

Las estaciones de monitoreo miden las señales de ondas de radio que son transmitidas continuamente por los satélites y pasan esa información a la estación de control principal. Ésta usa la información para determinar la órbita exacta de los satélites y para ajustar sus señales de navegación, por ejemplo: error de reloj, correcciones, estado del satélite. [13].

III. Segmento de usuario

Este sector está compuesto por cualquier receptor que reciba las señales GPS, determinando la ubicación del usuario por medio de las señales recibidas. Un aparato GPS consta de una antena, un receptor, el software específico y los sistemas de comunicación [19].

Algunas aplicaciones típicas dentro del segmento Usuario son: la navegación en tierra, ubicación de vehículos, topografía, navegación marítima y aérea, control de maquinaria, entre otras.

4.2.1.2 Sistemas de Posicionamiento GLONASS

GLONASS es un Sistema Global de Navegación por Satélite desarrollado por Rusia. Este sistema está administrado por las Fuerzas Espaciales del Gobierno de la Confederación Rusa y tiene importantes aplicaciones civiles además de las militares. Por motivos de seguridad, las señales de los usuarios civiles ofrecen una precisión dentro de los 100 metros, mientras que las señales de los usuarios militares ofrecen una precisión de 10-20 metros.

Este sistema está formado por una constelación de 24 satélites, 21 satélites activos y 3 satélites de repuesto, situados en tres planos orbitales con 8 satélites cada uno y siguiendo una órbita inclinada de $64,8^\circ$ con un radio de 25510 kilómetros. La constelación GLONASS se mueve en órbita alrededor de la tierra con una altitud de 19.100 kilómetros, un poco más bajo que el GPS, y tarda aproximadamente 11 horas y 15 minutos en completar una órbita. La configuración del sistema GLONASS proporciona datos de navegación a usuarios que se encuentren incluso por encima de los 2000 Km. sobre la superficie terrestre [13].

Actualmente se ha conseguido construir receptores que puedan recibir señales pertenecientes a los dos grupos de satélites, GLONASS y GPS. Esto permite incrementar el número de satélites visibles simultáneamente y por lo tanto las técnicas de posicionamiento serán más rápidas, mejor y más fiables.

4.2.1.3 Sistemas de Posicionamiento GALILEO

En el inicio de la década del 90, la Unión Europea vio la necesidad de que Europa tuviera su propio sistema de navegación por satélite independiente del GPS de Estados Unidos. De esa forma, la Comisión Europea y la Agencia Espacial Europea se unieron para construir GALILEO.

El principio de funcionamiento del sistema europeo será idéntico al GPS norteamericano. Estará formado por 30 satélites geostacionarios distribuidos en tres órbitas circunferenciales situadas aproximadamente a 24.000 Km. de altura sobre la Tierra. De ese total de satélites en órbita se encontrarán siempre operativos 27, mientras los 3 restantes se mantendrán en reserva.

La Agencia Espacial Europea prevé que el sistema Galileo sea mucho más preciso que el GPS teniendo en cuenta la tecnología de los satélites de nueva generación, los sistemas de control que se utilizarán desde Tierra y su carácter civil. De hecho el margen de error se calcula que sea solamente de 4 metros en horizontal y menos de 8 en vertical, prácticamente la décima parte del GPS. Además, gracias a la amplitud territorial que abarcará y la mayor precisión de las señales de los satélites del sistema Galileo, éstas podrán ser captadas también en algunas latitudes remotas hasta donde no llegan todavía las señales del sistema GPS.

Una de sus características es la interoperabilidad con los sistemas GPS y GLONASS, permitiendo que el usuario consiga obtener su localización con el mismo receptor teniendo cualquier combinación de satélites.

Todos estos sistemas descritos anteriormente tienen como objetivo básico proveer la posición de un punto, en cualquier parte del mundo, en cualquier momento y bajo cualquier condición climática. Sin embargo, el uso de esta tecnología va más allá del posicionamiento para fines de mapeo.

4.2.1.4 Método de Triangulación

Todos los sistemas de posicionamiento satelitales utilizan la triangulación como método para determinar una posición.

Para utilizar la fórmula de triangulación, es necesario utilizar como mínimo 3 señales satelitales para ser interpretadas por el receptor GPS. Usando la información que transmiten los satélites al receptor y mediante el cálculo del tiempo que dura una señal en llegar a este, el receptor es capaz de determinar la distancia exacta entre él y el satélite.

Mediante la recolección de 3 ó más señales de satélite en órbitas cuidadosamente monitoreadas, el receptor es capaz de calcular su relación espacial con cada satélite y así obtener una lectura precisa de su ubicación en el planeta. Para comprender mejor este método, planteamos el siguiente ejemplo:

Supongamos que medimos nuestra distancia al primer satélite y resulta ser de 20.000 Km.

Sabiendo que estamos a 20.000 Km. de un satélite determinado, no podemos por lo tanto estar en cualquier punto del universo sino que esto limita nuestra posición a la superficie de una esfera que tiene como centro dicho satélite y cuyo radio es de 20.000 Km.

A continuación medimos nuestra distancia a un segundo satélite y descubrimos que estamos a 21.000 Km. del mismo. Esto nos dice que no estamos solamente en la primer esfera, correspondiente al primer satélite, sino también sobre otra esfera que se encuentra a 21.000 Km. del segundo satélite. Estamos en algún lugar de la circunferencia que resulta de la intersección de las dos esferas.

Si ahora medimos nuestra distancia a un tercer satélite y descubrimos que estamos a 22.000 Km. del mismo, esto limita nuestra posición aún más, a los dos puntos en los cuales la esfera de 22.000 Km. corta la circunferencia que resulta de la intersección de las dos primeras esferas.

O sea, que midiendo nuestra distancia a tres satélites limitamos nuestro posicionamiento a solo dos puntos posibles. Para decidir cual de ellos es nuestra posición verdadera, podríamos efectuar una nueva medición a un cuarto satélite. Pero normalmente uno de los dos puntos posibles resulta ser muy improbable por su ubicación demasiado lejana de la superficie terrestre y puede ser descartado sin necesidad de mediciones posteriores [13].

4.2.2 Sistema de posicionamiento Indoor

Para muchas aplicaciones es un requisito imprescindible conocer la posición del usuario. Los actuales sistemas de posicionamiento global pierden su alcance cuando el receptor se encuentra dentro de un edificio o en áreas urbanas donde las señales satelitales no son lo suficientemente fuertes. Debido a esto, es que en los últimos años han surgidos tecnologías de posicionamiento Indoor para entornos cerrados o reducidos.

A continuación se describen brevemente algunas de las tecnologías de posicionamiento Indoor más utilizadas:

a) Sistema de Posicionamiento Satelital Asistido (GPS-A):

Se trata de una tecnología que ha mejorado el funcionamiento de navegadores GPS de los dispositivos smartphones o teléfonos móviles.

Este sistema usa un teléfono móvil para lograr una localización y posicionamiento. En cuanto se enciende el teléfono, se envía a un servidor externo la identificación de la antena y el teléfono obtendrá como respuesta los satélites que tiene encima y su posición. La posición de los satélites está almacenada en el servidor externo, y según nuestra posición nos dará datos de unos satélites u otros, de manera que la puesta en marcha de la navegación es más rápida. La información que recibimos en el teléfono móvil llega

normalmente vía HTTP³, independientemente de la empresa que nos provea el servicio. Este sistema tiene como ventaja que requiere menor tiempo que el GPS para la puesta en funcionamiento y menor necesidad de señales fuertes provenientes de los satélites al inicio de la navegación [20].

b) Localización por infrarrojos (Active Badge) :

Cada usuario del sistema lleva un pequeño transmisor de infrarrojos (IR), que emite periódicamente un identificador único. El lugar debe estar provisto de sensores que transmitan los datos recibidos a un servidor para calcular la posición de cada usuario. La precisión dependerá del área cubierta por cada sensor, llamada microcelda [20].

c) Sistemas basados en la medida del nivel de potencia de la señal recibida o la relación señal a ruido (Cricket y Active Bat):

Se basa en insertar en cada dispositivo a localizar, un transmisor, que periódicamente emite señal. Dependiendo de la señal transmitida, la localización puede ser por radiofrecuencia (RF) o ultrasonidos. La ventaja que tiene este método con respecto al anterior es que requiere menos receptores [20].

d) Posicionamiento Bluetooth (Blueps):

Este es un sistema de localización para entornos indoor basado en la de radiofrecuencia proporcionada por una red Bluetooth⁴. El sistema se basa en la disminución del nivel de la señal transmitida cuando atraviesa una cierta distancia. Un dispositivo que quiere saber su posición, recibe los niveles de señal de los puntos de acceso cercanos y envía esta información a un servidor central, donde se procesa dicha información para determinar la posición. El uso de un servidor central de proceso permite la implementación de complejos algoritmos y la conexión a este es posible a través de la misma red bluetooth utilizada para la localización [37].

e) Posicionamiento WiFi:

³ HTTP: protocolo de transferencia de hipertexto o, en inglés HyperText Transfer Protocol.

⁴ Bluetooth es una tecnología que permite conectar dispositivos electrónicos entre sí de forma inalámbrica. Por lo tanto pueden conectarse computadoras de escritorio o portátiles, celulares, PDAs y otros dispositivos entre si.

Se trata de utilizar la misma idea del GPS pero con redes Wi-fi⁵. Lo han bautizado WPS (Wi-fi Positioning System).

Básicamente el funcionamiento es el siguiente: cuando se inicia el programa, busca la posición aproximada, con un error de entre 20 y 40 metros, de cualquier dispositivo que funcione sobre 802.11, ya sea teléfono móvil, PDA, smartphone, router o PC. Puede utilizarse como una alternativa más económica que el GPS, pero tiene como desventaja que la precisión de la posición depende mucho de la densidad de redes [21].

Una vez descritas las diferentes tecnologías de posicionamiento, surge la duda de cuál es la más adecuada para servicios de localización. La respuesta se encuentra en las diversas aplicaciones de estos servicios, de tal forma que una tecnología será la más adecuada según el problema a resolver.

Actualmente las computadoras personales son sólo una de las distintas formas de acceder a los recursos de información y servicios, debido a la existencia de una gran variedad de dispositivos y a que la tecnología apunta a ser cada vez más móvil.

Movilidad es quizás la tendencia tecnológica más importante en ICT (Information and Communication Technology). Esto puede atribuirse al incremento del uso de la telefonía móvil, la cual está generalmente disponible para todo tipo de usuario. Estos teléfonos móviles, equipados con GPRS (General Packet Radio Service), proveen un servicio de valor agregado que permite que la información sea enviada y recibida a través de la red de teléfonos móviles. Los sistemas de posicionamiento están disponibles para la localización de usuarios brindando la posibilidad de trabajar con servicios e información personalizada y basada en ubicación. Tanto los sistemas de posicionamiento global y local como los sistemas de posicionamiento InDoor están disponibles y pueden ser usados en asociación con dispositivos móviles.

La forma en que la gente usa los recursos de información cambia constantemente debido al avance de la infraestructura móvil, que cada vez cuenta con mayor ancho de banda y conexión constante a Internet desde cualquier lugar. Esto facilita el desarrollo de servicios de valor agregado “3a”, que pueden ser definidos como: el acceso al sistema es posible desde cualquier lugar (anywhere), en cualquier momento (anytime), y con interacción personalizada (anyhow) [36].

⁵ WI-FI: servicio de internet inalámbrica

El modelo tomado como base para el presente trabajo, representa la ubicación del usuario independientemente de si éste se encuentra en escenarios Indoor u Outdoor y brinda una forma de proveer movilidad a través de ellos. Dependiendo de la aplicación específica se utilizará la tecnología mas adecuada en cada uno de estos escenarios. La selección de dicha tecnología esta fuera del alcance de este trabajo, el cual tiene como objetivo analizar e implementar la funcionalidad de la capa de movilidad que plantea el modelo propuesto.

Capítulo 5

En este capítulo haremos una introducción a la arquitectura en capas en el modelado Orientado a Objetos, explicaremos el diagrama de clases propuesto en [1] y el comportamiento que hemos definido e implementado para cada clase.

5.1 Arquitectura en capas

Los métodos de diseño son un punto importante a tener en cuenta al momento de desarrollar aplicaciones SIG. Un buen diseño nos permite tener soluciones modulares, reusable y fácilmente extensibles. Por todo esto es que la tecnología orientada a objetos es una buena alternativa a la hora de diseñar aplicaciones SIG [28].

Combinando objetos y patrones de diseños, podemos obtener una arquitectura modular en capas que desacopla las entidades del mundo real, que constituyen el dominio de la aplicación, de aquellas características geográficas propias de una aplicación SIG [28].

En la mayoría de las aplicaciones SIG se definen dos capas: una capa Conceptual y una capa Geográfica. La capa Conceptual modela el comportamiento de los objetos de acuerdo a las características propias de la aplicación, mientras que la capa Geográfica extiende al modelo conceptual incorporando información espacial referente a la ubicación de los objetos [28].

El modelado en capas provee una alternativa flexible a la subclasificación o modificación del modelo conceptual, a la hora de extender funcionalidad, permitiendo además la reutilización de la Capa Geográfica en el desarrollo de nuevas aplicaciones de esta naturaleza [28].

El modelo tomado como base, muestra como utilizando la estrategia de diseño previamente descrita, se puede incorporar a la capa Geográfica el concepto de movilidad propio de aplicaciones móviles.

El objetivo de nuestra Tesis es analizar e implementar parte de la funcionalidad provista por la capa de Movilidad independientemente de la capa conceptual.

5.2 Análisis del Modelo

Como ya se menciona en el apartado anterior, basamos el desarrollo de nuestra Tesis en el análisis del diagrama de clases propuesto en [1], y en proponer una implementación que lo sustente. Es decir, establecer si es posible desacoplar los conceptos de movilidad de aquellos propios de una aplicación geográfica, junto con la interoperabilidad de movilidades indoor y outdoor, como se afirma en dicho trabajo.

En el trabajo en el cual nos basamos se plantea un modelo como solución a un conjunto de problemáticas. Las mismas se detallan a continuación:

- Análisis de recorridos, el cual se divide en dos posibles escenarios:
 - Recorridos internos en una sala/edificio.
 - Recorridos que involucran salir de un edificio.
- Determinar la habitación (o área física en general) en la cual se encuentra el usuario.
- Si el usuario se pierde, recalcularle el camino o guiarlo para que retome el camino propuesto.

En el diagrama de clases de la Figura 5.2 se distinguen dos grandes capas. Por un lado la capa Geográfica, la cual incorpora información espacial, y por otro lado, la capa de Movilidad, que permite independizar el comportamiento de estas entidades de lo que se refiere a información geográfica propiamente dicha.

A continuación explicaremos detalladamente cada una de estas capas.

5.2.1 Capa Geográfica

En la capa Geográfica se modela todo lo referente a información espacial, agregando semántica de Georeferenciamiento a entidades conceptuales mediante el concepto de Location [3], el mismo se representa en el modelo mediante la clase “Ubicación”.

Esta capa representa objetos del mundo real en términos de su ubicación espacial, mediante un Sistema de Referencia, DATUM y un Sistema de Coordenadas (conceptos descritos en el Capítulo 3), permitiendo establecer la posición de un objeto particular.

La Figura 5.2.1.1 muestra el diseño de clases de la capa Geográfica.

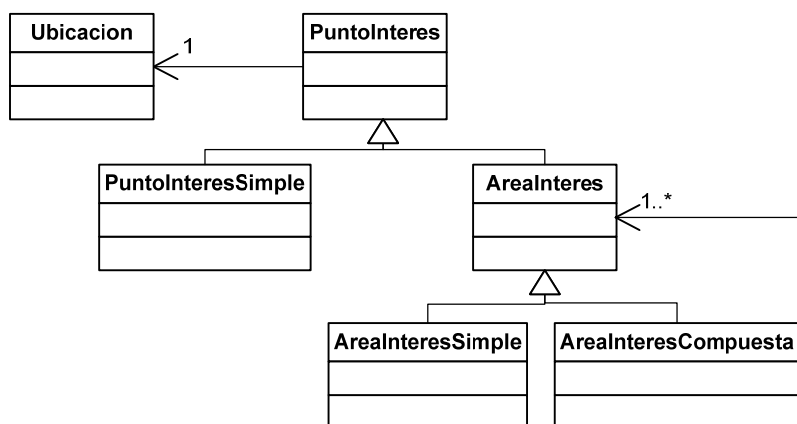


Figura 5.2.1.1 - Capa Geográfica

En el diagrama de la Figura 5.2.1.1, la clase “**PuntoInterés**”, representa los objetos físicos o lógicos del mundo real, de los que es posible conocer su posición geográfica mediante el conocimiento con la clase “Ubicacion”. La abstracción de estos objetos esta representada por la jerarquía de clases PuntoInterés. La clase “PuntoInteres”, abstrae aquellas entidades de la capa Conceptual de las cuales, según la aplicación, es necesario determinar su ubicación geográfica.

Un Punto de interés para una persona puede ser desde una ciudad, como por ejemplo Mar del Plata, hasta un objeto específico en un lugar determinado, como por ejemplo, una obra de arte de un artista famoso.

Como se muestra en la Figura 5.2.1.1 existen dos tipos de puntos de interés: “AreaInteres” y “PuntoInteresSimple”.

AreaInteres: un área de interés puede ser simple o compuesta. Decimos que un área de interés es simple cuando ésta, por si sola, es suficiente para modelar la entidad en cuestión, como puede ser una sala dentro de un museo, mientras que un área de interés compuesta es aquella que se compone de otras áreas, simples o compuestas, como puede ser un museo el cual tiene varias salas. Esta clase esta representada utilizando el patrón de diseño Composite [2].

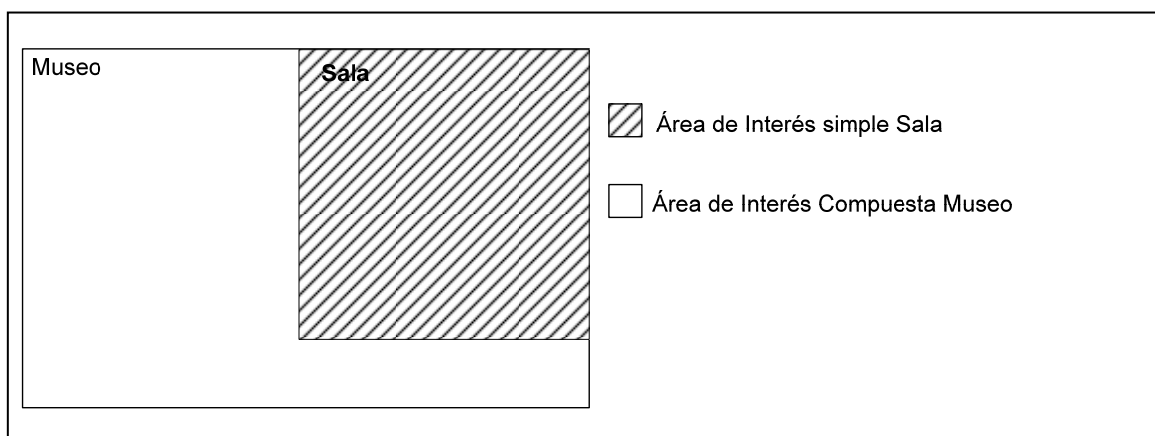


Figura 5.2.1.2 - Área de Interés

En la Figura 5.2.1.2, se muestra al museo como un Área de Interés compuesta, formado por un Área de Interés Simple llamada Sala

PuntoInteresSimple: son aquellos objetos representados topológicamente como un punto. Cada PuntoInteresSimple estará contenido en un AreaInteres, dicha relación puede ser calculada mediante operaciones de topología contemplando el marco geográfico del mismo [1]. A Diferencia de un Área de Interés, un punto de interés simple no puede estar compuesto de otros puntos de Interés.

Un ejemplo de la clase “PuntoInteresSimple” podría ser una obra de arte ubicada en una sala de un museo.



Figura 5.2.1.3 - Punto de Interés Simple

En la Figura 5.2.1.3 se muestra un punto de interés simple llamado Obra de Arte, que se encuentra ubicada en una sala, representada como un Área de Interés.

Una misma entidad podría ser representada como un objeto de la clase “PuntoInteresSimple” o como un objeto de la clase “AreaInteres” dependiendo del nivel de abstracción con el que esta entidad se represente en la aplicación. Por ejemplo, supongamos que, por un lado se quiere modelar el mapa de rutas de la República

Argentina, en el cual solo se necesite saber las ciudades que atraviesan cada una de las rutas. En este caso sería suficiente representar cada ciudad como un punto de interés simple, debido a que no es relevante saber como esta formada cada ciudad. Por otro lado, supongamos que se quiere modelar el mapa de una ciudad turística, determinando los lugares más importantes de esta. En este caso se debería representar a dicha ciudad como un área de interés compuesta, ya que es importante mostrar como esta formada la misma.

5.2.2 Capa de Movilidad

Esta capa agrega el concepto de movilidad a la capa Geográfica. En este trabajo se define la movilidad como la posibilidad de asistir al usuario al momento de encontrar un camino hacia un lugar determinado, brindarle información del contexto, poder establecer los lugares por donde pasó el usuario en un momento determinado, entre otras.

El conjunto de clases que forman la capa de movilidad se pueden dividir, de acuerdo a su responsabilidad, en cuatro grupos, como se observa en la Figura 5.2.2.1.

- I.** PIMovilidad, PuntoDeAcceso y PuntoConexión
- II.** Movilidad y UnidadDeDesplazamiento.
- III.** Usuario, UbicaciónUsuario y Contexto.
- IV.** CaminoSugerido.

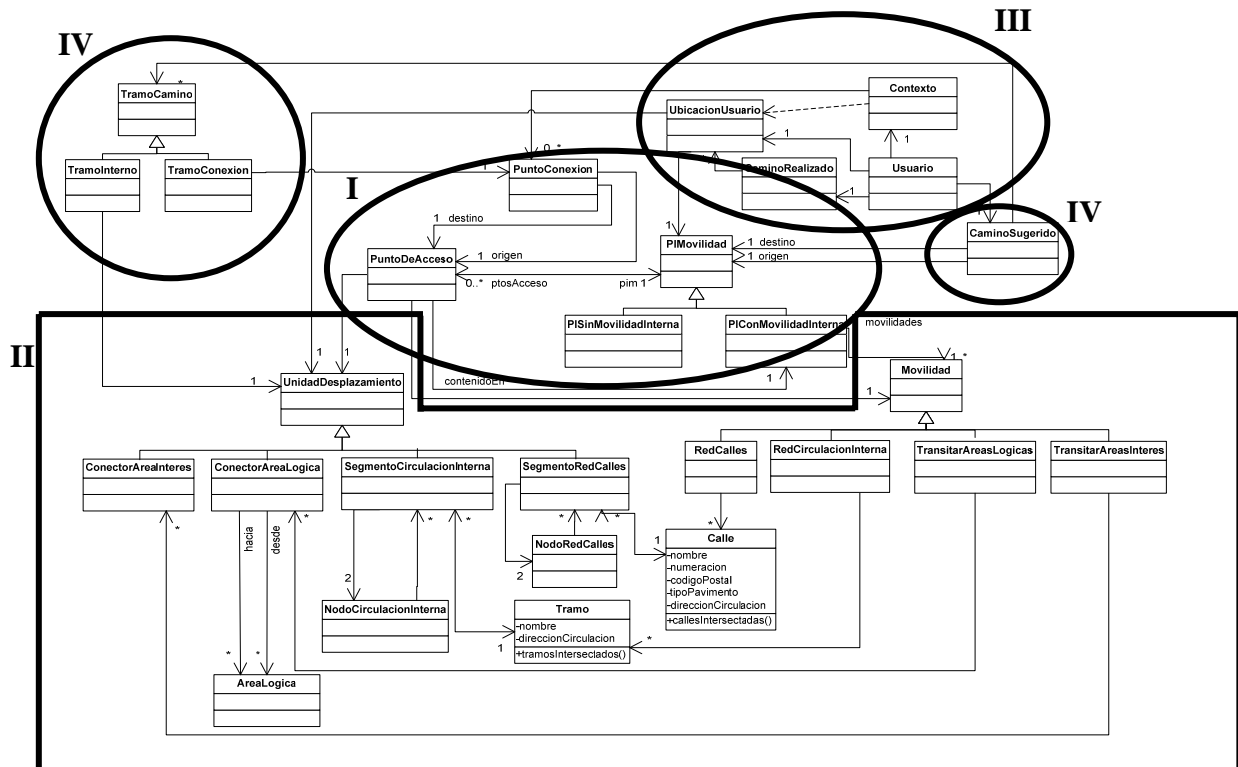


Figura 5.2.2.1 - Grupo de Clases según su responsabilidad

Grupo I- Clases: PIMovilidad, PuntoDeAcceso y PuntoConexion.

La jerarquía de clases “PIMovilidad”, representa los puntos de interés con movilidad que constituyen un decorador [2] sobre los puntos de interés descritos en la capa geográfica. Agrega a los puntos de interés la forma en que un usuario puede recorrerlos, lo que en este trabajo se denomina semántica de movilidad.

Hay dos tipos de objetos de la clase PIMovilidad:

- “PIConMovilidadInterna” (PICMI)
- “PISinMovilidadInterna” (PISMI).

Un objeto de la clase PICMI es un punto de interés que tiene al menos una movilidad asociada, a través de la cual puede ser recorrido. Por el contrario, los objetos de la clase PISMI representan aquellos puntos de interés de los que sólo se sabe como llegar hasta ellos, pero no como movernos en su interior. Por ejemplo: una Obra de Arte en un museo o un edificio al que se tiene acceso restringido.

A un PIMovilidad sólo se puede acceder a través de sus puntos de accesos. Un punto de acceso representa una relación entre una unidad de desplazamiento, el PIMovilidad al cual permite acceder y una de las movilidades que éste tiene disponible, como se ve en la Figura 5.2.2.2. De lo que surge que un PIMovilidad puede ser accedido a través de sus puntos de acceso para una determinada movilidad (Ej: redCirculacionInterna) y, sin embargo, no poder ser accedido en otro tipo de movilidad.

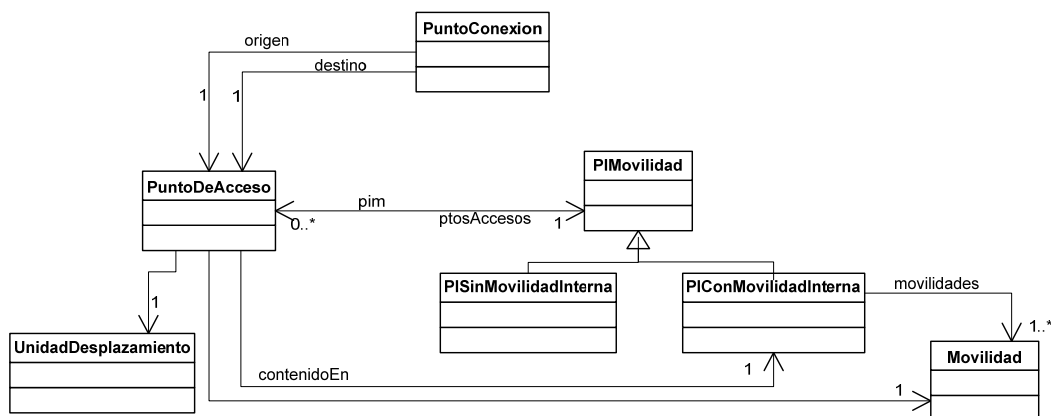


Figura 5.2.2.2 - Relaciones de la Clase PuntoDeAcceso

Cada objeto de la clase PIMovilidad conoce todos los puntos de acceso que permiten llegar hasta él, mediante la relación *ptosAccesos*, un punto de acceso pertenece a un único PIMovilidad al que conoce mediante la relación *pim* y a una única unidad de desplazamiento. Un punto de acceso, además, permite relacionar el PIMovilidad al que hace referencia con el PIMovilidad que lo contiene mediante la relación *contenidoEn*. En otras palabras permite representar un PIMovilidad compuesto por otros PIMovilidad, por ejemplo, un conjunto de salas contenidas dentro de un museo.

Siguiendo con el ejemplo, supongamos tener una sala, la cual tiene dos movilidades asociadas y dos puntos de acceso por los cuales se puede acceder a ella, uno por cada movilidad. La relación *contenidoEn* de cada punto de acceso hace referencia al museo, mientras que la relación *pim* a la sala, como se observa en la Figura 5.2.2.3

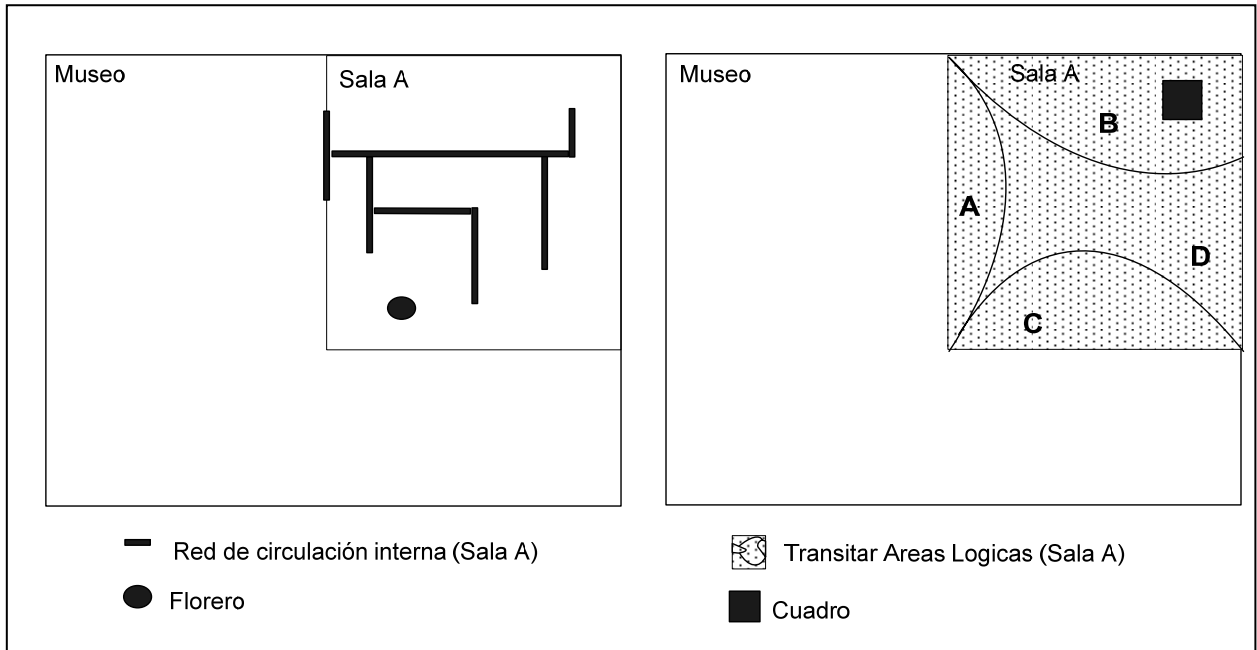


Figura 5.2.2.3 - Sala con dos modalidades diferentes incluida en el Museo.

Para poder movernos entre distintos PIMovilidad es necesario que se relacione el punto de acceso que permite salir de un PIMovilidad y el punto de acceso que permite ingresar a otro. Esta relación entre dos puntos de acceso esta dada por la clase "PuntoConexión". Cada punto conexión conoce dos puntos de acceso, relación origen y destino. Es esta clase la que hace a este modelo distinto de otros, ya que permite en forma transparente pasar de un esquema Indoor a uno Outdoor y viceversa.

Siguiendo con el ejemplo anterior, la comunicación entre la sala y el museo esta dada por un punto conexión que relaciona dos puntos de acceso, uno perteneciente a la sala y otro perteneciente al museo.

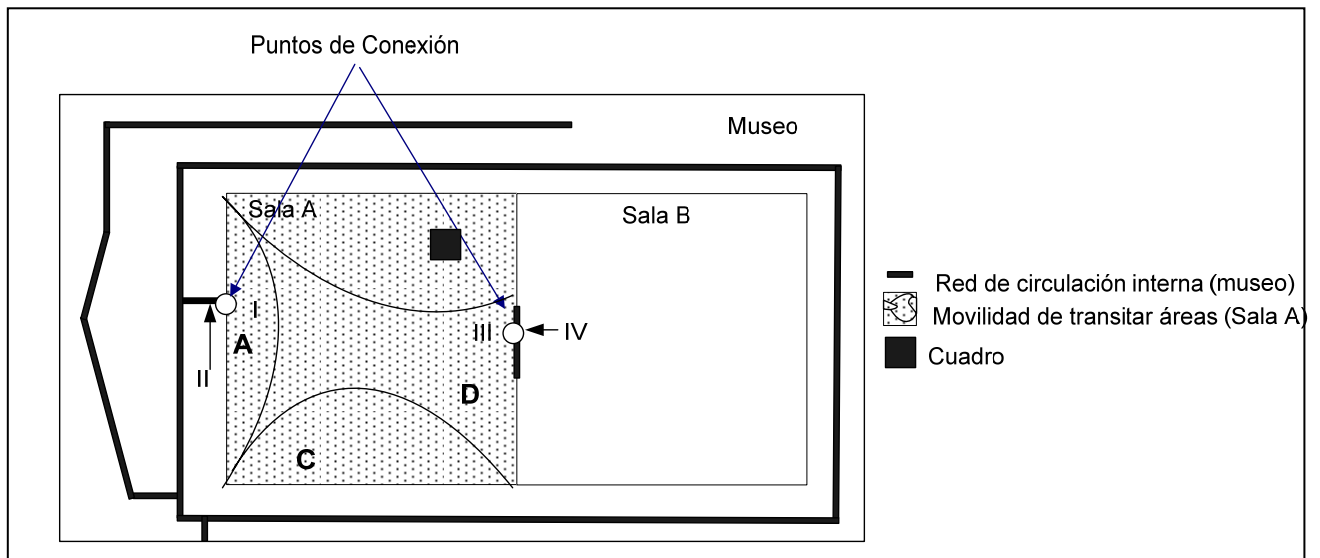


Figura 5.2.2.4 - Puntos conexión entre sala y museo

En la Figura 5.2.2.4 se muestra dos puntos de conexión que permiten acceder a la Sala A, cada uno de ellos compuesto por dos puntos de acceso. Uno permite la comunicación entre la sala A y el museo, mediante los puntos de acceso I y II, mientras que el otro permite la comunicación entre la Sala A y la Sala B, mediante los puntos de acceso III y IV, destacados en la Figura 5.2.2.4.

Grupo II- Clases: Movilidad y Unidad de desplazamiento.

Como mencionamos anteriormente, un objeto de la clase “PIConMovilidadInterna” puede ser recorrido de distintas formas, cada una de estas está representada por un tipo de movilidad.

Los distintos tipos de movilidad asociados a un objeto de la clase “PIConMovilidadInterna” están dados, en el modelo propuesto, por medio de la relación *movilidades* entre las clases “PIConMovilidadInterna” y “Movilidad”.

La jerarquía de la clase “Movilidad” establece los distintos tipos de movilidad contemplados por el Modelo. Estos tipos de movilidades conocen directa o indirectamente a un conjunto de conectores, clase “UnidadDesplazamiento”. Una unidad de desplazamiento representa el menor segmento por donde puede desplazarse un usuario, y esta modelado respetando el patrón de diseño SINGLETON [2].

La clase “UnidadDesplazamiento” establece la jerarquía de los distintos tipos de conectores que se tienen en cuenta en el modelo propuesto, cada tipo de conector relaciona dos objetos, instancias de una misma clase, que corresponden a un mismo tipo de movilidad.

Dadas las relaciones de conocimiento entre las clases anteriormente descritas, podemos ver a la clase “Movilidad” como la clase que encapsula el concepto de grafo, la clase “UnidadDesplazamiento” como la clase que representa las aristas y a las clases que estas conectan, como a los nodos. Por lo tanto cada tipo de movilidad representa un grafo de circulación distinto. En estos grafos nos basaremos para implementar la búsqueda del camino deseado. Si bien se puede establecer una analogía entre esta parte del modelo y la estructura de un grafo, debido a que sus relaciones lo permiten, éste tiene un comportamiento mucho más amplio ya que permite establecer por ejemplo, la ubicación de determinado ‘objeto’, y si el mismo se encuentra fuera o dentro de un cierto lugar.

La forma de diseño utilizada facilita la incorporación de nuevos grafos de circulación, debido a que, cada uno de estos grafos es independiente del resto. Para agregar una nueva forma de poder recorrer un PIconMovilidadInterna es necesario definir la mínima unidad de movilidad, que constituirá las aristas del grafo, como subclase de la clase “UnidadDeDesplazamiento”, y los nodos que estas relacionan. Por ultimo, se debe definir la clase que contenga a este grafo como un nuevo tipo de movilidad, es decir como subclase de la clase “Movilidad”.

En la Figura 5.2.2.5 se muestran los distintos tipos de movilidades que representan los grafos de circulación contemplados en el modelo.

5. Análisis del Modelo

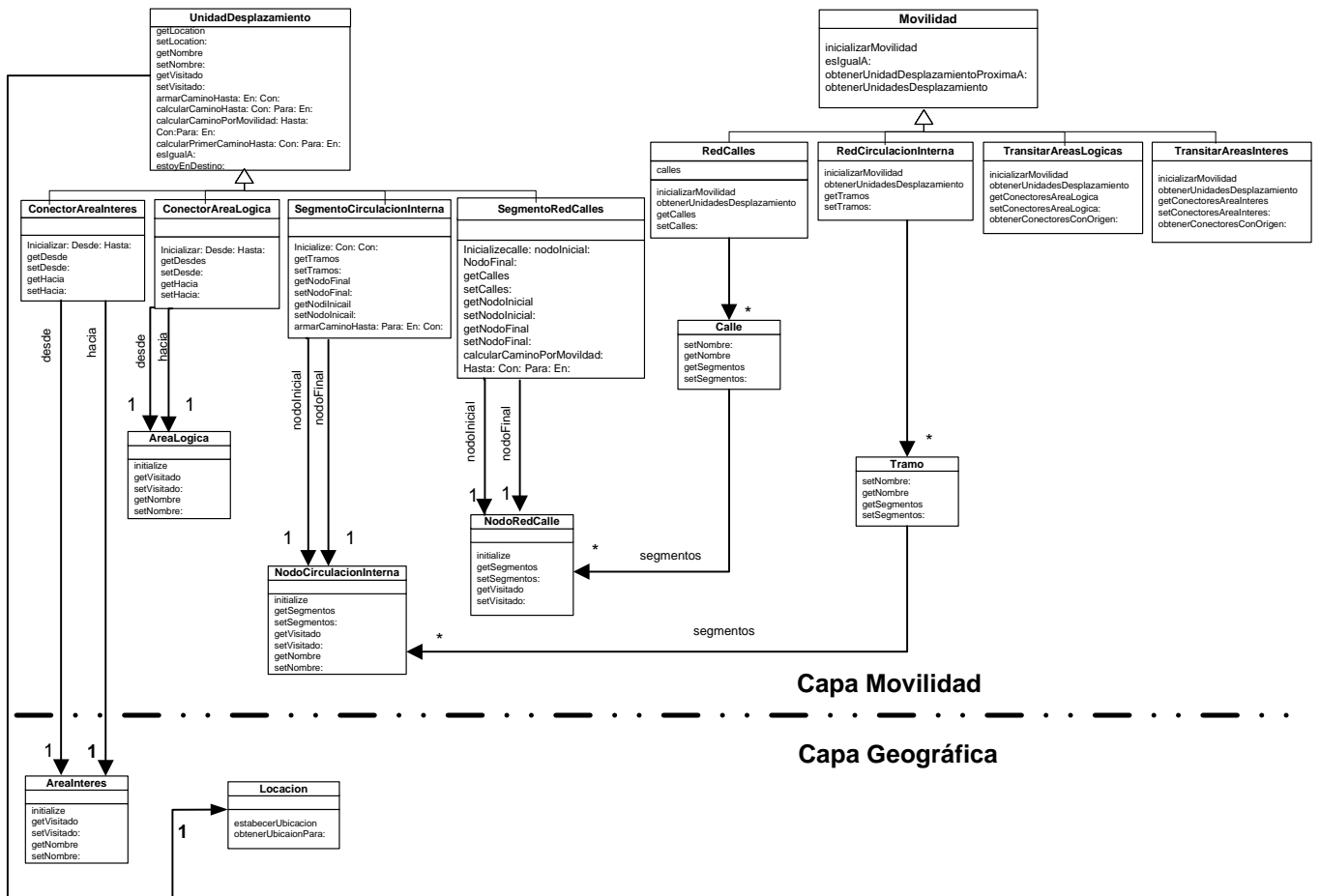


Figura 5.2.2.5 – Tipos de movilidades

Hasta acá hemos introducido los conceptos de Movilidad y Unidad de Desplazamiento, los cuales se relacionan, como se mencionó anteriormente, mediante la clase “PuntoDeAcceso”, cuando es necesario conectar dos movilidades diferentes, como se observa en la Figura 5.2.2.6.

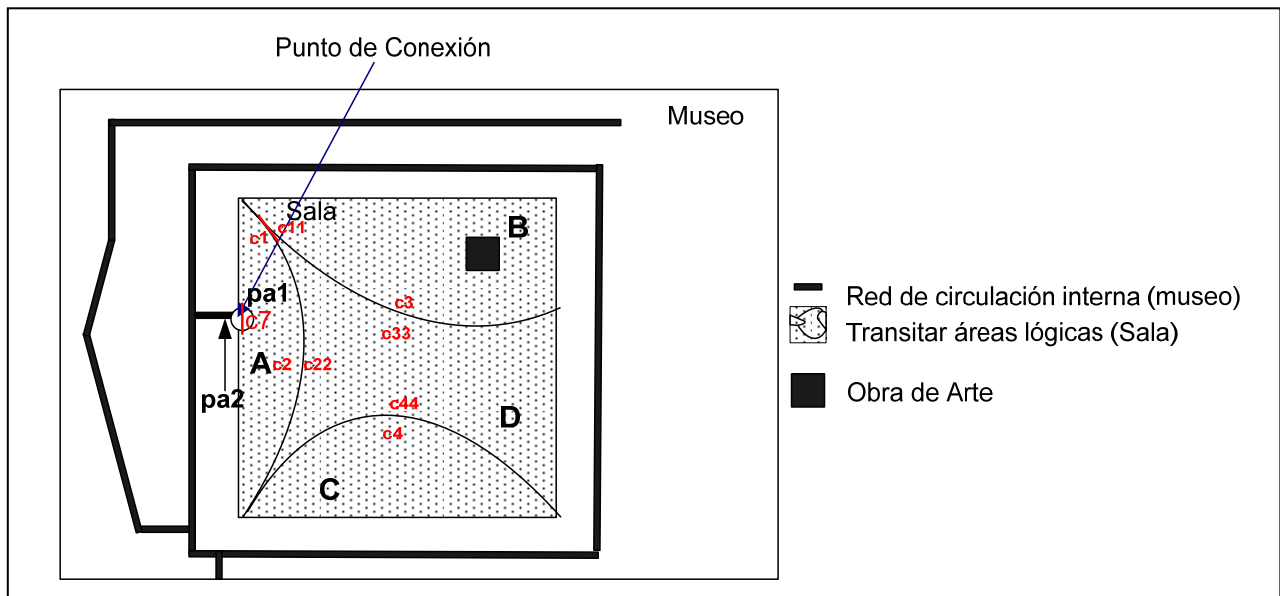


Figura 5.2.2.6 – Punto de Conexión.

En la Figura 5.2.2.6 se representa una sala, con la movilidad “TransitarAreasLogicas”, en el contexto del Museo, al cual se le asocia el tipo de movilidad “RedCirculacionInterna”. Para comunicar estos dos “PIConMovilidadInterna”, existe un punto de conexión entre ambos. Este punto de conexión cuenta con un punto de acceso llamado **pa1**, que hace referencia a un conector de la sala, y con otro llamado **pa2**, que hace referencia a un conector del Museo. El punto de acceso **pa1** relaciona un conector del tipo “ConectorAreaLogica” denominado **c7**, con la movilidad TransitarAreasLogicas perteneciente a la sala. De la misma forma, el punto de acceso **pa2** relaciona un conector del tipo “SegmentoCirculacionInterna” con la movilidad RedCirculacionInterna del Museo. Es decir, cada punto de acceso relaciona una unidad de desplazamiento con la movilidad correspondiente.

Según el modelo planteado, cada unidad de desplazamiento, conecta dos instancias de la misma clase, mediante dos relaciones de conocimiento. Sin embargo, debido a que un punto de conexión relaciona dos instancias de distintas subclases de la clase “UnidadDsplazamiento“, sucede que no se puede representar una unidad de desplazamiento con un solo extremo, es decir, con solo un origen o un destino, como es el caso del conector llamado **c7**.

5. Análisis del Modelo

Para poder resolver esta situación, realizamos una modificación al modelo que nos permite representar conectores con estas características, es decir, con un solo nodo. Dicha modificación consiste en introducir para cada clase que constituye un “NODO” del grafo, el tipo nodo nulo correspondiente, mediante el patrón NullObject [24].

La Figura 5.2.2.7 refleja el cambio mencionado.

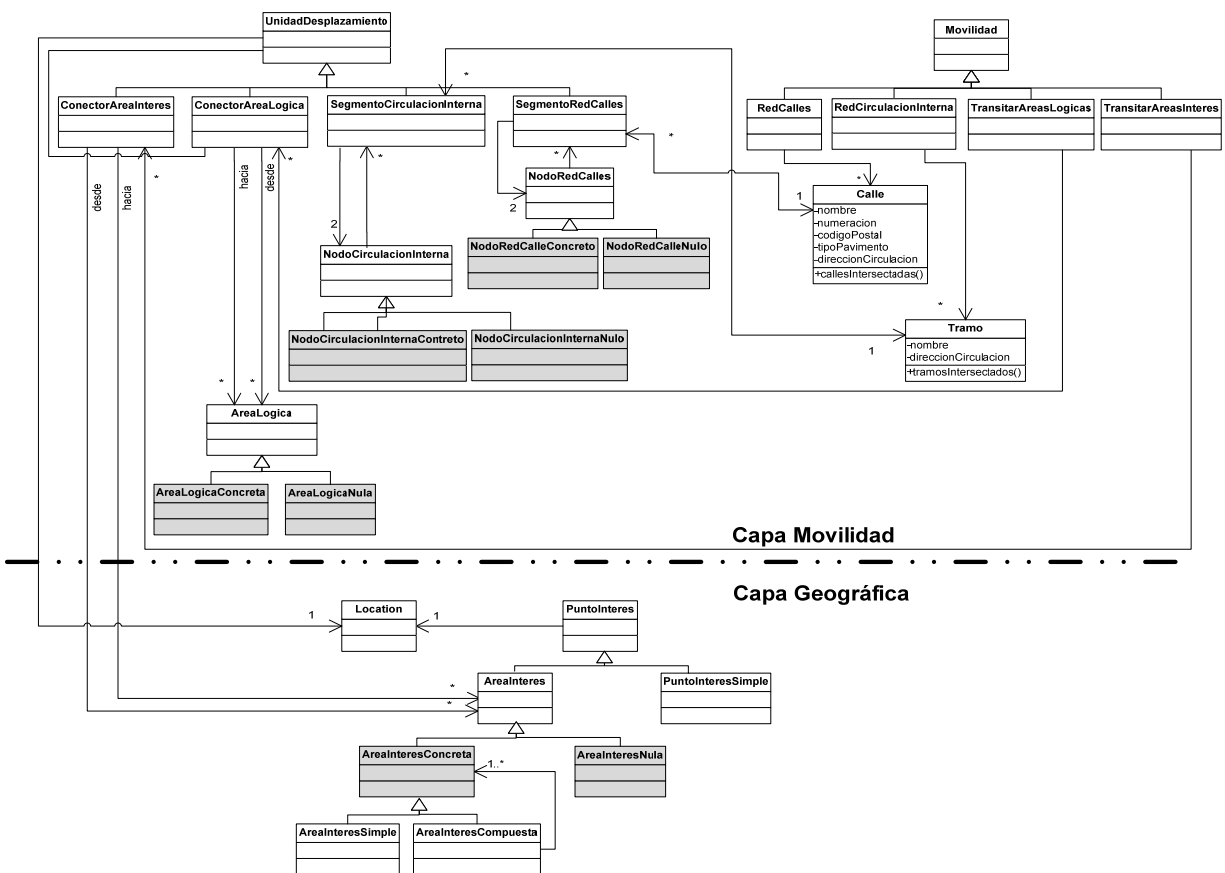


Figura 5.2.2.7 – Diagrama de clases modificado.

En la Figura 5.2.2.7, se muestra el diagrama de clases en el cual se han resaltado las clases incorporadas. Cada jerarquía de clases que constituyen los nodos de los distintos grafos de circulación esta formada por dos subclases, una clase que representa al Nodo concreto,

como se representaba antes de la modificación y otra clase que representa al Nodo Nulo correspondiente.

Grupo III - Clases: Usuario, Ubicación y Contexto.

El modelo contempla al usuario, su ubicación y su contexto.

Cuando se habla de usuario nos referimos al dispositivo móvil que interactúa con el sistema.

Cuando hablamos de ubicación hacemos referencia a la posición geográfica más cercana a la ubicación real del usuario. Recordemos que el sistema representa la ubicación geográfica a través de las unidades de desplazamiento.

La clase “UbicaciónUsuario” representa la ubicación actual del usuario. Esta ubicación se expresa en términos de una instancia de la clase “UnidadDesplazamiento”, ya que es ésta la que tiene acceso a los mecanismos de georeferenciamiento provistos por la clase “Ubicación”, y a un PIMovilidad que la contextualiza.

El contexto, representa la información vinculada a las preferencias del usuario, y los puntos de conexión habilitados o deshabilitados para el mismo, con lo cual depende de la ubicación del usuario. La implementación de este concepto queda fuera del alcance del presente trabajo

En la Figura 5.2.2.8 se muestra como se relacionan las clases Usuario, UbicaciónUsuario y Contexto. Con línea punteada se refleja la dependencia que existe entre la clase “Contexto” y la clase “UbicaciónUsuario”.

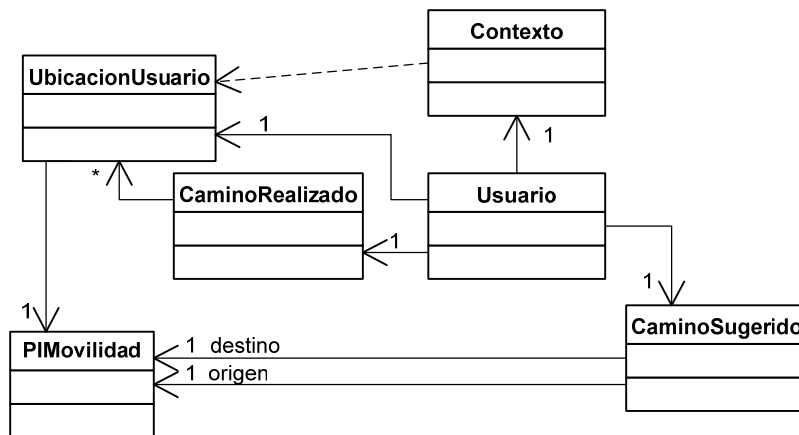


Figura 5.2.2.8 - Usuario, Ubicación y Contexto

Además, como se puede observar en la Figura 5.2.2.8, la clase “Usuario” mantiene una relación de conocimiento con el camino que le propone el Sistema, clase “CaminoSugerido” y con el camino que éste lleva realizado, clase “CaminoRealizado”. La implementación de este concepto también queda fuera del alcance del presente trabajo.

Grupo IV – Clase: Camino Sugerido

Otro grupo de clases son las que forman el camino sugerido, que representa el camino que debe recorrer un usuario para alcanzar el destino deseado.

La clase “CaminoSugerido” conoce a un conjunto de tramos, representados mediante la clase “TramoCamino”. Esta última clase constituye una jerarquía compuesta por dos subclases, “TramoInterno” y “TramoConexion”, que representan respectivamente, tramos internos a un mismo “PIMovilidad” y tramos que permiten cambiar de “PIMovilidad”. Para realizar esto, un tramo interno hace referencia a una unidad de desplazamiento y un tramo conexión a un punto de conexión, como se puede observar en la Figura 5.2.2.9.

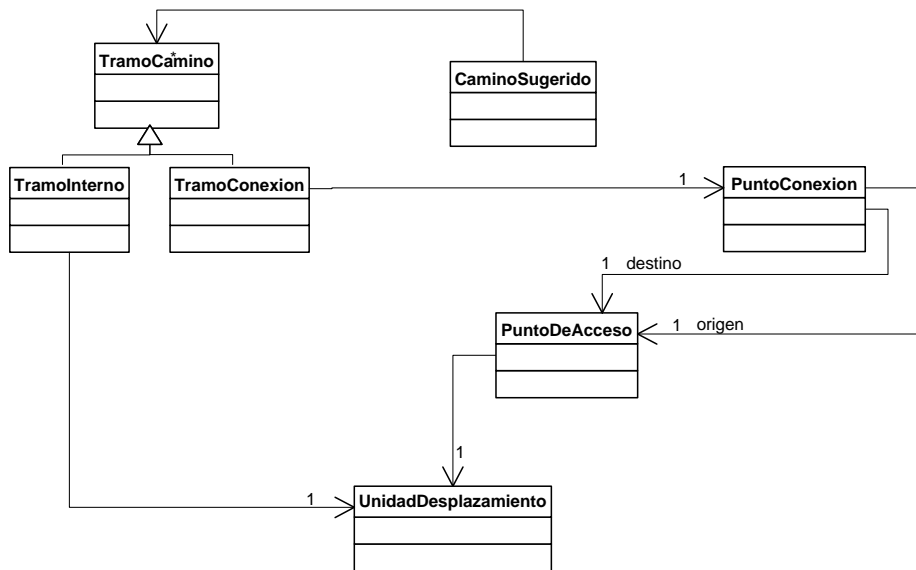


Figura 5.2.2.9 - Camino Sugerido.

Como se visualiza en la Figura 5.2.2.9, el camino sugerido, mediante el conjunto de tramos que lo forman, establece una sucesión de unidades de desplazamiento pertenecientes a una o varias movilidades.

5.3 Movilidades

En esta sección explicaremos detalladamente cada una de las diferentes movilidades que se mencionaron anteriormente en este capítulo. En cada caso presentaremos ejemplos en forma gráfica.

Como ya hemos visto en la sección 5.2.2, en el modelo se proponen cuatro tipos de movilidades que son subclase de de la clase “Movilidad”, donde cada una representa una forma de recorrer una instancia de la clase “PIConMovilidadInterna” de una manera particular:

Estos tipos de movilidades son:

- Transitar Áreas de Interés

- Transitar Áreas Lógicas
- Red de Circulación Interna
- Red de Calles

A continuación se describen cada una de estas movilidades.

5.3.1 Movilidad “Transitar Áreas Interés”

Este tipo de movilidad se constituye por la clase “TransitarAreaInteres”, la cuál se relaciona con un conjunto de conectores, objetos de la clase “ConectorAreaInteres”. Estos conectores representan las aristas del grafo de circulación, y como tales, relacionan dos nodos, los cuales son instancias de la clase “AreaInteres”. Esta relación se expresa mediante el conocimiento llamado *desde* y *hacia*. La Figura 5.3.1.1 muestra las relaciones previamente descritas, entre las clases mencionadas.

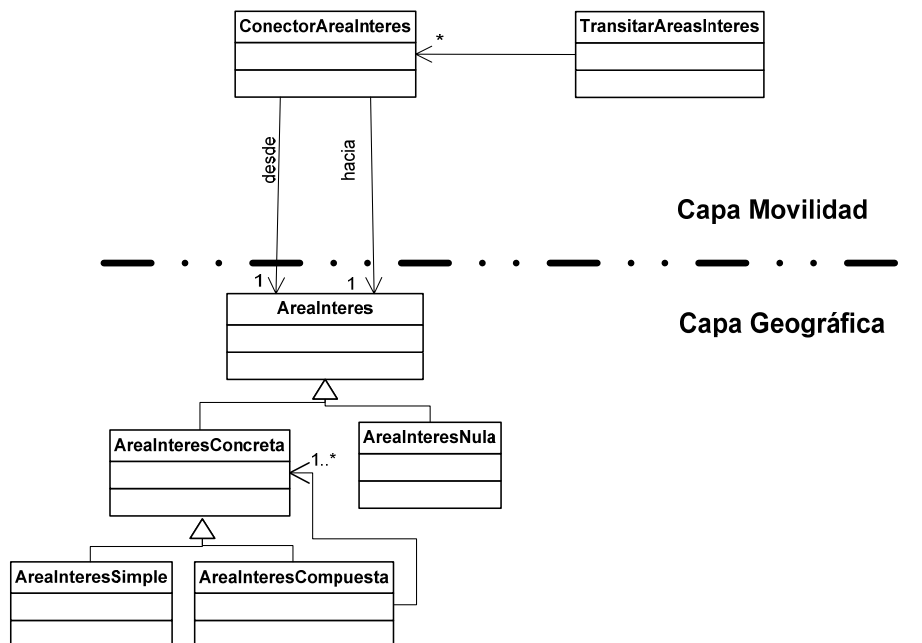


Figura 5.3.1.1 - Movilidad Transitar Áreas de Interés

La Figura 5.3.1.1 muestra el diagrama de clases correspondiente a la movilidad Transitar Áreas de Interés.

Un área de interés representa una división establecida en la capa Geográfica.

Tomando todos los conectores que tienen una misma área como origen, podemos determinar todas las áreas con las que ésta se comunica.

En la Figura 5.3.1.2 se muestra un ejemplo para la movilidad Transitar Áreas Interés:

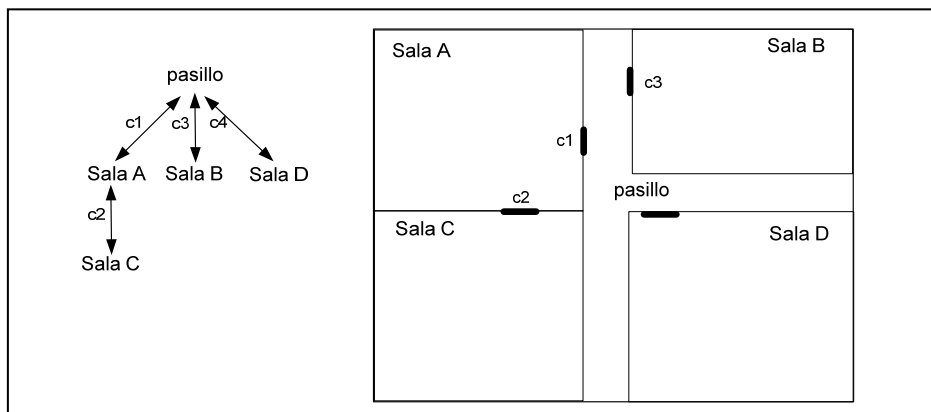


Figura 5.3.1.2 – Movilidad Transitar Áreas Interés

En la Figura 5.3.1.2, se observa al Museo, el cual es una instancia de la clase “AreaInteresCompuesta” formada por las áreas **Pasillo**, **SalaA**, **SalaB**, **SalaC** y **SalaD**, las cuales son instancias de la clase “AreaInteresSimple”. Esta movilidad establece que el pasillo del Museo se comunica tanto con la **SalaA**, con la **SalaB** y con la **SalaD**. Para poder llegar a la **SalaC**, es indispensable tener que pasar por la **SalaA**. En la parte gráfica la conexión entre las áreas se identifican como **c1**, **c2**, **c3** y **c4**.

5.3.2 Movilidad “Transitar Áreas Lógicas”

Este tipo de movilidad se utiliza para poder recorrer una instancia de la clase “PIConMovilidadInterna” dividida lógicamente en áreas, estas áreas pueden estar trazadas por ejemplo por objetos como una soga o una línea dibujada en el piso.

A este tipo de movilidad lo compone un conjunto de conectores de la clase “ConectorAreaLogica”. Cada conector relaciona dos objetos, instancias de la clase “AreaLogica”, mediante las relaciones de conocimiento *desde* y *hacia*. Tomando todos los conectores que tienen a un mismo área como origen, es decir, relación desde, podemos determinar con que otras áreas se comunica.. La relación entre estas clases se muestra en la Figura 5.3.2.1.

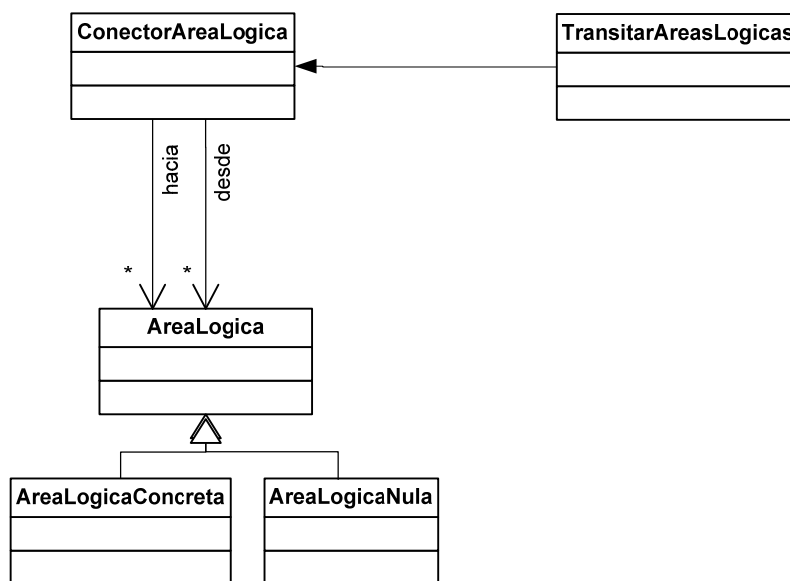


Figura 5.3.2.1 - Movilidad Transitar Áreas Lógicas

La Figura 5.3.2.2 muestra una instancia de la clase “PIConMovilidadInterna“, denominada Sala, asociada a la movilidad Transitar Áreas Lógicas.

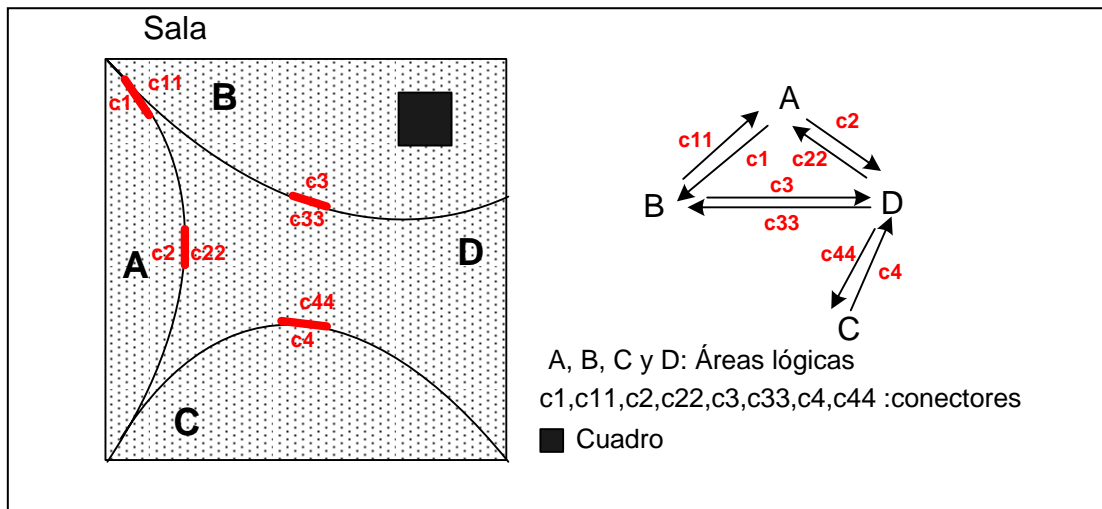


Figura 5.3.2.2 - Sala con Movilidad Transitar Áreas Lógicas

Como podemos ver en la Figura 5.3.2.2, la Sala, esta dividida en 4 áreas lógicas, etiquetadas como A, B, C y D.

Cada área tiene asociado un conjunto de conectores, en este caso,

- El área A tiene asociado el conector llamado **c1** que va desde el área A hacia el área B y el conector llamado **c2** que va desde el área A hacia el área D.
- El área B tiene asociado el conector llamado **c11** que va desde el área B hacia el área A y el conector llamado **c3** que va desde el área B hacia el área D.
- El área C tiene asociado solo el conector llamado **c4** que va desde el área C hacia el área D.
- El área D tiene asociado el conector llamado **c22** que va desde el área D hacia el área A, el conector llamado **c33** que va desde el área D hacia el área B y el conector llamado **c44** que va desde el área D hacia el área C.

5.3.3 Movilidad “Red de Circulación Interna”

Este tipo de movilidad está determinado por la clase “RedCirculacionInterna” y se compone de un conjunto de tramos, instancias de la clase “Tramo”. Cada tramo tiene un nombre y una dirección de circulación, la cual se determina a partir de la orientación de todos los segmentos que componen al tramo, objetos de la clase “SegmentoCirculacionInterna”. Cada segmento pertenece a un solo Tramo, y conecta a dos **nodos** del grafo, que son instancias de la clase “NodoCirculacionInterna”. Para determinar la orientación de un segmento, llamamos a uno de los extremos *nodoInicial* y al otro *nodoFinal*.

La Figura 5.3.3.1 muestra un Segmento de Circulación Interna con un nodo Inicial llamado A y un nodo Final llamado B



Figura 5.3.3.1 – Segmento de Circulación Interna.

Como se muestra en el ejemplo de la Figura 5.3.3.1, el nodo inicial es A y el final es B, por lo que se dice que la orientación del segmento es de A hacia B (A-B).

Cuando un nodo conecta dos segmentos cumple el rol de *nodoFinal* para uno y de *nodoInicial* para el siguiente, tal como se puede observar en la Figura 5.3.3.2 para el nodo llamado N2.

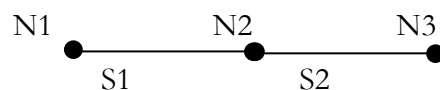


Figura 5.3.3.2 – Relación entre dos segmentos.

Esto no significa que un nodo solo conecte dos segmentos, en realidad, como se muestra en la Figura 5.3.3.3, un mismo nodo puede ser *nodoInicial* o *nodoFinal* de dos o más segmentos.

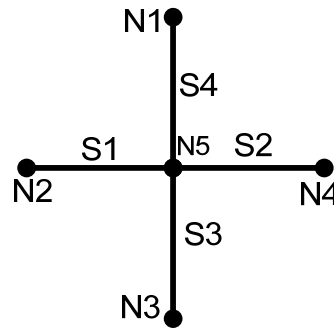


Figura 5.3.3.3 – Relación entre mas de dos segmentos.

En la Figura 5.3.3.3 se muestra la intersección de cuatro segmentos (S1, S2, S3 y S4) unidos por un mismo nodo (N5).

En la Figura 5.3.3.4 se observa el diagrama de clases para la movilidad Red de Circulación Interna.

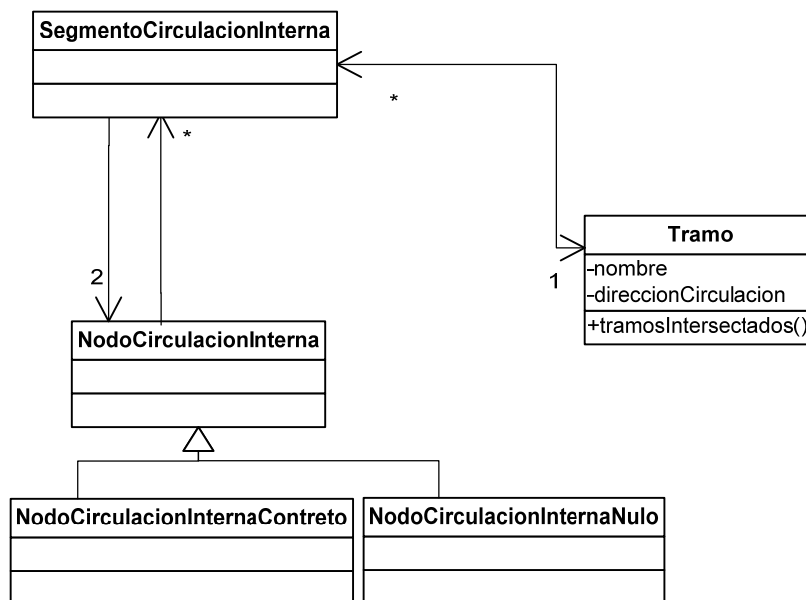


Figura 5.3.3.4 – Movilidad Red de Circulación Interna

La Figura 5.3.3.5 muestra un PIconMovilidadInterna llamado Sala, representado con la movilidad Red de Circulación Interna.

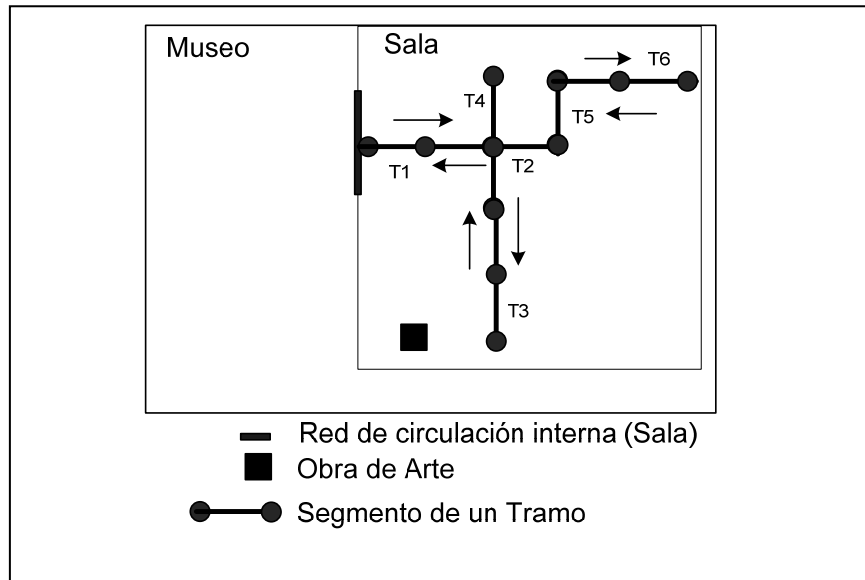


Figura 5.3.3.5 – Sala con movilidad Segmento Circulación Interna

En la Figura 5.3.3.5 se representa a la Sala con la movilidad “RedCirculacionInterna”, donde la misma esta formada por 6 tramos, que se identifican en la Figura mediante los tramos llamados T1 a T6, como se muestra en la Figura 5.3.3.6

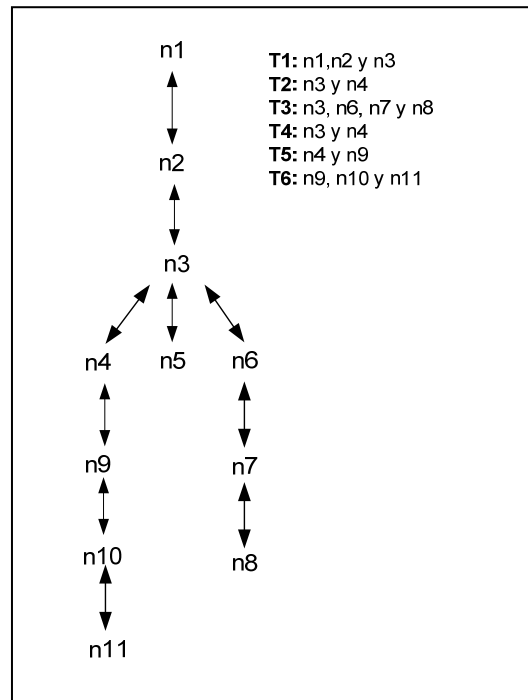


Figura 5.3.3.6- Grafo de Circulación de Segmento de Circulación Interna

La Figura 5.3.3.6 muestra la representación del grafo de circulación establecido para el ejemplo de la Figura 5.3.3.5.

5.3.4 Movilidad Red de Calles

Este tipo de movilidad está formado por la clase “RedDeCalle” que se relaciona con un conjunto de calles, objetos de la clase “Calle”. A su vez una calle está constituida por un conjunto de segmentos, representados por instancias de la clase “SegmentoRedCalle”.

Cada segmento conoce la calle a la cual pertenece y a los nodos que constituyen sus extremos. Los nodos que relacionan estos segmentos son objetos de la clase “NodoRedCalles”, llamados *nodoInicial* y *nodoFinal*. Cada nodo puede pertenecer a más de un segmento. La Figura 5.3.4.1 muestra el diagrama de clases correspondiente.

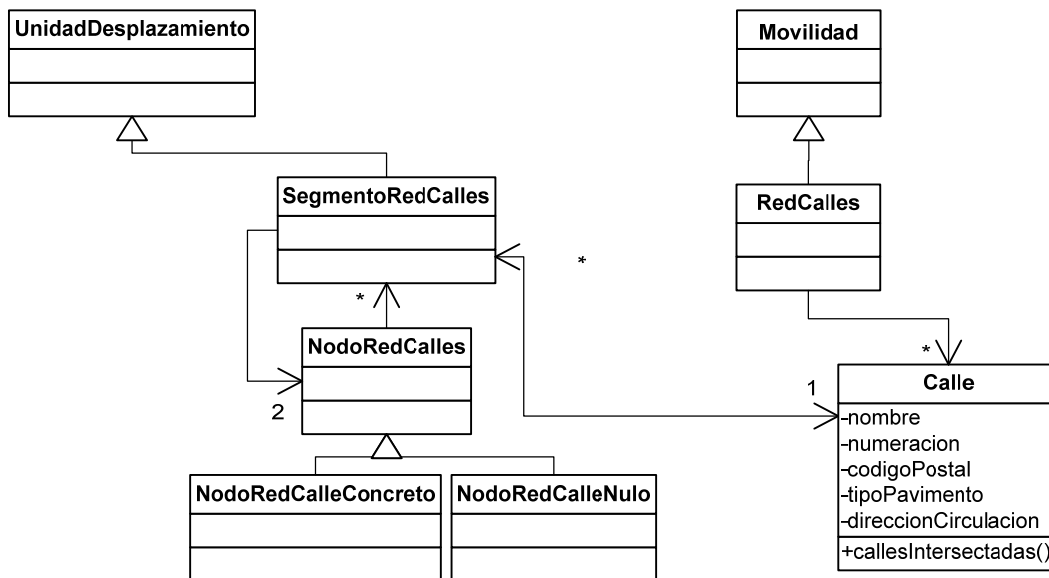


Figura 5.3.4.1 - Movilidad Red De Calles

Analizando este último tipo de movilidad hemos observado que la definición de la clase “Calle” se adecua más a una clase de la capa Conceptual que a una clase de la capa de Movilidad, ya que representa una entidad del mundo real propia de un dominio de aplicación específico. Esto difiere, en cierta forma, con el concepto de arquitectura en capas.

Si comparamos esta movilidad con la movilidad Red de Circulación Interna veremos que la clase “Calle” es una representación específica de la clase “Tramo” y en definitiva la movilidad Red de Calles es un caso particular de la movilidad Red de Circulación Interna. Por otro lado la movilidad RedCalles se asocia a una movilidad outdoor, y la movilidad RedCirculacionInterna se asocia a una movilidad indoor. Además, el concepto de calle está muy ligado a la cartografía, ya que es la forma más utilizada para establecer una ubicación. Es por ello que se justifica que la capa de Movilidad contemple el modelado de la calle [29].

La Figura 5.3.4.2 muestra un ejemplo de la movilidad Red de Calles.

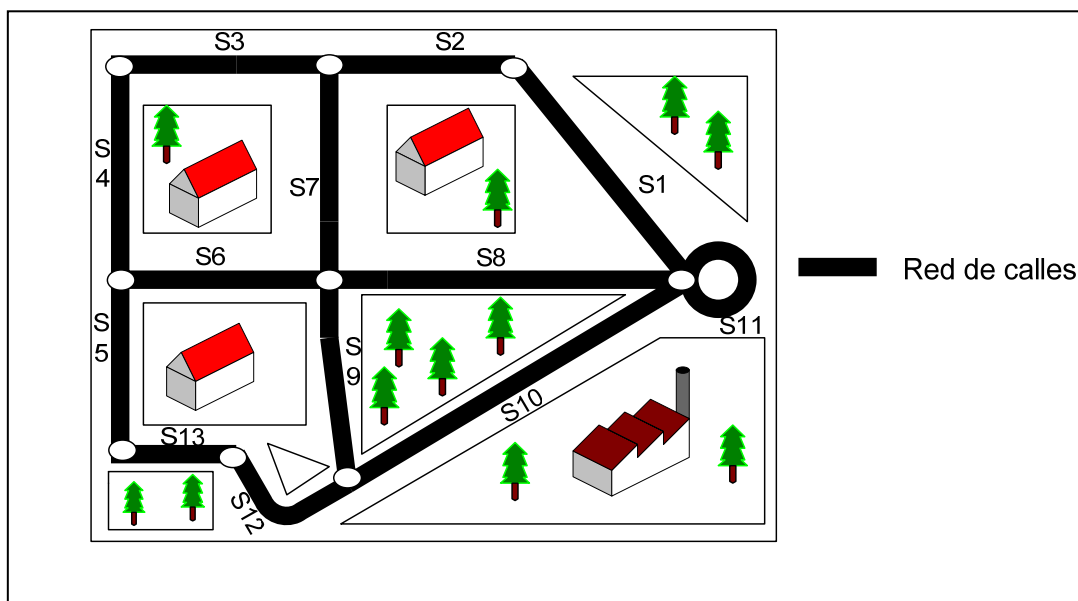


Figura 5.3.4.2 - Movilidad Red De Calles

En la Figura 5.3.4.2, se puede observar el modelado de calles, las cuales están formadas por un conjunto de segmentos. Por ejemplo, se puede ver, una calle formada por dos segmentos llamados S7 y S9, otra formada por tres segmentos llamados S10, S12 y S13. Algunas calles formadas por un solo segmento, como por ejemplo la constituida por el segmento S1.

En la Figura 5.3.4.3, se muestra como se forma el grafo de circulación para este ejemplo.

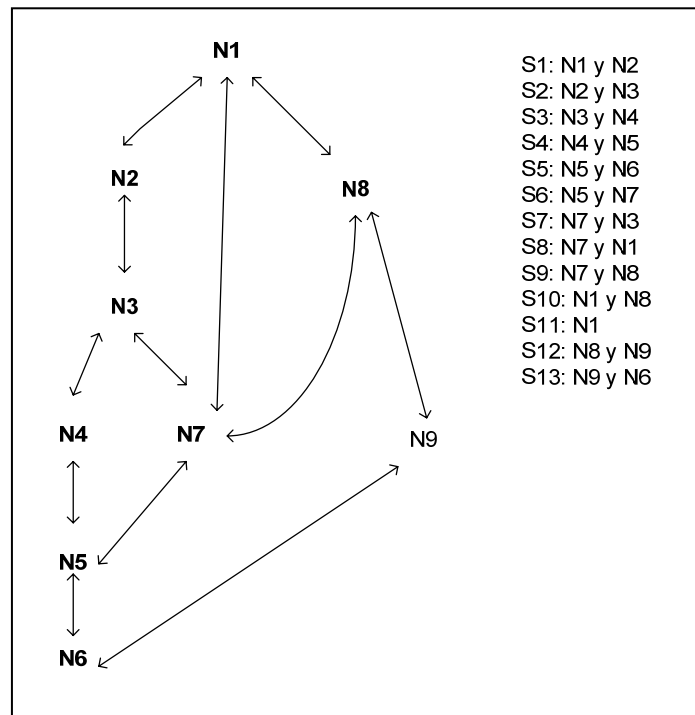


Figura 5.3.4.3 - Grafo de Circulación Red De Calles

La Figura 5.3.4.3 muestra la representación del grafo de circulación establecido para el ejemplo de la Figura 5.3.4.2.

5.4 Reposicionamiento del Usuario

Como se mencionó anteriormente, un mismo objeto, instancia de la clase “PIMovilidad”, puede tener uno o mas tipos de movilidad asociados, los cuales permiten recorrerlo de distintas formas.

En nuestra implementación, cuando buscamos un camino, lo hacemos recorriendo la colección de movilidades del PIMovilidad, hasta encontrar el destino o hasta que se haya finalizado dicha colección.

Para comenzar la búsqueda tomamos la primera movilidad de esta colección, a la cual llamamos *movilidad actual*. Si al buscar por esta movilidad no se puede establecer un camino debemos tomar la siguiente movilidad. El cambio de movilidad provoca tener que reposicionar al usuario en una nueva unidad de desplazamiento equivalente, perteneciente a

la nueva movilidad. Para realizar este reposicionamiento nos valemos de la funcionalidad provista por la capa geográfica, de modo tal de establecer la unidad de desplazamiento perteneciente a la nueva movilidad más cercana a la posición actual del usuario, en términos de coordenadas geográficas. La unidad de desplazamiento obtenida será tomada como la nueva ubicación del usuario.

En la Figura 5.4.1 se observa gráficamente un ejemplo.

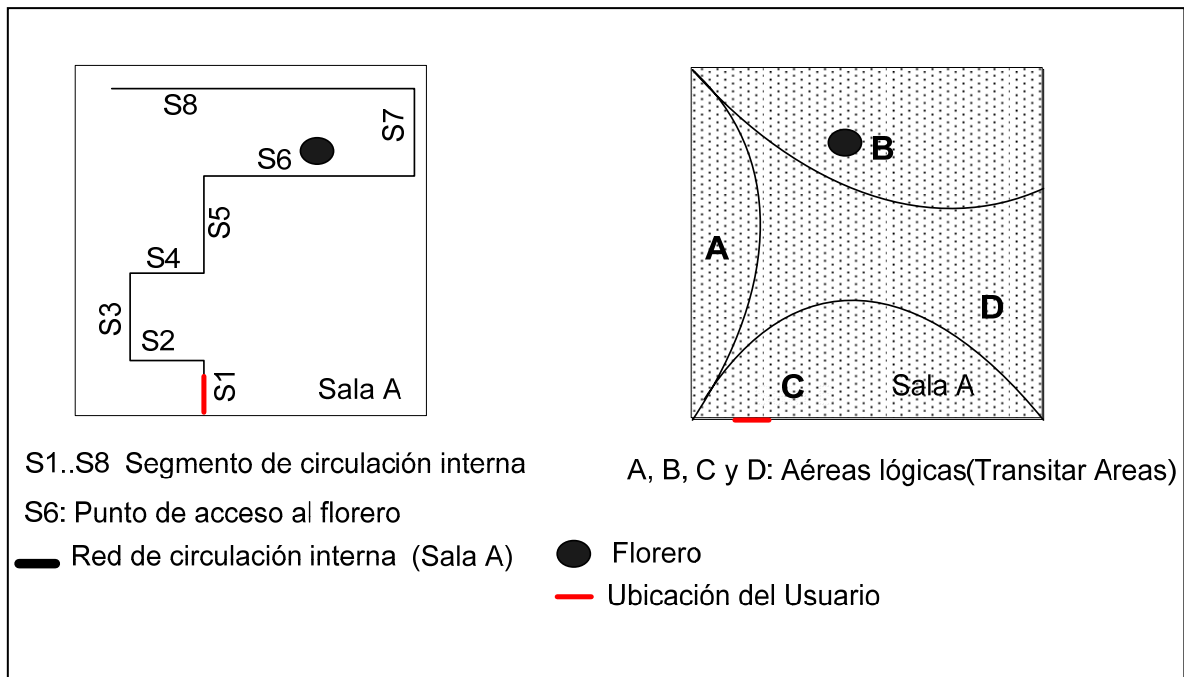


Figura 5.4.1 – Reubicación del Usuario al cambiar de Movilidad

Como se puede observar en la Figura 5.4.1, el usuario se encuentra posicionado en el Segmento de Circulación Interna S1, perteneciente a la movilidad “RedCirculacionInterna”. Al cambiar a la movilidad “TransitarAreasLogicas”, el usuario es reposicionado en el Punto de Acceso a la Sala que se encuentra en el AreaLogica C.

Capítulo 6

En este capítulo explicaremos la funcionalidad implementada del Sistema, mediante la utilización de diagramas de casos de uso y diagramas de Secuencia, expresados en UML 2 [25].

6.1 Casos de Uso

Más allá de toda la funcionalidad que el modelo puede proveer en esta capa, nosotros nos centramos en el análisis de recorridos.

En esta sección explicaremos nuestra implementación, que permite determinar la posición actual del usuario y buscar un camino desde esta posición hasta un punto de interés determinado por el mismo, contemplando los distintos tipos de movilidades.

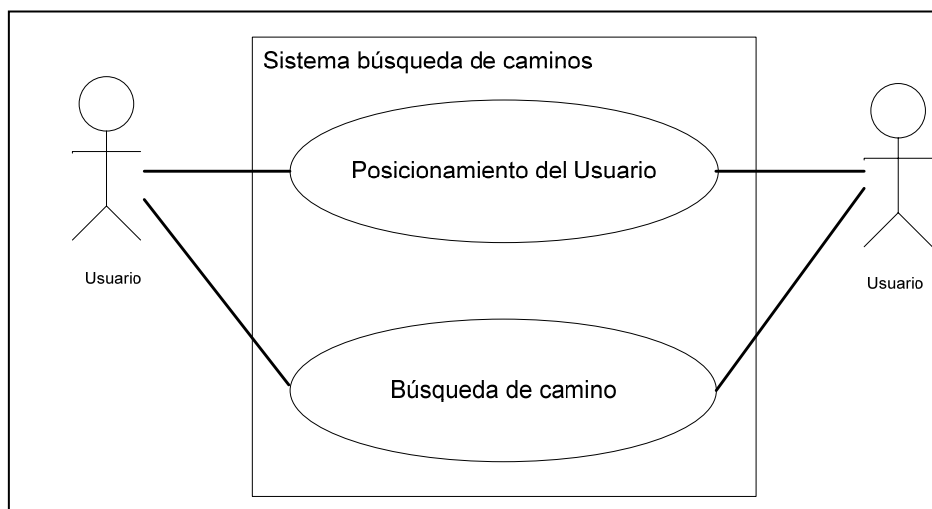


Figura 6.1 - Diagrama de Casos de uso.

En la Figura 6.1 se muestran los casos de usos contemplados en el presente trabajo. A continuación agregamos la conversación usando notación UML 2 [25] de los casos de usos introducidos.

Caso de Uso: **Posicionamiento del Usuario**

Resumen de la identificación		
Título	Posicionamiento del Usuario	
Descripción Contextual	El usuario enciende su dispositivo móvil, ingresa al sistema de búsqueda de caminos, el cual determina en ese momento la posición donde se encuentra el dispositivo móvil. La posición obtenida se muestra gráficamente en el dispositivo móvil.	
Fecha de Creación	01/04/2008	
Desarrollado por	Claudia Ilarragorri - Leticia Durante	
Prioridad	Alta	
Número de Versión	1.0	
Ultima Modificación		
Actores Involucrados	Usuario, Sistema de búsqueda de caminos	
Descripción de recursos		
- Dispositivo móvil provisto de sistemas de posicionamiento.		
Descripción de la conversación		
Pre-Condiciones	El sistema esta listo para determinar la ubicación	
Post-Condiciones	Se muestra la ubicación actual de usuario en su dispositivo móvil. El sistema esta listo para inicial una búsqueda de camino.	
Acciones Del Actor	Respuesta de Sistema	
	Curso Normal	Curso Alternativo
1- El usuario enciende su dispositivo móvil.		
2- Ingresa al Sistema de búsqueda de caminos.	2.1-El sistema determina la ubicación del dispositivo móvil.	2.2 - No se puede determinar la posición, por falta de señal.
3- Se muestra la posición obtenida, en la interfase del dispositivo móvil.		

Curso de Error	<ul style="list-style-type: none"> - No funciona el dispositivo móvil del usuario. - No hay señal y se debe volver a encender el dispositivo móvil, volviendo al paso1.
Observaciones	La respuesta dada por el sistema será la posición más cercana a la ubicación real del dispositivo móvil.

En este caso se supone que el usuario consta de un dispositivo móvil provisto de un sistema de Posicionamiento, lo enciende e ingresa al sistema de búsqueda de caminos. En ese momento el sistema determina automáticamente la posición actual de dispositivo, se supone que en el dispositivo se muestra un grafico de un mapa donde se visualiza la ubicación actual. Puede suceder que al momento de conectarse al sistema no haya señal suficiente como para que este pueda determinar la ubicación del dispositivo, en ese caso fallará la respuesta.

Caso de Uso: **Búsqueda de camino.**

Resumen de la identificación	
Título	Búsqueda de camino.
Descripción Contextual	El usuario selecciona un destino de una lista de destinos disponibles y solicita al Sistema el camino hacia este. El camino obtenido se visualiza en el dispositivo móvil.
Fecha de Creación	01/04/2008
Desarrollado por	Claudia Iarragorri - Leticia Durante
Prioridad	Alta
Número de Versión	1.0
Ultima Modificación	
Actores Involucrados	Usuario, Sistema de búsqueda de caminos
Relaciones	
Include	CU-Determinar Posición Actual
Descripción de recursos	
- Dispositivo móvil provisto de sistemas de posicionamiento.	
Descripción de la conversación	
Pre-Condicion	<ul style="list-style-type: none"> - El Sistema dispone de un conjunto de destinos. - El Sistema dispone de la información de la ubicación del dispositivo móvil.

6. Funcionalidad del Sistema

Post-Condiciones	Se muestra el camino sugerido hacia el destino en el dispositivo móvil. El sistema esta listo para una nueva búsqueda de camino.	
Acciones Del Actor	Respuesta de Sistema	
	Curso Normal	Curso Alternativo
1- El usuario selecciona un destino.		
2- El usuario solicita la búsqueda de un camino hacia el destino.	2.1-El sistema determina un camino sugerido hasta el destino solicitado.	2.2 - No se puede determinar un camino hacia el destino.
3- Se muestra el camino sugerido, en la interfase del dispositivo móvil.		
Curso de Error	<ul style="list-style-type: none"> - No funciona el dispositivo móvil del usuario. - No hay señal y se debe volver a encender el dispositivo móvil, volviendo al paso1. 	
Observaciones	La respuesta dada por el sistema será el primer camino encontrado hacia el destino.	

En este caso se supone al igual que en el caso anterior que el usuario consta de un dispositivo móvil provisto de un sistema de Posicionamiento, lo enciende e ingresa al sistema de búsqueda de caminos. En ese momento el sistema determina automáticamente la posición actual del dispositivo, se supone que en el dispositivo se muestra un grafico de un mapa donde se visualiza la ubicación actual. En la misma pantalla se mostrará una lista de los posibles puntos de interés disponibles para el usuario. Este seleccionará un lugar de dicha lista, el sistema le sugerirá un camino a seguir para alcanzar el destino deseado. Si el sistema logra determinar un camino lo mostrará en el mapa del dispositivo, de lo contrario se visualizará un mensaje de camino no encontrado.

Puede suceder que al momento de conectarse al sistema no haya señal suficiente como para que este pueda determinar la ubicación del dispositivo, con lo cual se deberá apagar y volver a encender el mismo, comenzado la operación nuevamente. También puede fallar el dispositivo móvil del usuario, con lo cual no se podrá ingresar al sistema.

La incorporación de estas conversaciones, tienen como objetivo brindar al lector un ejemplo concreto de una aplicación que utilice la implementación propuesta en el presente trabajo.

6.2 Diagramas de Secuencia

Para el análisis de la búsqueda de caminos, partimos del planteo de un conjunto de escenarios que pueden presentarse en la vida real, en función de los cuales fundamentamos la implementación. Dichos escenarios se basan en establecer un camino cuando:

- a. el usuario se encuentra posicionado en un punto de acceso del destino.
- b. el usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIMovilidad”, y existe un camino directo entre ellas en la movilidad actual.
- c. el usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIMovilidad”, y existe un camino entre ellas a través de una movilidad distinta a la actual.
- d. el usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIMovilidad”, y no existe un camino posible entre ellas en ninguna de las movilidades disponibles. Para hallar el camino es necesario pasar por puntos conexión.
- e. el usuario y los punto de acceso al destino se encuentran en unidades de desplazamiento que no hacen referencia a la misma instancia de la clase “PIMovilidad”. Para hallar el camino es necesario pasar por puntos conexión, que pueden ser tanto indoor como outdoor.

Como parte de nuestra implementación incluimos al Sistema como la clase que contiene al usuario, los puntos de conexión, los PIMovilidad y todas las movilidades representadas en el modelo. La clase “Sistema” representa el único punto de acceso a la Aplicación. Es decir es la clase desde donde se invocan los métodos para poder establecer un camino, respetando el patrón de diseño Facade [2].

Como primera parte del análisis de los escenarios planteados realizamos los diagramas de secuencia, para establecer como se van a comunicar las distintas clases que componen nuestro sistema para lograr la obtención de un camino entre dos puntos de interés.

Para todos los escenarios planteados supondremos que el usuario cuenta en su dispositivo móvil con una interfase que, entre otras cosas, le brinda la posibilidad de seleccionar un Punto de Interés determinado, entre un conjunto de Puntos de interés disponibles y de buscar un camino desde su ubicación hasta el destino seleccionado. Una vez determinado éste, el usuario solicitará al Sistema la búsqueda de un camino desde donde se encuentra posicionado hacia el destino elegido.

Cuando el usuario se conecta al Sistema, éste determinará la ubicación actual del mismo.

A continuación se explican cada uno de los escenarios junto con el diagrama de Secuencia correspondiente.

Escenario a

En este escenario se asume que el usuario, con su dispositivo móvil, se encuentra posicionado en un punto de acceso del destino. Recordemos que el usuario se traslada únicamente por unidades de desplazamiento y que un punto de acceso relaciona una instancia de la clase “PIConMovilidadInterna“ con una instancia de la clase “UnidadDeDesplazamiento“ . Por lo tanto, en este caso, estamos suponiendo que el usuario se encuentra posicionado en una UnidadDeDesplazamiento que forma parte del PuntoDeAcceso, el cual relaciona esa unidad de desplazamiento con el PIMovilidad, que representa al destino seleccionado inicialmente por el Usuario.

En la Figura 6.2.1 se visualiza un ejemplo que se corresponde con lo explicado anteriormente, en el cual el florero es el destino solicitado por el usuario.

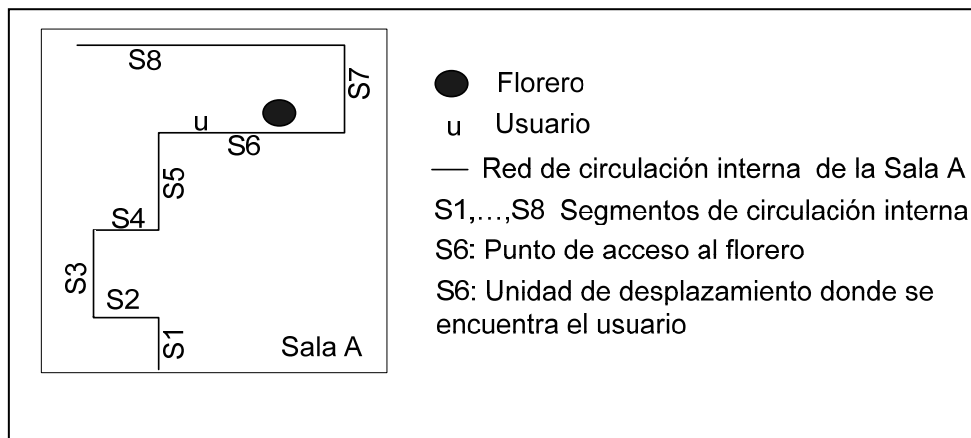


Figura 6.2.1 – El usuario se encuentra en un punto de acceso del destino

Como se ve en la Figura 6.2.1, el usuario, que se representa en la figura como **u**, se encuentra posicionado en el segmento de circulación interna llamado **S6**. Este mismo segmento forma parte de un punto de acceso del florero.

Al solicitar al Sistema la búsqueda de un camino hacia un PIMovilidad, éste delega el requerimiento a la clase Usuario. En esta clase se obtiene la instancia de PIMovilidad donde se encuentra posicionado el usuario, y se delega a ésta el requerimiento. La clase “PIMovilidad” tiene la responsabilidad de crear la instancia de la clase “Camino”, que será devuelta al dispositivo móvil. Como primer paso, se determina si el usuario se encuentra en algún punto de acceso del destino. Esto se cumple para el escenario planteado, con lo cual, el usuario ya encontró el destino solicitado. Por último se agrega al camino un único tramo que hace referencia al punto de acceso del destino.

El diagrama de secuencia que se corresponde con este escenario se observa en la figura 6.2.2.

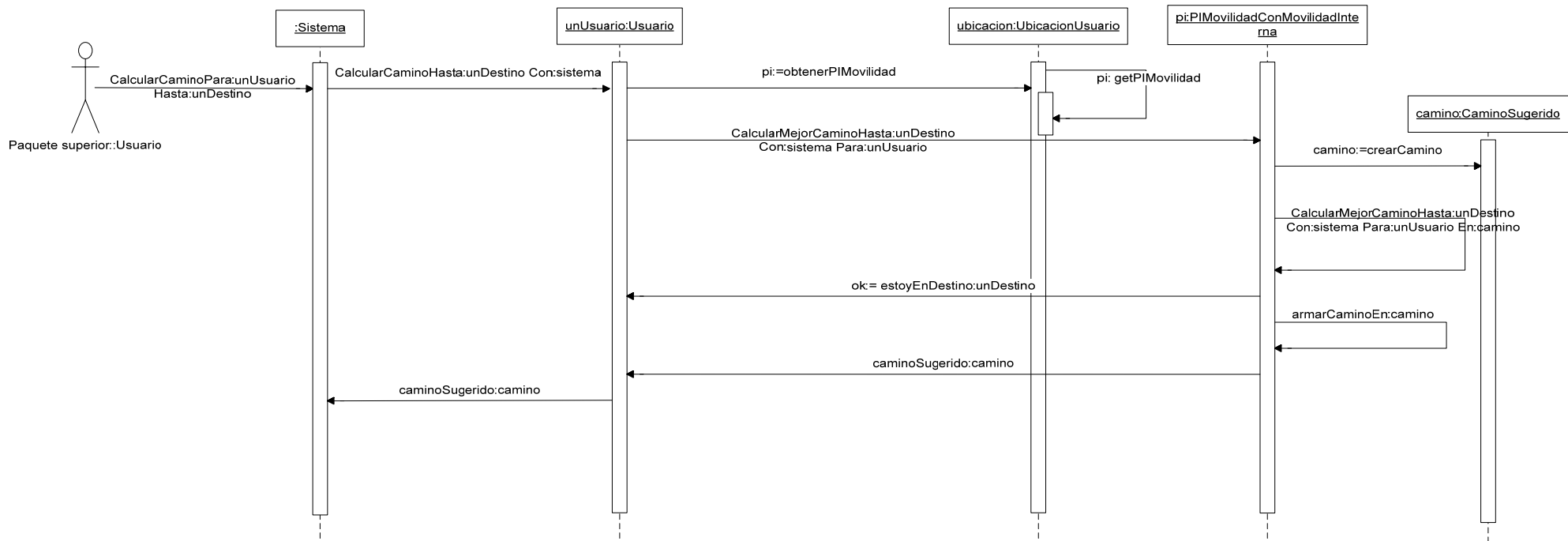


Figura 6.2.2 - Diagrama de Secuencia del escenario a.

A continuación, se detallan los métodos que se utilizan para resolver la búsqueda del camino en el escenario a.

La búsqueda de un camino se inicia en el Sistema mediante el siguiente método:

Sistema>> calcularMeyorCaminoPara: unUsuario hasta: unDestino

El Sistema, a través de este método, delega la responsabilidad a la instancia de la clase Usuario, la cual es el parámetro llamado unUsuario, invocando al siguiente método:

Usuario>> calcularMeyorCaminoHasta: unDestino Con: sistema

La clase Usuario delega el método a la clase UbicaciónUsuario, debido a que es ésta la que cuenta con la información de dónde se encuentra posicionado el usuario móvil y del PIMovilidad al que hace referencia.

UbicacionUsuario>> calcularMeyorCaminoHasta: unDestino Con: sistema Para: unUsuario

La clase UbicacionUsuario delega el método a la instancia de la clase PIMovilidad. La clase “PIMovilidad” conoce a todos sus puntos de acceso y a las distintas movilidades con las que puede ser recorrido.

PIconMovilidadInterna>>

calcularMeyorCaminoHasta: unDestino Con: sistema Para: uniusuario

En esta clase se instancia al camino estableciéndose ésta como origen, y al destino solicitado como destino. Una vez instanciado el camino con su origen y destino se realiza una llamada al método:

PIConMovilidadInterna>>

calcularMejorCaminoHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

Es en este método donde se determina si el usuario se encuentra posicionado en un punto de acceso del destino. La búsqueda del camino, para este escenario, finaliza en este método.

Como veremos en los próximos escenarios, hasta aquí, la secuencia de métodos es la misma. Es en este punto donde se determinan los pasos a seguir para cada tipo de escenario.

Escenario b:

Para explicar este escenario se asume que el Usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIConMovilidadInterna”, y que existe un camino directo en la movilidad actual del usuario, es decir, que existe un punto de acceso del destino con esta movilidad.

En la figura 6.2.3, se observa un ejemplo de este escenario.

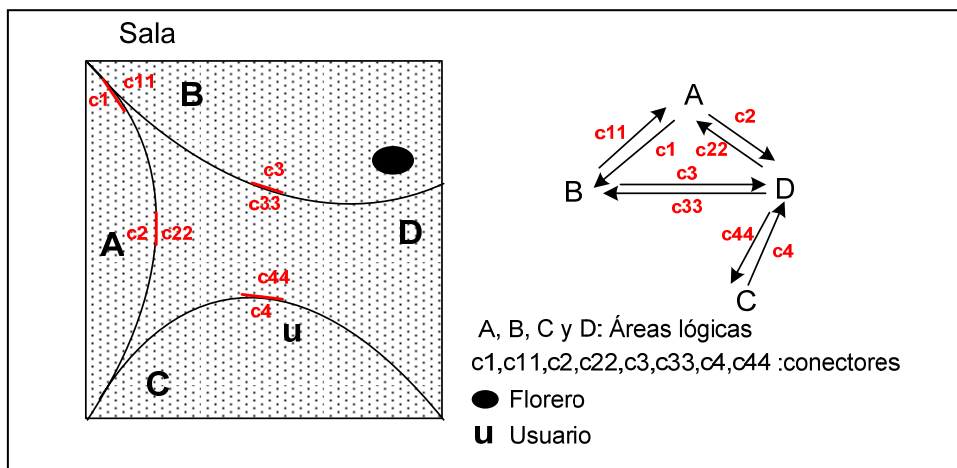


Figura 6.2.3 -El usuario y el destino se encuentran en el mismo PIMovilidad

En la Figura 6.2.3, se observa al usuario representado como **u**, que se encuentra posicionado en el conector denominado **c4** del área lógica **C**. El usuario y el destino se encuentran el en mismo PIConMovilidadInterna **Sala**, con la movilidad Transitar Áreas

Lógicas. Como se observa en el grafo de circulación, existe un camino directo hacia el destino dentro de la Sala para la movilidad actual.

En la Figura 6.2.4, se muestra el diagrama de secuencia para la búsqueda de un camino en este escenario.

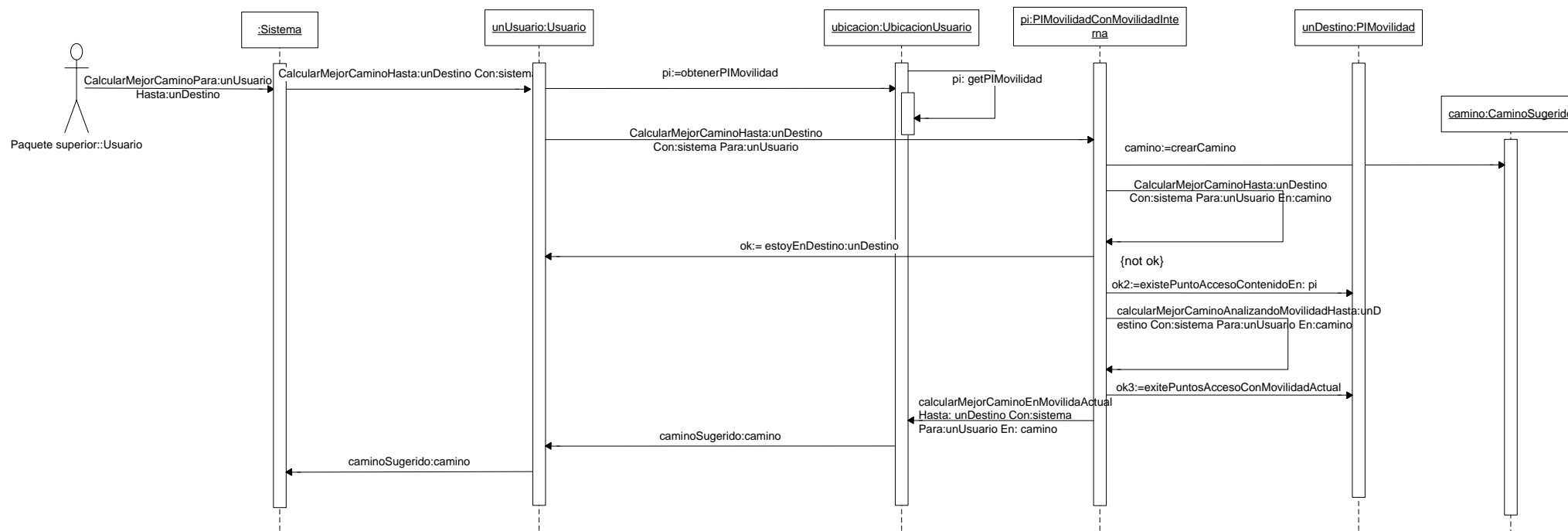


Figura 6.2.4 - Diagrama de Secuencia del escenario b.

Como explicamos anteriormente, la secuencia de métodos hasta la clase `PIConMovilidadInterna`, es la misma para todos los escenarios, por lo tanto, partiremos explicando la secuencia de métodos desde el siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

En este método se determina si existe un punto de acceso del destino contenido en el `PIMovilidad` del usuario. Como esto se cumple para este escenario, se busca el camino analizando las movilidades asociadas al `PIMovilidad` invocando al método.

PIConMovilidadInterna>>

calcularMejorCaminoAnalizandoMovilidadHasta: unDestino **Con:** sistema
Para: unUsuario **En:** camino.

Recordemos que un `PIConMovilidadInterna` tiene asociado un conjunto de movilidades, que representan las distintas formas en que éste puede ser recorrido. En la implementación se asume que la primera movilidad de este conjunto representa la movilidad actual.

En este caso se invoca al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoPorMovilidadHasta: unDestino **Con:** sistema
Para: unUsuario **En:** camino.

En este método se obtienen los puntos de acceso del destino con la movilidad actual, y para cada punto de acceso se obtiene la unidad de desplazamiento asociada. Además se determina la unidad de desplazamiento asociada a la ubicación de usuario, y se delega a ésta el método:

UnidadDesplazamiento>>

calcularMejorCaminoPorMovilidad: m **Hasta:** udd **Con:** sistema **Para:** unUsuario **En:** camino.

En donde **m** es la movilidad actual y **udd** es la unidad de desplazamiento asociada a un punto de acceso del destino.

En este método se busca el camino en el grafo de circulación que representa a la movilidad actual, utilizando la estrategia de búsqueda A*.

Escenario c.

En este escenario se asume que el Usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIconMovilidadInterna“, y que existe un camino directo en alguna movilidad distinta a la actual.

En la Figura 6.2.5 se visualiza un ejemplo del escenario c, en este ejemplo se muestra una Sala incluida en un museo la cual tiene dos tipos de movilidades.

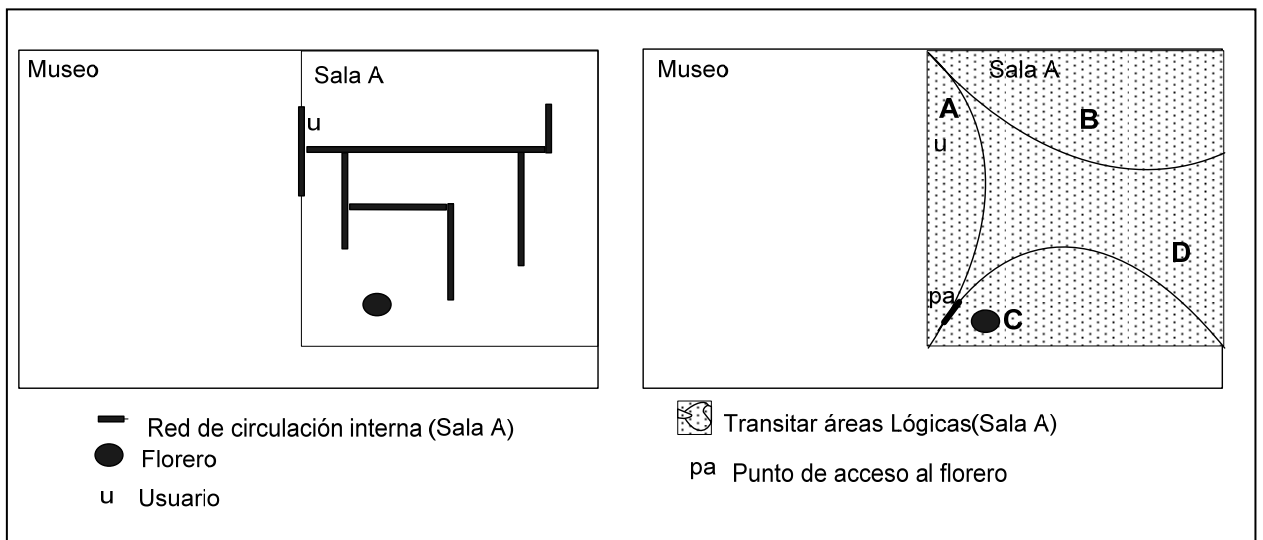


Figura 6.2.5 - El punto de acceso al destino no pertenece a la movilidad actual del Usuario.

En la Figura 6.2.5, se puede observar que, para el caso de la sala con la movilidad “RedCirculacionInterna”, no existe ningún punto de acceso al Florero para esta movilidad. En el otro caso, para la movilidad “TransitarAreasLogicas”, existe un punto de acceso al florero, con la cual se puede calcular un camino al mismo.

En es escenario, donde es necesario cambiar de movilidad, se debe reposicionar al usuario en una unidad de desplazamiento perteneciente a la nueva movilidad.

En la figura 6.2.6 se observa el diagrama de secuencia correspondiente.

6. Funcionalidad del Sistema

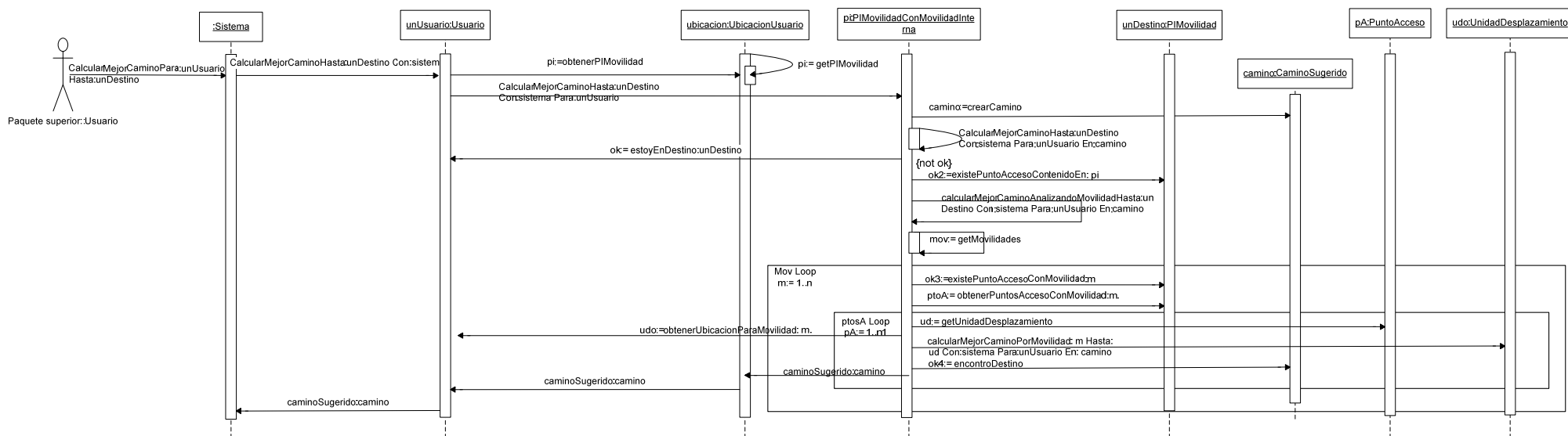


Figura 6.2.6 - Diagrama de Secuencia del escenario c.

En este caso también partiremos explicando la secuencia de métodos desde el siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

En este método se determina si existe un punto de acceso del destino contenido en el PIMovilidad del usuario. Como esto se cumple para este escenario, se busca el camino analizando las movilidades asociadas al PIMovilidad invocando al método.

PIConMovilidadInterna>>

calcularMejorCaminoAnalizandoMovilidadHasta: unDestino **Con:** sistema
Para: unUsuario **En:** camino.

En este caso, como el usuario y el destino se encuentran en el mismo PIConMovilidadInterna se invoca al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoPorMovilidadHasta: unDestino **Con:** sistema
Para: unUsuario **En:** camino.

En este método se recorre el conjunto de movilidades del PIMovilidad. Para cada movilidad se obtienen los puntos de acceso del destino en la misma, y la unidad de desplazamiento asociada a estos. Además se determina la unidad de desplazamiento asociada a la ubicación de usuario correspondiente a esa movilidad, y se delega a ésta el método:

UnidadDesplazamiento>>

calcularMejorCaminoPorMovilidad: m **Hasta:** udd **Con:** sistema **Para:** unUsuario **En:** camino.

En donde **m** es la movilidad actual y **udd** es la unidad de desplazamiento asociada a un punto de acceso del destino.

En este método se busca el camino en el grafo de circulación que representa a esa movilidad, utilizando la estrategia de búsqueda A*.

Este método será invocado con cada una de las movilidades correspondientes al PIMovilidad hasta encontrar un camino.

Escenarios d.

En este escenario, para hallar un camino, es necesario que el usuario deba pasar por puntos conexión, debido a que, el usuario y algún punto de acceso al destino se encuentran en unidades de desplazamiento que hacen referencia a la misma instancia de la clase “PIMovilidad”, pero no existe un camino posible para ninguna de las movilidades disponibles. Un ejemplo de esta situación se puede observar en la Figura 6.2.7.

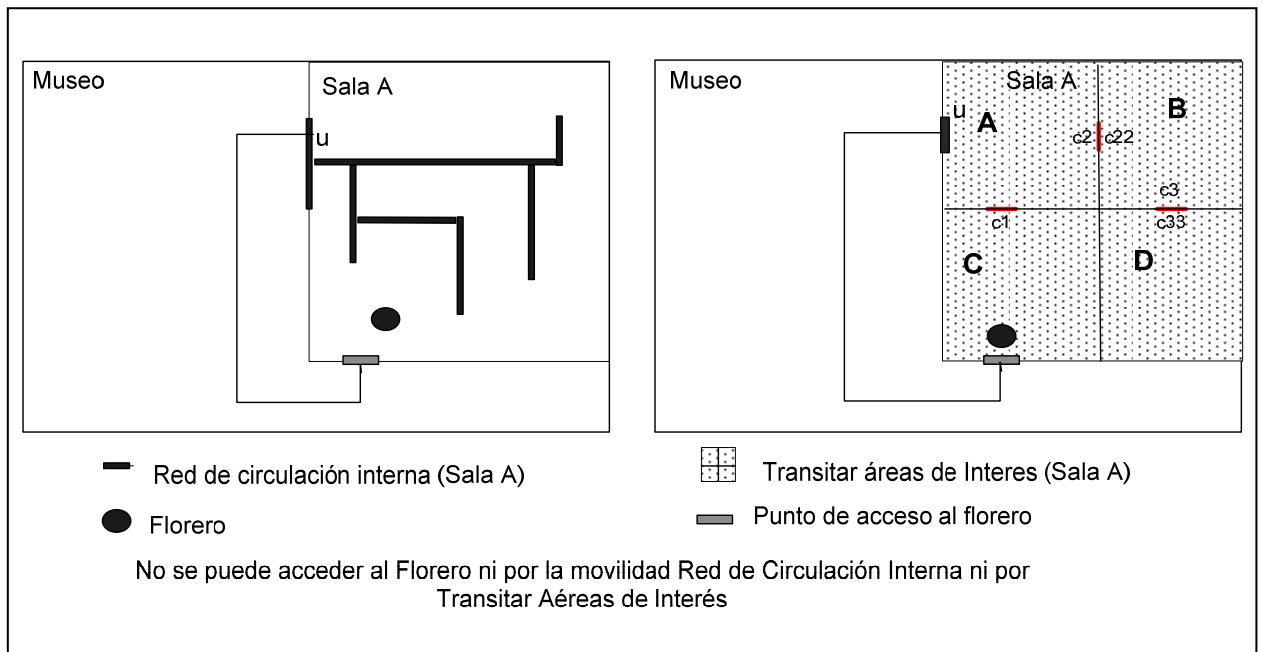


Figura 6.2.7 – No existe un camino posible para ninguna de las movilidades disponibles.

En la Figura 6.2.7, el único punto de acceso al Florero se encuentra en el área de Interés llamada **C**, que pertenece a la movilidad “TransitarAreasInteres”. Este punto de acceso no puede ser accedido desde la ubicación, representada como **u**, donde se encuentra el usuario. Por lo tanto, el usuario deberá primero salir de la sala, y luego buscar el camino hacia el mencionado punto de acceso.

En la Figura 6.2.9 se observa el diagrama de secuencia de este escenario.

6. Funcionalidad del Sistema

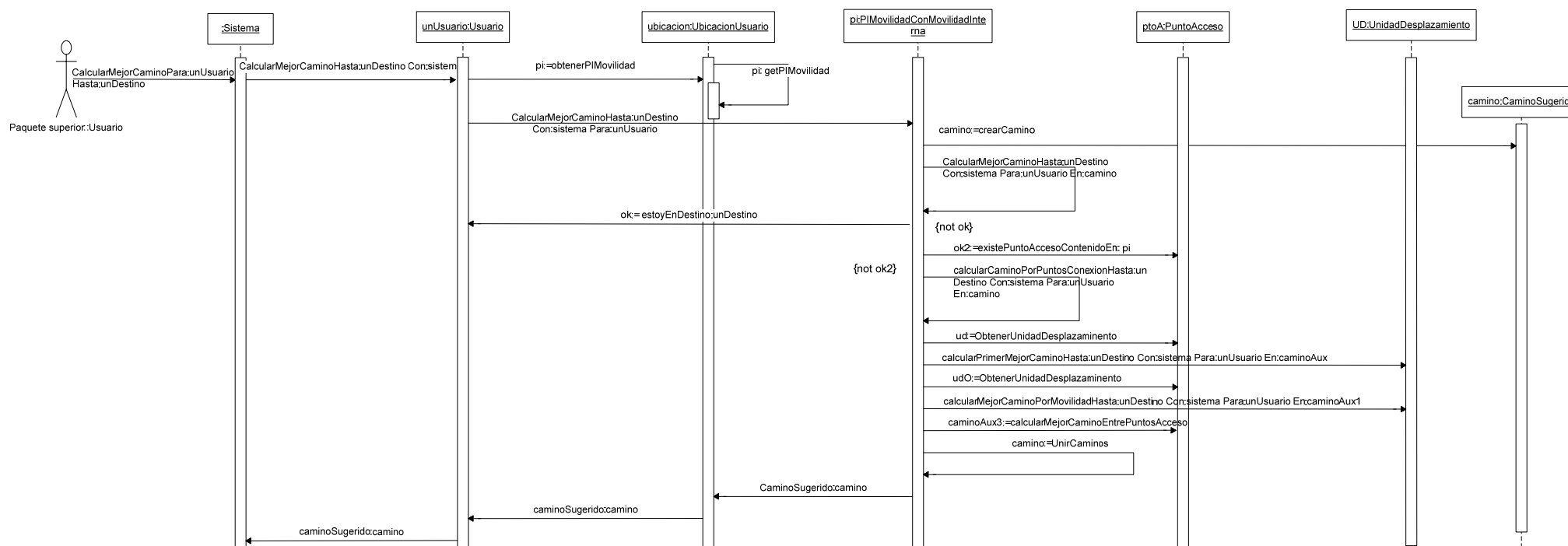


Figura 6.2.9 - Diagrama de Secuencia del escenario d.

En este caso también partiremos explicando la secuencia de métodos desde el siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

En este método, se determina que existe un punto de acceso del destino contenido en el PIMovilidad del usuario, con lo cual se invoca al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoAnalizandoMovilidadHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

Este método, como primer paso invoca al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoPorMovilidadHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

En este método se recorre el conjunto de movilidades del PIMovilidad y se determina que no existe un camino hacia el destino para ninguna de ellas.

Por lo tanto, al no existir un camino, como segundo paso, se invoca al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoPorPuntosConexionHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino

Escenario e.

En este escenario, para hallar un camino, es necesario que el usuario deba pasar por puntos conexión, debido a que, el usuario y los punto de acceso al destino no se encuentran en unidades de desplazamiento que hagan referencia a la misma instancia de la clase “PIMovilidad”.

Esta situación se puede observar en la Figura 6.2.8.

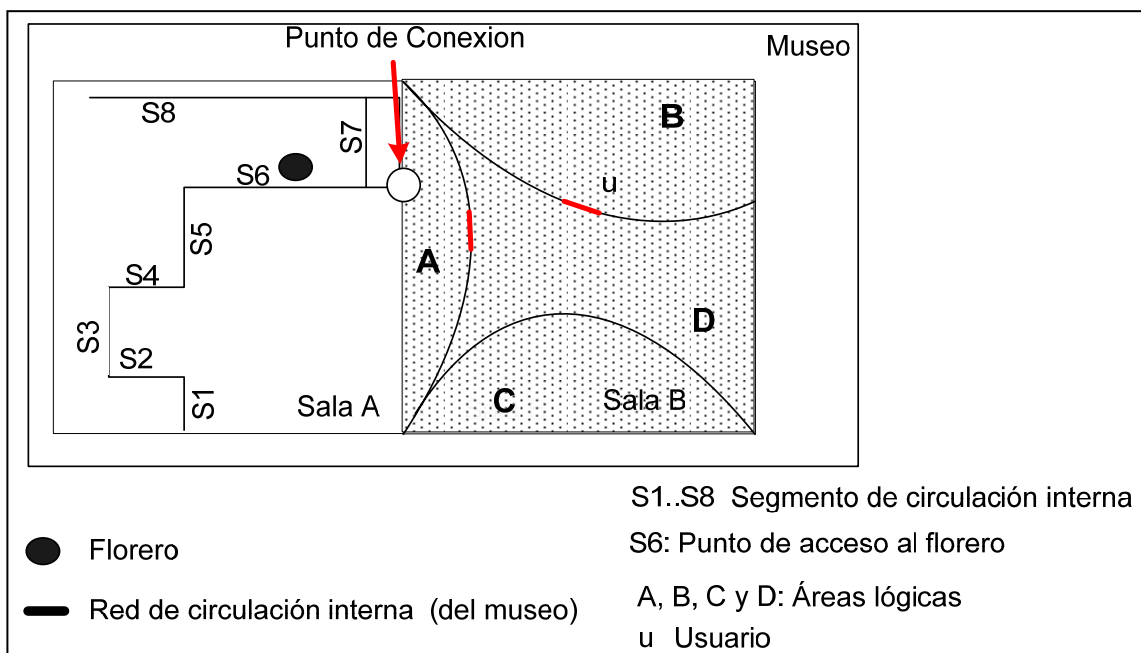


Figura 6.2.8 – Para llegar al Destino se debe pasar por Puntos Conexión.

En la Figura 6.2.8, supongamos que el usuario se encuentra en la Sala B, y el Florero en la Sala A, por lo que necesariamente el usuario deberá trasladarse de una sala a otra. Para lograr esto, el camino estará formado por la porción del camino incluido en la Sala B, con tramos pertenecientes a la movilidad “TransitarAreasLogicas”, el tramo conexión que vincule ambas salas y la porción del camino correspondiente a la Sala A, con tramos pertenecientes a la movilidad “RedCirculacionInterna”.

En la Figura 6.2.10 se observa el diagrama de secuencia de este escenario.

6. Funcionalidad del Sistema

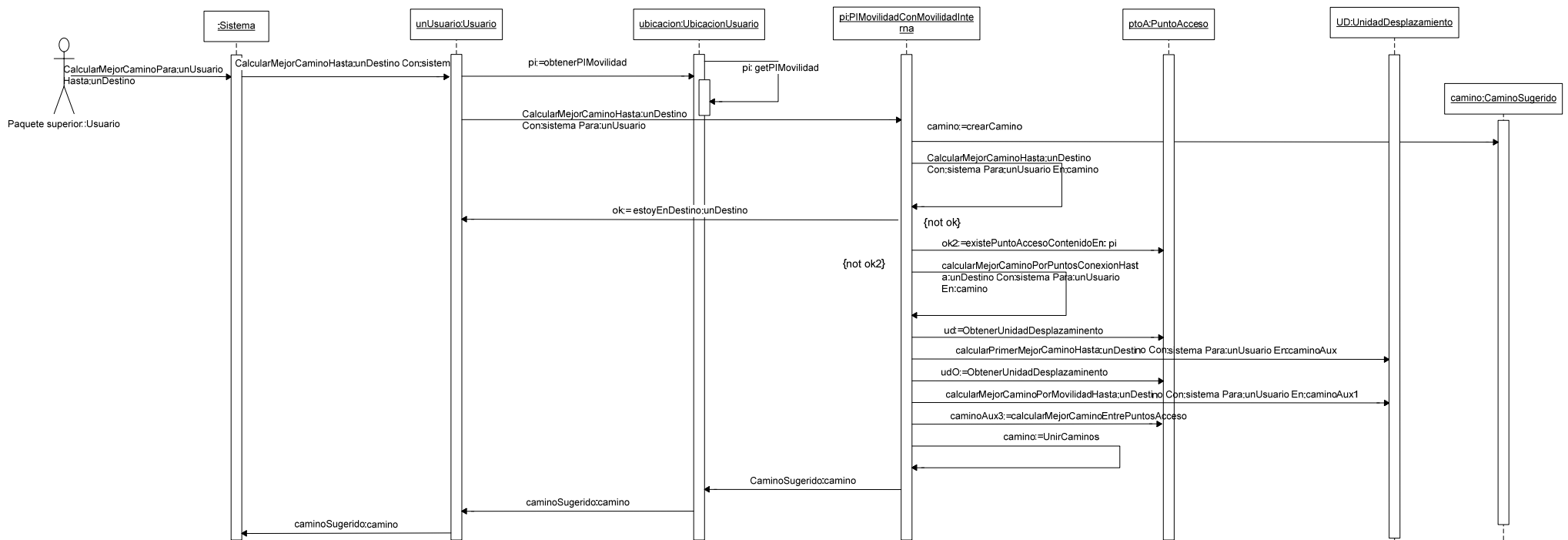


Figura 6.2.10 - Diagrama de Secuencia del escenario e.

En este caso también partiremos explicando la secuencia de métodos desde el siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoHasta: unDestino **Con:** sistema **Para:** unUsuario **En:** camino.

Como el usuario y el destino están en distintos PIMovilidad, se invoca directamente al siguiente método:

PIConMovilidadInterna>>

calcularMejorCaminoPorPuntosConexionHasta: unDestino **Con:** sistema **Para:** unUsuario
En: camino

En este caso el camino se arma en tres etapas:

En la primera etapa, se arman todos los caminos posibles desde la posición del usuario hasta los puntos de acceso del PIMovilidad donde se encuentra, contemplando los escenarios anteriores. Dicho punto de acceso pertenece a un punto conexión.

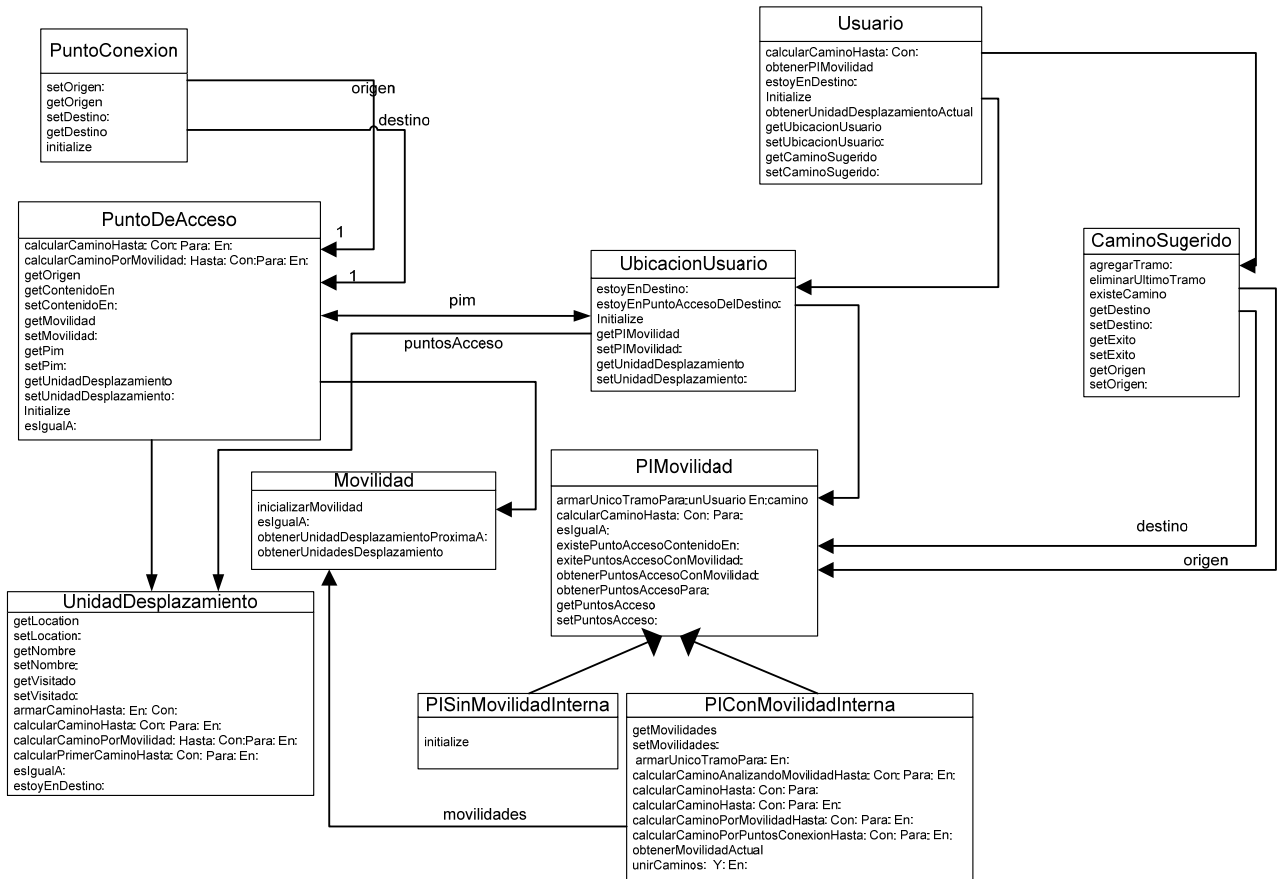
En la segunda etapa se arman todos los caminos posibles desde el destino hasta los puntos de acceso del PIMovilidad que lo contiene, los cuales pertenecen a un punto conexión.

Por último se recorren los conjuntos y se determina si es posible unir estos dos caminos, contemplando todos los escenarios.

6.3 Diagrama de Clase

A continuación en la Figura 6.3 mostraremos el diagrama de clase junto con su comportamiento.

6. Funcionalidad del Sistema



6. Funcionalidad del Sistema

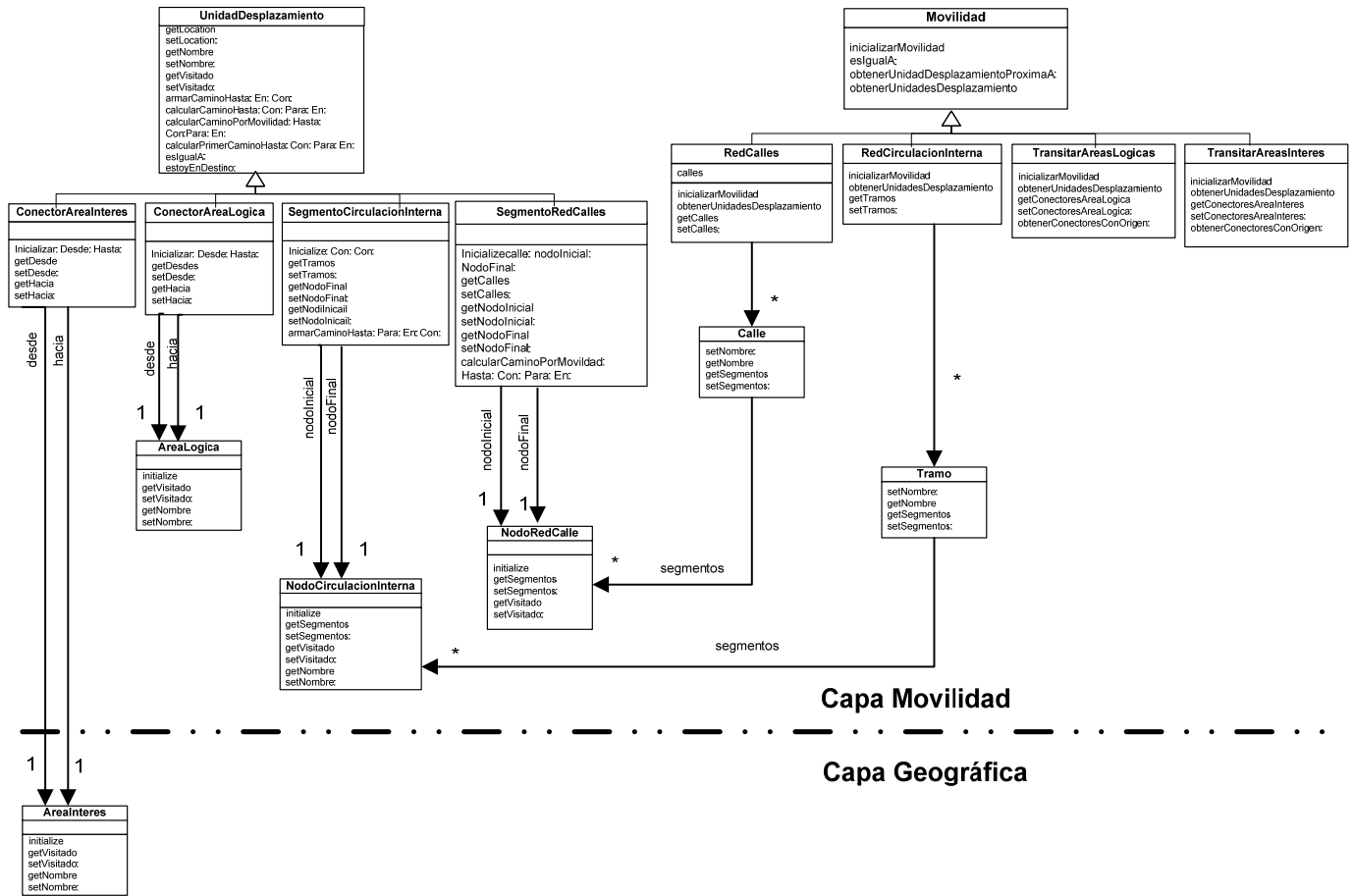


Figura 6.3 – Diagrama de clase

En la Figura 6.3 se muestra el comportamiento entre las clases involucradas para la implementación del presente trabajo.

Capítulo 7

En este capítulo se comentan las conclusiones que surgen del presente trabajo y los trabajos futuros que se podrían desprender del mismo.

7.1 Conclusiones

El modelo resultante brinda la posibilidad de proveer movilidad independientemente del escenario involucrado, es decir, si se trata de un escenario Indoor, Outdoor, o Indoor-Outdoor.

El modelo además da la posibilidad de incorporar nuevos tipos de movilidades que permitan recorrer, de alguna otra forma, una instancia de la clase “PIMovilidad” (explicada en el capítulo 5), gracias al desacoplamiento de las estrategias de circulación representada por la jerarquía de la clase “Movilidad”. En el modelo se plantea que un mismo “PIMovilidad” puede estar asociado a más de un tipo movilidad, es decir, puede ser recorrido utilizando distintas estrategias. En la implementación desarrollada para la búsqueda de un camino, el cambio de estrategia, esta contemplado de forma tal que este se realiza automáticamente hasta lograr, de ser posible, alcanzar el destino solicitado por el usuario.

Hemos establecido dos tipos de algoritmos de búsqueda para implementar el recorrido de un camino. Por un lado hemos desarrollado el algoritmo Breadth First Search (BFS), el cuál devuelve el primer camino encontrado entre el origen y el destino. Por otro lado, hemos implementado el algoritmo de búsqueda A*, el cuál retorna el camino más corto según la heurística establecida. En el presente trabajo, tomamos como función heurística, la cantidad de unidades de desplazamiento que involucra el recorrido.

7.2 Trabajos Futuros

Las estrategias de búsqueda (BFS, A*) que se han desarrollado en el presente trabajo, fueron utilizadas como una manera de mostrar la factibilidad de la implementación del modelo.

A continuación plantearemos posibles extensiones para este trabajo:

- Establecer distintas heurísticas para el algoritmo de búsqueda A*, como por ejemplo, el recorrido con menor cantidad de semáforos, el más económico o el más corto. También se podrían incorporar nuevos algoritmo de búsqueda. De esta forma se le brindaría al usuario la posibilidad de optar entre distintas alternativas de acuerdo a sus preferencias.
- Incorporar al desarrollo del Sistema el contexto del usuario, el cual entre otras cosas, podría establecer sus gustos o determinar los lugares a los que éste tiene acceso. Se podría tener en cuenta el historial de los caminos realizados por el usuario de forma tal de poder sugerirle recorridos que sean de su interés.
- Puede ocurrir que, en un determinado momento, ante situaciones imprevistas, como por ejemplo una calle cortada, un accidente de tránsito, un edificio clausurado, no se pueda transitar por el camino propuesto. Para que el sistema sea capaz de manejar este tipo de situaciones, se podría recurrir a la programación Orientada a Aspectos [38]. Por ejemplo, se podría implementar un Aspecto que tenga en cuenta situaciones de estas características al momento de sugerir un camino al usuario.

Las alternativas propuestas anteriormente no han sido incluidas en el presente trabajo, pero otorgaría valor agregado a la hora de desarrollar una aplicación específica.

Referencias bibliográficas

- [1] “Movilidades indoor-outdoor para aplicaciones GIS” - Challiol Cecilia, Lliteras Alejandra - 2007.
- [2] “Design Patterns. Elements of reusable object-oriented software”. E. Gamma., R. Helm, R. Johnson, J. Vlissides - Addison Wesley 1995.
- [3] Generating the architecture of GIS applications with design patterns, In Proceedings of the GIS '97, S. Gordillo, F. Balaguer and F. Das Neves - ACM Press, 1997, pp. 30 - 34.
- [4] Location-based Services: Fundamentals and Operation Axel Kupper - John Wiley & Sons, Ltd 2005.
- [5] Geographic Information System. A Management Perspective. Stan Aronoff – WDL Publications 1989.
- [6] “Design and implementation of an indoor mobile navigation system”, Master's thesis, Massachusetts Institute of Technology - A.K.L. Miu, 2002.
- [7] SISTEMAS DE REFERENCIA Y PROYECCIONES CARTOGRÁFICAS- Rubén C. Rodríguez - XXI Reunión Científica AAGG Rosario (Argentina) – 23 al 27 de Septiembre de 2002.
- [8] Internet GIS Distributed Geographic Information Services for the Internet and Wireless Networks. Zhong – Ren Peng y Ming- hsiang Tsou. 2003.
- [9] An Introducción to Spatial Database System. VLDB Journal (Vol. 3)- Güting, Ralf G Octubre 1994.
- [10] Spatial Databases with Application to GIS - Rigaux, P., Scholl M. &, Voisard, A., Editorial Morgan Kaufmann Publishers. 2002.
- [11] Spatial Data Structures. Addison - Samet Hanan, Wesley /ACM Press, MA, 1995.
- [12] “Sistemas de Información Geográfica”, Joaquín Bosque Sendra.
- [13] Los Sistemas de Posicionamiento por Satélite y sus aplicaciones. José Francisco Saldaña Hernández, Instituto de Información Territorial del Estado de Jalisco
- [14] Location- Based services, Jochen Schiller, Agnes Voisard. 2004

-
- [15] <http://www.fcagr.unr.edu.ar/mdt/GTS/Zonaedu/GPS3.htm>
- [16] http://www.mobilein.com/location_based_services.htm
- [17] <http://www.esri.com>
- [18] www.gis.com
- [19] <http://www.gisits.com/sig/avl.html#1>
- [20] <http://www.alegsa.com.ar>.
- [21] <http://www.xataka.com/2005/06/22-wps-un-sistema-de-posicionamiento-por-wi-fi>
- [22] <http://www.sigagropecuario.gov.ar/>
- [23] http://www.udistrital.edu.co/comunidad/profesores/rfranco/vector_raster.htm
- [24] A Principles of Geographical Information Systems for land resources assessment - BURROUGH, P. New York Oxford University Press. 1986.
- [25] "Information Technology in Geography and planning. Including principles of G.I.S" London: Routledge p. 31 - Bracken y Webster 1900.
- [26] Managing Geographic Information System Projects - Huxhold, W. E., and Levinsohn, A. G. - New York, Oxford: Oxford University Press 1995.
- [27] "Visualización Geográfica y nuevas Cartografías", GeoFocus (Artículos), nº 2, p. 61-77. - Bosque Sendra, J. y Zamora Ludovic, H. 2002.
- [28] Refining an object-oriented GIS design model: Topologies and Field Data. - Silvia Gordillo and Federico Balanguer. LIFIA- Departamento de Informática Fac. Ciencias Exactas-UNLP.
- [28] Tecnología de los Sistemas de Información Geográfica. Ed. RA-MA. Madrid.- MOLDES TEO, Javier. 1995.
- [29] Geographic Information Systems – Longley, Goodchild , Maguire, Rhind. 1999.
- [30] <http://www.geogra.uah.es>
- [31] Bases de Datos Espaciales. - CÉSPEDES, Claudia; MUÑOZ, Álvaro; ORELLANA, Cristian; PÉREZ, Claudio. - Universidad de Concepción. Chile. 2002
- [32] <http://www.geotecnologias.com/Documentos/GIS.pdf>
- [33] PROYECCIONES CARTOGRÁFICAS Y DATUM ¿Qué son y para qué sirven? - Jorge Fallas Laboratorio de Teledetección y Sistemas de Información Geográfica PRMVS-EDECA. 2003
- [34] Sistema de Referencia - Emilio Benitez Aguado Y Luis García-Asenjo Villamayor - Laboratorio de Geodesia

- [35] Localizaciones Geográficas. Las Coordenadas Geográficas y las Proyecciones UTM - Ignacio Alonso Fernández Coppel - Universidad de Valladolid. 2001
- [36] Mobile Information Systems – Barbara Permici – Springer-Verlag Berlin Heidelberg 2006
- [37] Blueps: sistema de localización en interiores utilizando Bluetooth.
- [38] Aspect Oriented Analysis and Design – Clarke, Baniassad. Addison Wesley 2005.
- [39] Realm-Based Spatial Data Types: The ROSE Algebra. - R.H. Güting and M. Schneider. VLDB Journal 4, pp.100-143, 1995.
- [40] Sistemas de Información Geográfica:Revisión de su Estado Actual - Nieves R. Brisaboa, José A. Cotelo Lema, Antonio Fariña, Miguel R. Luaces, José R. R. Viqueira - Laboratorio de Bases de Datos. Facultad de de Informática. Universidad de Coruña - España.

Código de ejemplo.

En este anexo, se da el código para poder instanciar distintos ejemplos que demuestren la viabilidad de la implementación.

En primer lugar mostraremos ejemplos de recorridos en las distintas movilidades que ofrece el modelo. Luego mostraremos un ejemplo que plantea un recorrido que va desde un edificio a otro, teniendo que transitar las calles de una ciudad, es decir, un ejemplo que involucre pasar de un escenario Indoor a uno Outdoor y viceversa.

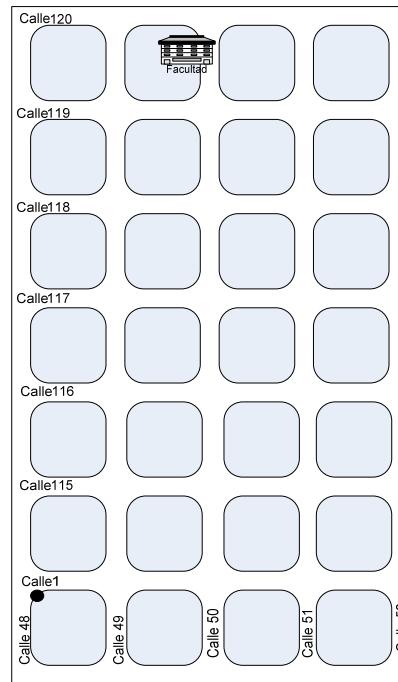
Por último mostraremos un ejemplo en el cual es necesario cambiar de movilidad en el mismo escenario para alcanzar el destino deseado.

En todos los casos para poder ejecutar el código expuesto, se deberá copiar el mismo y agregarlo en una nueva hoja del Workspace del Visual Works, una vez copiado ahí, se deberá seleccionar todo el código (CONTROL + A), y ejecutar con el botón derecho del mouse la opción de “Print it”, de esta forma se mostrará el camino sugerido para el usuario.

A. Ejemplos en las distintas movilidades.

1. Movilidad Red de Calles.

Para mostrar esta movilidad supondremos al usuario ubicado en calle 1 y 48 el cuál quiere llegar hasta la Facultad de Informática ubicada en calle 50 y 120.



```
s := Sistema new.
ciudad := PIconMovilidadInterna new.
facultad:= PISinMovilidadInterna new.
facultad setDescripcion:'facultad'.
```

"Armamos calle50"

```
c50 := Calle new.
c50 initialize: 'calle 50'.
```

```
n1:= NodoRedCalles new.
n1 initialize:'n1'.
```

```
n2:= NodoRedCalles new.
n2 initialize:'n2'.
```

```
n3:= NodoRedCalles new.
n3 initialize:'n3'.
```

```
n4:= NodoRedCalles new.
n4 initialize:'n4'.
```

```
n5:= NodoRedCalles new.
n5 initialize:'n5'.
```

```
n6:= NodoRedCalles new.
n6 initialize:'n6'.
```

```
s1 := SegmentoRedCalles new.
s1 initialize:'seg50-de-1-115' calle:c50 nodoInicial: n1 nodoFinal:n2.
```

```
s2 := SegmentoRedCalles new.  
s2 initialize:'seg50-de-115-116' calle:c50 nodoInicial: n2 nodoFinal:n3.  
  
s3 := SegmentoRedCalles new.  
s3 initialize:'seg50-de-116-117' calle:c50 nodoInicial: n3 nodoFinal:n4.  
  
s4 := SegmentoRedCalles new.  
s4 initialize:'seg50-de-117-118' calle:c50 nodoInicial: n4 nodoFinal:n5.  
  
s5 := SegmentoRedCalles new.  
s5 initialize:'seg50-de-118-119' calle:c50 nodoInicial: n5 nodoFinal:n6.
```

```
"agregamos los segmentos para cada nodo"
```

```
(n1 getSegmentos) add:s1.  
(n2 getSegmentos) add:s1.  
(n2 getSegmentos) add:s2.  
(n3 getSegmentos) add:s2.  
(n3 getSegmentos) add:s3.  
(n4 getSegmentos) add:s3.  
(n4 getSegmentos) add:s4.  
(n5 getSegmentos) add:s4.  
(n5 getSegmentos) add:s5.  
(n6 getSegmentos) add:s5.
```

```
segmentos := OrderedCollection new.  
segmentos add: s1.  
segmentos add: s2.  
segmentos add: s3.  
segmentos add:s4.  
segmentos add:s5.
```

```
c50 setSegmentos:segmentos.
```

```
"Armamos calle 1"
```

```
c1 := Calle new.  
c1 initialize: 'calle 1'.
```

```
n11:= NodoRedCalles new.  
n11 initialize:'n11'.
```

```
n12:= NodoRedCalles new.  
n12 initialize:'n12'.
```

```
n14:= NodoRedCalles new.  
n14 initialize:'n14'.
```

```
s11 := SegmentoRedCalles new.  
s11 initialize:'seg1-de-48-49' calle:c1 nodoInicial: n11 nodoFinal:n12.
```

```
s12 := SegmentoRedCalles new.
```

s12 initialize:'seg1-de-49-50' calle:c1 nodoInicial: n12 nodoFinal:n1.

s13 := SegmentoRedCalles new.

s13 initialize:'seg1-de-50-51' calle:c1 nodoInicial: n1 nodoFinal:n14.

"agregamos los segmentos para cada nodo"

(n11 getSegmentos) add:s11.

(n12 getSegmentos) add:s11.

"(n12 getSegmentos) add:s12."

(n1 getSegmentos) add:s12.

(n1 getSegmentos) add:s13.

(n14 getSegmentos) add:s13.

segmentos1 := OrderedCollection new.

segmentos1 add: s11.

segmentos1 add: s12.

segmentos1 add: s13.

c1 setSegmentos:segmentos1.

"Armamos calle48"

c48 := Calle new.

c48 initialize: 'calle 48'.

n482:= NodoRedCalles new.

n482 initialize:'n482'.

n483:= NodoRedCalles new.

n483 initialize:'n483'.

n484:= NodoRedCalles new.

n484 initialize:'n484'.

n485:= NodoRedCalles new.

n485 initialize:'n485'.

n486:= NodoRedCalles new.

n486 initialize:'n486'.

s481 := SegmentoRedCalles new.

s481 initialize:'seg48-de115-1' calle:c48 nodoInicial: n11 nodoFinal:n482.

s482 := SegmentoRedCalles new.

s482 initialize:'seg48-de-116-115' calle:c48 nodoInicial: n482 nodoFinal:n483.

s483 := SegmentoRedCalles new.

s483 initialize:'seg48-de-117-116' calle:c48 nodoInicial: n483 nodoFinal:n484.

s484 := SegmentoRedCalles new.

s484 initialize:'seg48-de-118-117' calle:c48 nodoInicial: n484 nodoFinal:n485.

s485 := SegmentoRedCalles new.
s485 initialize:'seg48-de-119-118' calle:c48 nodoInicial: n485 nodoFinal:n486.

"agregamos los segmentos para cada nodo"

(n11 getSegmentos) add:s481.
(n482 getSegmentos) add:s481.
(n482 getSegmentos) add:s482.
(n483 getSegmentos) add:s482.
(n483 getSegmentos) add:s483.
(n484 getSegmentos) add:s483.
(n484 getSegmentos) add:s484.
(n485 getSegmentos) add:s484.
(n485 getSegmentos) add:s485.
(n486 getSegmentos) add:s485.

segmentos := OrderedCollection new.
segmentos add: s481.
segmentos add: s482.
segmentos add: s483.
segmentos add:s484.
segmentos add:s485.

c48 setSegmentos:segmentos.

"Armamos calle49"

c49 := Calle new.
c49 initialize: 'calle 49'.

n492:= NodoRedCalles new.
n492 initialize:'n492'.

n493:= NodoRedCalles new.
n493 initialize:'n493'.

n494:= NodoRedCalles new.
n494 initialize:'n494'.

n495:= NodoRedCalles new.
n495 initialize:'n495'.

n496:= NodoRedCalles new.
n496 initialize:'n496'.

s491 := SegmentoRedCalles new.
s491 initialize:'seg49-de-1-115' calle:c49 nodoInicial: n12 nodoFinal:n492.

s492 := SegmentoRedCalles new.
s492 initialize:'seg49-de115-116' calle:c49 nodoInicial: n492 nodoFinal:n493.

s493 := SegmentoRedCalles new.
s493 initialize:'seg49-de-116-117' calle:c49 nodoInicial: n493 nodoFinal:n494.

s494 := SegmentoRedCalles new.
s494 initialize:'seg49-de-117-118' calle:c49 nodoInicial: n494 nodoFinal:n495.

s495 := SegmentoRedCalles new.
s495 initialize:'seg49-de-118-119' calle:c49 nodoInicial: n495 nodoFinal:n496.

"agregamos los segmentos para cada nodo"

(n12 getSegmentos) add:s491.
(n492 getSegmentos) add:s491.
(n492 getSegmentos) add:s492.
(n493 getSegmentos) add:s492.
(n493 getSegmentos) add:s493.
(n494 getSegmentos) add:s493.
(n494 getSegmentos) add:s494.
(n495 getSegmentos) add:s494.
(n495 getSegmentos) add:s495.
(n496 getSegmentos) add:s495.

segmentos := OrderedCollection new.
segmentos add: s491.
segmentos add: s492.
segmentos add: s493.
segmentos add:s494.
segmentos add:s495.

c49 setSegmentos:segmentos.

"Armamos calle 115"

c115 := Calle new.
c115 initialize: 'calle 115'.

s1151 := SegmentoRedCalles new.
s1151 initialize:'seg115-de-48-49' calle:c115 nodoInicial: n482 nodoFinal:n492.

s1152 := SegmentoRedCalles new.
s1152 initialize:'seg115-de-49-50' calle:c115 nodoInicial: n492 nodoFinal:n1.

"agregamos los segmentos para cada nodo"

(n482 getSegmentos) add:s1151.
(n492 getSegmentos) add:s1151.
(n492 getSegmentos) add:s1152.
(n1 getSegmentos) add:s1152.

segmentos115 := OrderedCollection new.

segmentos115 add: s1151.
segmentos115 add: s1152.
c115 setSegmentos:segmentos115.

"Armamos calle 116"

c116 := Calle new.
c116 initialize: 'calle 116'.

s1161 := SegmentoRedCalles new.
s1161 initialize:'seg116-de-48-49' calle:c116 nodoInicial: n483 nodoFinal:n493.

s1162 := SegmentoRedCalles new.
s1162 initialize:'seg116-de-49-50' calle:c116 nodoInicial: n493 nodoFinal:n2.

"agregamos los segmentos para cada nodo"

(n483 getSegmentos) add:s1161.
(n493 getSegmentos) add:s1161.
(n493 getSegmentos) add:s1162.
(n2 getSegmentos) add:s1162.

segmentos116 := OrderedCollection new.
segmentos116 add: s1161.
segmentos116 add: s1162.
c116 setSegmentos:segmentos116.

"Armamos calle 117"

c117 := Calle new.
c117 initialize: 'calle 117'.

s1171 := SegmentoRedCalles new.
s1171 initialize:'seg117-de-48-49' calle:c117 nodoInicial: n484 nodoFinal:n494.

s1172 := SegmentoRedCalles new.
s1172 initialize:'seg117-de-49-50' calle:c117 nodoInicial: n494 nodoFinal:n3.

"agregamos los segmentos para cada nodo"

(n484 getSegmentos) add:s1171.
(n494 getSegmentos) add:s1171.
(n494 getSegmentos) add:s1172.
(n3 getSegmentos) add:s1172.

segmentos117 := OrderedCollection new.
segmentos117 add: s1171.
segmentos117 add: s1172.
c117 setSegmentos:segmentos117.

"Armamos calle 118"

c118 := Calle new.
c118 initialize: 'calle 118'.

s1181 := SegmentoRedCalles new.
s1181 initialize:'seg118-de-48-19' calle:c118 nodoInicial: n485 nodoFinal:n495.

s1182 := SegmentoRedCalles new.
s1182 initialize:'seg118-de-49-50' calle:c118 nodoInicial: n495 nodoFinal:n4.

"Armamos calle 119"

c119 := Calle new.
c119 initialize: 'calle 119'.

s1191 := SegmentoRedCalles new.
s1191 initialize:'seg119-de48-49' calle:c119 nodoInicial: n486 nodoFinal:n496.

s1192 := SegmentoRedCalles new.
s1192 initialize:'seg119-de-49-50' calle:c119 nodoInicial: n496 nodoFinal:n5.

"agregamos los segmentos para cada nodo"

(n486 getSegmentos) add:s1191.
(n496 getSegmentos) add:s1191.
(n496 getSegmentos) add:s1192.
(n5 getSegmentos) add:s1192.

segmentos119 := OrderedCollection new.
segmentos119 add: s1191.
segmentos119 add: s1192.
c119 setSegmentos:segmentos119.
calles := OrderedCollection new.

calles add: c118.
calles add: c119.
calles add: c50.
calles add: c48.
calles add: c116.
calles add: c117.
calles add: c49.
calles add: c115.
calles add: c1.
movRC := RedDeCalles new.
movRC setCalles:calles.

“puntos de acceso”

pa1:= PuntoDeAcceso new.
pa1 setPim:ciudad.
pa1 setUnidadDesplazamiento:s5.
pa1 setMovilidad: movRC.
pa1 setContenidoEn: ciudad.

(ciudad getMovilidades) add: movRC.

(facultad getPuntosAcceso) add: pa1.

u:= Usuario new.

ubicacion := UbicacionUsuario new.

ubicacion setPIMovilidad: ciudad.

ubicacion setUnidadDesplazamiento: s11.

u setUbicacionUsuario: ubicacion.

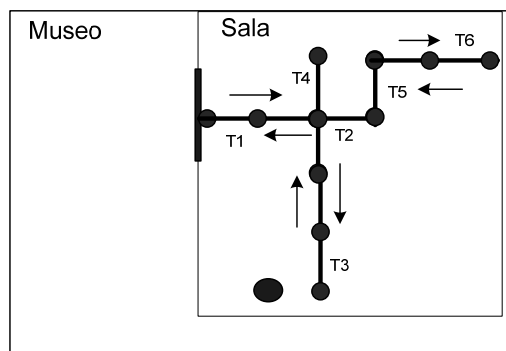
s setUsuario:u.

s calcularMejorCaminoPara:u Hasta:facultad.

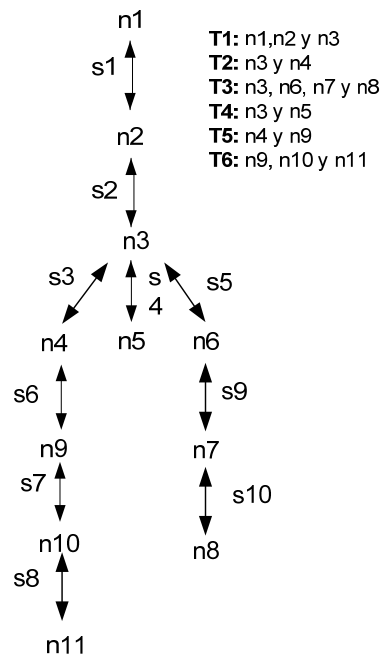
(u getCaminoSugerido) getTramos

2. Movilidad Red De Circulación Interna

Para mostrar un ejemplo de esta movilidad supondremos una Sala ubicada en un Museo, en la cual hay un Florero. El usuario esta ubicado al ingreso de la sala y quiere llegar hasta el Florero.



● Florero



*****movilidad Red Circulación Interna*****

s:= Sistema new.

sala := PIconMovilidadInterna new.

florero := PISinMovilidadInterna new.

*****Creación de Nodos*****

n1 := NodoCirculacionInterna new.

n1 initialize: 'nodo1'.

n2 := NodoCirculacionInterna new.
n2 initialize: 'nodo 2 '.

n3 := NodoCirculacionInterna new.
n3 initialize: 'nodo 3'.

n4 := NodoCirculacionInterna new.
n4 initialize: 'nodo4'.

n5 := NodoCirculacionInterna new.
n5 initialize: 'nodo 5 '.

n6 := NodoCirculacionInterna new.
n6 initialize: 'nodo 6'.
n7 := NodoCirculacionInterna new.
n7 initialize: 'nodo 7'.

n8 := NodoCirculacionInterna new.
n8 initialize: 'nodo 8'.

n9 := NodoCirculacionInterna new.
n9 initialize: 'nodo 9'.

n10 := NodoCirculacionInterna new.
n10 initialize: 'nodo 10'.

n11 := NodoCirculacionInterna new.
n11 initialize: 'nodo 11 '.

" Segmentos "

seg1:= SegmentoCirculacionInterna new.
seg1 initialize: 'seg1 ' Con: n1 Con:n2.
seg11:= SegmentoCirculacionInterna new.
seg11 initialize: 'seg11 ' Con: n2 Con:n1.

seg2:= SegmentoCirculacionInterna new.
seg2 initialize: 'seg2 ' Con: n2 Con:n3.
seg22:= SegmentoCirculacionInterna new.
seg22 initialize: 'seg22 ' Con: n3 Con:n2.

seg3:= SegmentoCirculacionInterna new.
seg3 initialize: 'seg 3 ' Con: n3 Con:n4.
seg33:= SegmentoCirculacionInterna new.
seg33 initialize: 'seg 33 ' Con: n4 Con:n3.

seg4:= SegmentoCirculacionInterna new.
seg4 initialize: 'seg4 ' Con: n3 Con:n5.

seg44:= SegmentoCirculacionInterna new.
seg44 initialize: 'seg44 ' Con: n5 Con:n3.

seg5:= SegmentoCirculacionInterna new.
seg5 initialize: 'seg5 ' Con: n3 Con:n6.
seg55:= SegmentoCirculacionInterna new.
seg55 initialize: 'seg55 ' Con: n6 Con:n3.

seg6:= SegmentoCirculacionInterna new.
seg6 initialize: 'seg 6 ' Con: n4 Con:n9.
seg66:= SegmentoCirculacionInterna new.
seg66 initialize: 'seg 66 ' Con: n9 Con:n4.

seg7:= SegmentoCirculacionInterna new.
seg7 initialize: 'seg7 ' Con: n9 Con:n10.
seg77:= SegmentoCirculacionInterna new.
seg77 initialize: 'seg77 ' Con: n10 Con:n9.

seg8:= SegmentoCirculacionInterna new.
seg8 initialize: 'seg8 ' Con: n10 Con:n11.
seg88:= SegmentoCirculacionInterna new.
seg88 initialize: 'seg88 ' Con: n11 Con:n10.

seg9:= SegmentoCirculacionInterna new.
seg9 initialize: 'seg9 ' Con: n6 Con:n7.
seg99:= SegmentoCirculacionInterna new.
seg99 initialize: 'seg99 ' Con: n7 Con:n6.

seg10:= SegmentoCirculacionInterna new.
seg10 initialize: 'seg10 ' Con: n7 Con:n8.
seg01:= SegmentoCirculacionInterna new.
seg01 initialize: 'seg01 ' Con: n8 Con:n7.

segmentos := OrderedCollection new.
segmentos add: seg1 .
n1 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg2.
segmentos add: seg11.
n2 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg22 .
segmentos add: seg3 .
segmentos add: seg4 .
segmentos add: seg5 .
n3 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg33.
segmentos add: seg6.
n4 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg44.
n5 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg9.
segmentos add: seg55.
n6 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg10.
segmentos add: seg99.
n7 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg01.
n8 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg7.
segmentos add: seg66.
n9 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg8.
segmentos add: seg77.
n10 setSegmentos: segmentos.

segmentos := OrderedCollection new.
segmentos add: seg88.
n11 setSegmentos: segmentos.

tramo1 := Tramo new.
tramo1 setNombre: 'Tramo 1'.

tramos1:= OrderedCollection new.
tramos1 add: seg1.
tramos1 add: seg2.
tramos1 add: seg11.
tramos1 add: seg22.
tramo1 setSegmentos:tramos1.

tramo2 := Tramo new.

tramo2 setNombre: 'Tramo 2'.

tramos2:= OrderedCollection new.
tramos2 add: seg3.
tramos2 add: seg33.
tramo2 setSegmentos:tramos2.

tramo3 := Tramo new.
tramo3 setNombre: 'Tramo 3'.

tramos3:= OrderedCollection new.
tramos3 add: seg3.
tramos3 add: seg22.

tramo3 setSegmentos:tramos3.

tramo4 := Tramo new.
tramo4 setNombre: 'Tramo 4'.

tramos4:= OrderedCollection new.
tramos4 add: seg4.
tramos4 add: seg44.
tramo4 setSegmentos:tramos4.

tramo5 := Tramo new.
tramo5 setNombre: 'Tramo5'.
tramos5:= OrderedCollection new.
tramos5 add: seg6.
tramos5 add: seg66.
tramo5 setSegmentos:tramos5.

tramo6 := Tramo new.
tramo6 setNombre: 'Tramo6'.
tramos6:= OrderedCollection new.
tramos6 add: seg7.
tramos6 add: seg8.
tramos6 add: seg77.
tramos6 add: seg88.
tramo6 setSegmentos:tramos6.

seg1 setTramo:tramo1.
seg11 setTramo:tramo1.
seg2 setTramo:tramo1.
seg22 setTramo:tramo1.
seg3 setTramo:tramo2.
seg33 setTramo:tramo2.
seg4 setTramo:tramo4.
seg44 setTramo:tramo4.
seg5 setTramo:tramo6.
seg55 setTramo:tramo6.

```
seg6 setTramo:tramo5.  
seg66 setTramo:tramo5.  
seg7 setTramo:tramo6.  
seg77 setTramo:tramo6.  
seg8 setTramo:tramo6.  
seg88 setTramo:tramo6.  
seg9 setTramo:tramo3.  
seg99 setTramo:tramo3.  
seg10 setTramo:tramo3.  
seg01 setTramo:tramo3.
```

```
tramos:= OrderedCollection new.  
tramos add:tramo1.  
tramos add:tramo2.  
tramos add:tramo3.  
tramos add:tramo4.  
tramos add:tramo5.  
tramos add:tramo6.
```

```
movRCI := RedCirculacionInterna new.  
movRCI setTramos:tramos.
```

```
(sala getMovilidades) add: movRCI.
```

```
"*****puntos de Acceso*****"
```

```
pa1:= PuntoDeAcceso new.  
pa1 setPim:florero.  
pa1 setUnidadDesplazamiento:seg10.  
pa1 setMovilidad: movRCI.  
pa1 setContenidoEn: sala.
```

```
(florero getPuntosAcceso) add: pa1.
```

```
"/*****CREACION USUARIO*****/"
```

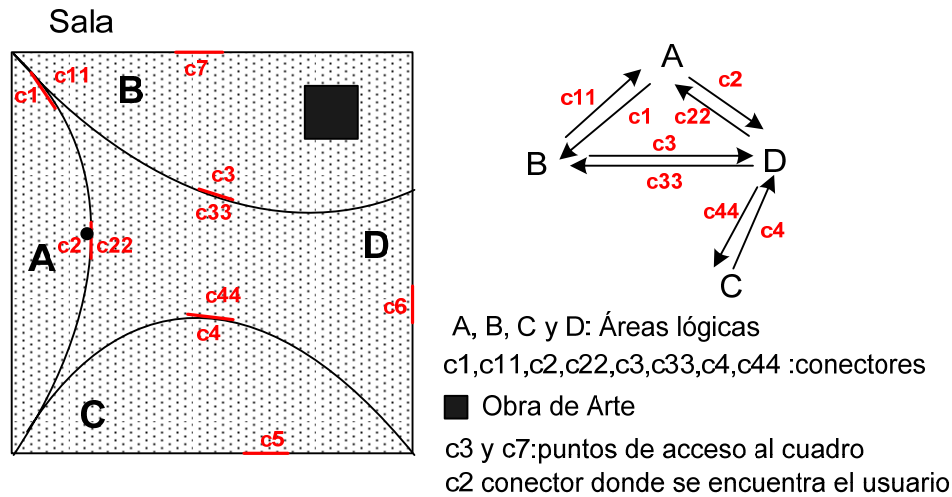
```
u:= Usuario new.  
ubicacion := UbicacionUsuario new.  
ubicacion setPIMovilidad: sala.  
ubicacion setUnidadDesplazamiento: seg1.  
u setUbicacionUsuario: ubicacion.
```

```
s calcularMejorCaminoPara:u Hasta:florero.
```

3. Movilidad Transitar Áreas Lógicas

Para esta movilidad seguiremos con el ejemplo de la Sala, pero ahora el destino deseado por el usuario es una Obra de Arte, que se encuentra dentro de la misma.

El usuario se encuentra ubicado en el Área A de la sala. Y la obra de arte en el área B de la misma.



```
s := Sistema new.
sala := PIconMovilidadInterna new.
sala setDescripcion:'sala'.
cuadro:= PISinMovilidadInterna new.
cuadro setDescripcion:'facultad'.
```

```
aA := AreaLogica new.
aA setNombre: 'areaA'.
aB := AreaLogica new.
aB setNombre: 'areaB'.
aC := AreaLogica new.
aC setNombre: 'areaC'.
aD := AreaLogica new.
aD setNombre: 'areaD'.
aNula:= AreaLogicaNula new.
```

```
c1:= ConectorAreaLogica new.
c1 initialize:'c1' Desde:aA Hasta:aB.
c11:= ConectorAreaLogica new.
c11 initialize:'c11' Desde:aB Hasta:aA.
c2:= ConectorAreaLogica new.
c2 initialize:'c2' Desde:aA Hasta:aD.
c22:= ConectorAreaLogica new.
c22 initialize:'c22' Desde:aD Hasta:aA.
```

```
c3:= ConectorAreaLogica new.  
c3 initialize:'c3' Desde:aB Hasta:aD.  
c33:= ConectorAreaLogica new.  
c33 initialize:'c33' Desde:aD Hasta:aB.  
c4:= ConectorAreaLogica new.  
c4 initialize:'c4' Desde:aC Hasta:aD.  
c44:= ConectorAreaLogica new.  
c44 initialize:'c4' Desde:aD Hasta:aC.  
c5:= ConectorAreaLogica new.  
c5 initialize:'c5' Desde:aC Hasta:aNula.  
c6:= ConectorAreaLogica new.  
c6 initialize:'c6' Desde:aD Hasta:aNula.  
c7:= ConectorAreaLogica new.  
c7 initialize:'c7' Desde:aB Hasta:aNula.
```

```
conectores:= OrderedCollection new.
```

```
" conectores add: c1. "  
conectores add: c2.  
conectores add: c3.  
conectores add: c4.  
conectores add: c5.  
conectores add: c6.  
conectores add: c7.  
" conectores add: c11."  
conectores add: c22.  
conectores add: c33.  
conectores add: c44.
```

```
movTAL := TransitarAreasLogicas new.  
movTAL setConectoresAreaLogica: conectores.
```

```
"puntos de acceso"
```

```
pa1:= PuntoDeAcceso new.  
pa1 setPim:cuadro.  
pa1 setUnidadDesplazamiento:c3.
```

```
pa2:= PuntoDeAcceso new.  
pa2 setPim:cuadro.  
pa2 setUnidadDesplazamiento:c7.
```

```
pa1 setMovilidad: movTAL.  
pa1 setContenidoEn: sala.  
pa2 setMovilidad: movTAL.  
pa2 setContenidoEn: sala.
```

```
(sala getMovilidades) add: movTAL.
```

(cuadro getPuntosAcceso) add: pa1.

(cuadro getPuntosAcceso) add: pa2.

u:= Usuario new.

ubicacion := UbicacionUsuario new.

ubicacion setPIMovilidad: sala.

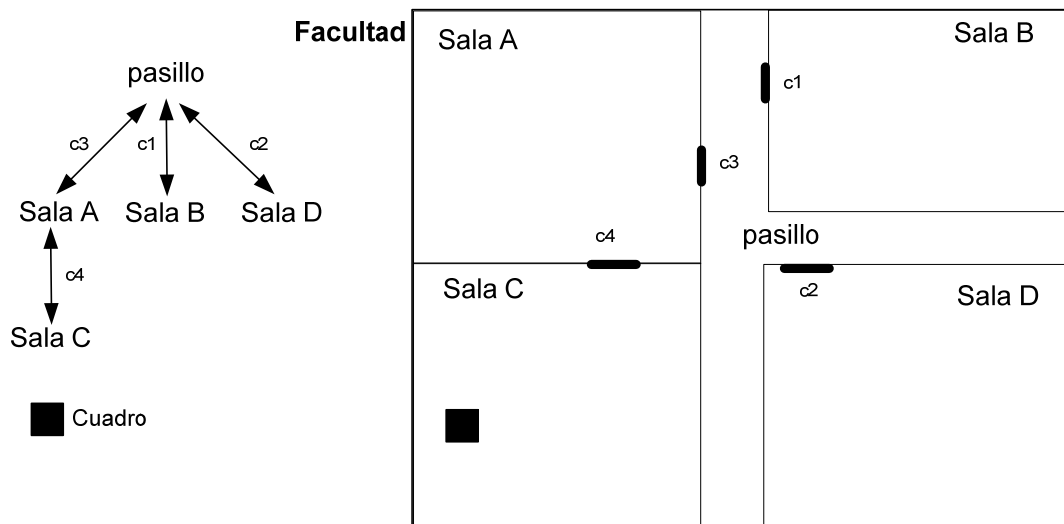
ubicacion setUnidadDesplazamiento: c2.

u setUbicacionUsuario: ubicacion.

s calcularMejorCaminoPara:u Hasta:cuadro.

4. Movilidad Transitar Área Interés

Para este ejemplo supondremos el modelado de una Facultad, en la cual hay una placa recordatoria que es de interés del Usuario, la placa se encuentra en la Sala C de la Facultad y el Usuario en la Sala D, de la misma.



s:= Sistema new.

sala := PIconMovilidadInterna new.

cuadro:= PISinMovilidadInterna new.

museo := PIconMovilidadInterna new.

sala setDescripcion:'sala'.

cuadro setDescripcion:'cuadro'.

museo setDescripcion: 'museo'.

"*****moviliadad Area Interes *****"

A := AreaInteresConcreta new.

A setNombre: 'Area I A'.
B := AreaInteresConcreta new.
B setNombre: 'Area I B'.
C := AreaInteresConcreta new.
C setNombre: 'Area I C'.
D := AreaInteresConcreta new.
D setNombre: 'Area I D'.
Pasillo := AreaInteresConcreta new.
Pasillo setNombre: 'Area I Pasillo'.

ANula := AreaInteresNula new.
ANula setNombre: 'Area Interes nula '.

c1:= ConectorAreaInteres new.
c1 initialize:'cB-Pasillo' Desde:B Hasta:Pasillo.

c11:= ConectorAreaInteres new.
c11 initialize:'cPasillo-B' Desde:Pasillo Hasta:B.

c2:= ConectorAreaInteres new.
c2 initialize:'cD-Pasillo' Desde:D Hasta:Pasillo.
c22:= ConectorAreaInteres new.
c22 initialize:'cPasillo-D' Desde:Pasillo Hasta:D.

c3:= ConectorAreaInteres new.
c3 initialize:'cPasillo-A' Desde:Pasillo Hasta:A.
c33:= ConectorAreaInteres new.
c33 initialize:'cA-Pasillo' Desde:A Hasta:Pasillo.

c4:= ConectorAreaInteres new.
c4 initialize:'cA-C' Desde:A Hasta:C.
c44:= ConectorAreaInteres new.
c44 initialize:'cC-A' Desde:C Hasta:A.

conectores:= OrderedCollection new.

conectores add: c1.
conectores add: c2.
conectores add: c3.
conectores add: c4.
conectores add: c11.
conectores add: c22.
conectores add: c33.
conectores add: c44.

movTAI := TransitarAreasInteres new.
movTAI setConectoresAreaInteres: conectores.

```
(sala getMovilidades) add: movTAI.
```

```
*****puntos de Acceso*****"
```

```
pa1:= PuntoDeAcceso new.
pa1 setPim:cuadro.
pa1 setUnidadDesplazamiento:c4.
pa1 setMovilidad: movTAI.
pa1 setContenidoEn:sala.
```

```
(cuadro getPuntosAcceso) add: pa1.
```

```
*****CREACION USUARIO*****"
```

```
u:= Usuario new.
ubicacion := UbicacionUsuario new.
ubicacion setPIMovilidad: sala.
ubicacion setUnidadDesplazamiento: c2.
u setUbicacionUsuario: ubicacion.
```

```
s calcularCaminoPara:u Hasta:cuadro.
```

B. Ejemplos que involucran mas de una movilidad

1. Ejemplo que involucra distintos escenarios

En ese último ejemplo mostraremos un caso en el cual el usuario se encuentra en una casa y quiere llegar hasta una computadora que se encuentra en una sala de una Facultad, teniendo que pasar por las calles de una ciudad.

```
s := Sistema new.
computadora := PISinMovilidadInterna new.
computadora setDescripcion:'computadora'.
```

```
***** CIUDAD *****"
```

```
ciudad := PIconMovilidadInterna new.
ciudad setDescripcion: 'ciudad'.
```

```
***Todos los nodos de la Ciudad***"
```

```
n1:= NodoRedCalles new.
n1 initialize:'n1'.
```

```
n2:= NodoRedCalles new.
n2 initialize:'n2'.
```

```
n3:= NodoRedCalles new.
n3 initialize:'n3'.
```

n4:= NodoRedCalles new.
n4 initialize:'n4'.

n5:= NodoRedCalles new.
n5 initialize:'n5'.

n6:= NodoRedCalles new.
n6 initialize:'n6'.

n7:= NodoRedCalles new.
n7 initialize:'n7'.

n8:= NodoRedCalles new.
n8 initialize:'n8'.

n9:= NodoRedCalles new.
n9 initialize:'n9'.

n10:= NodoRedCalles new.
n10 initialize:'n10'.

n11:= NodoRedCalles new.
n11 initialize:'n11'.

n12:= NodoRedCalles new.
n12 initialize:'n12'.

n13:= NodoRedCalles new.
n13 initialize:'n13'.

n14:= NodoRedCalles new.
n14 initialize:'n14'.

n15:= NodoRedCalles new.
n15 initialize:'n15'.

n16:= NodoRedCalles new.
n16 initialize:'n16'.

n17:= NodoRedCalles new.
n17 initialize:'n17'.

n18:= NodoRedCalles new.
n18 initialize:'n18'.

n19:= NodoRedCalles new.
n19 initialize:'n19'.

n20:= NodoRedCalles new.

n20 initialize:'n20'.

n21:= NodoRedCalles new.
n21 initialize:'n21'.

n22:= NodoRedCalles new.
n22 initialize:'n22'.

n23:= NodoRedCalles new.
n23 initialize:'n23'.

n24:= NodoRedCalles new.
n24 initialize:'n24'.

n25:= NodoRedCalles new.
n25 initialize:'n25'.

n26:= NodoRedCalles new.
n26 initialize:'n26'.

n27:= NodoRedCalles new.
n27 initialize:'n27'.

n28:= NodoRedCalles new.
n28 initialize:'n28'.

n29:= NodoRedCalles new.
n29 initialize:'n29'.

n30:= NodoRedCalles new.
n30 initialize:'n30'.

n31:= NodoRedCalles new.
n31 initialize:'n31'.

n32:= NodoRedCalles new.
n32 initialize:'n32'.

n33:= NodoRedCalles new.
n33 initialize:'n33'.

n34:= NodoRedCalles new.
n34 initialize:'n34'.

n35:= NodoRedCalles new.
n35 initialize:'n35'.

n36:= NodoRedCalles new.
n36 initialize:'n36'.

n37:= NodoRedCalles new.
n37 initialize:'n37'.

n38:= NodoRedCalles new.
n38 initialize:'n38'.

n39:= NodoRedCalles new.
n39 initialize:'n39'.

n40:= NodoRedCalles new.
n40 initialize:'n40'.

"Armamos las calles. Cada calle tiene 7 segmentos"

"Armamos calle48 "

c48 := Calle new.
c48 initialize: 'calle 48'.

s481 := SegmentoRedCalles new.
s481 initialize:'seg48-de-1-2' calle:c48 nodoInicial: n2 nodoFinal:n1.

s482 := SegmentoRedCalles new.
s482 initialize:'seg48-de-115-1' calle:c48 nodoInicial: n3 nodoFinal:n2.

s483 := SegmentoRedCalles new.
s483 initialize:'seg48-de116-115' calle:c48 nodoInicial: n4 nodoFinal:n3.

s484 := SegmentoRedCalles new.
s484 initialize:'seg48-de-117-116' calle:c48 nodoInicial: n5 nodoFinal:n4.

s485 := SegmentoRedCalles new.
s485 initialize:'seg48-de-118-117' calle:c48 nodoInicial: n6 nodoFinal:n5.

s486 := SegmentoRedCalles new.
s486 initialize:'seg48-de-119-118' calle:c48 nodoInicial: n7 nodoFinal:n6.

s487 := SegmentoRedCalles new.
s487 initialize:'seg48-de120-119' calle:c48 nodoInicial: n8 nodoFinal: n7.

" ****agregamos los segmentos para cada nodo*****"

(n2 getSegmentos) add:s481.
(n3 getSegmentos) add:s482.
(n4 getSegmentos) add:s483.
(n5 getSegmentos) add:s484.
(n6 getSegmentos) add:s485.
(n7 getSegmentos) add:s486.
(n8 getSegmentos) add:s487.

```
segmentos := OrderedCollection new.  
segmentos add:s481.  
segmentos add:s482.  
segmentos add:s483.  
segmentos add:s484.  
segmentos add:s485.  
segmentos add:s486.  
segmentos add:s487.
```

```
c48 setSegmentos:segmentos.
```

```
"Armamos calle49"
```

```
c49 := Calle new.  
c49 initialize: 'calle 49'.
```

```
s491 := SegmentoRedCalles new.  
s491 initialize:'seg49de-2-1' calle:c49 nodoInicial: n9 nodoFinal:n10.
```

```
s492 := SegmentoRedCalles new.  
s492 initialize:'seg49-de-1-115' calle:c49 nodoInicial: n10 nodoFinal:n11.
```

```
s493 := SegmentoRedCalles new.  
s493 initialize:'seg49-de-115-116' calle:c49 nodoInicial: n11 nodoFinal:n12.
```

```
s494 := SegmentoRedCalles new.  
s494 initialize:'seg49-de-116-117' calle:c49 nodoInicial: n12 nodoFinal:n13.
```

```
s495 := SegmentoRedCalles new.  
s495 initialize:'seg49-de-117-118' calle:c49 nodoInicial: n13 nodoFinal:n14.
```

```
s496 := SegmentoRedCalles new.  
s496 initialize:'seg49-de-118-119' calle:c49 nodoInicial: n14 nodoFinal:n15.
```

```
s497 := SegmentoRedCalles new.  
s497 initialize:'seg49-de-119-120' calle:c49 nodoInicial: n15 nodoFinal:n16.
```

```
"agregamos los segmentos para cada nodo"
```

```
(n9 getSegmentos) add:s491.  
(n10 getSegmentos) add:s492.  
(n11 getSegmentos) add:s493.  
(n12 getSegmentos) add:s494.  
(n13 getSegmentos) add:s495.  
(n14 getSegmentos) add:s496.  
(n15 getSegmentos) add:s497.
```

```
segmentos := OrderedCollection new.  
segmentos add: s491.  
segmentos add: s492.
```

segmentos add: s493.
segmentos add:s494.
segmentos add:s495.
segmentos add:s496.
segmentos add:s497.

c49 setSegmentos:segmentos.

"Armamos calle50"

c50 := Calle new.
c50 initialize: 'calle 50'.

s501 := SegmentoRedCalles new.
s501 initialize:'seg50-de-1-2' calle:c50 nodoInicial: n18 nodoFinal:n17.

s502 := SegmentoRedCalles new.
s502 initialize:'seg50-de-1-115' calle:c50 nodoInicial: n18 nodoFinal:n19.

s503 := SegmentoRedCalles new.
s503 initialize:'seg50-de-115-116' calle:c50 nodoInicial: n19 nodoFinal:n20.

s504 := SegmentoRedCalles new.
s504 initialize:'seg50-de-116-117' calle:c50 nodoInicial: n20 nodoFinal:n21.

s505 := SegmentoRedCalles new.
s505 initialize:'seg50-de-117-118' calle:c50 nodoInicial: n21 nodoFinal:n22.

s506 := SegmentoRedCalles new.
s506 initialize:'seg50-de-118-119' calle:c50 nodoInicial: n22 nodoFinal:n23.

s507 := SegmentoRedCalles new.
s507 initialize:'seg50-de-119-120' calle:c50 nodoInicial: n23 nodoFinal:n24.

s508 := SegmentoRedCalles new.
s508 initialize:'seg50-de-116-115' calle:c50 nodoInicial: n20 nodoFinal:n19.

s509 := SegmentoRedCalles new.
s509 initialize:'seg50-de-117-116' calle:c50 nodoInicial: n21 nodoFinal:n20.

s5010 := SegmentoRedCalles new.
s5010 initialize:'seg50-de-118-117' calle:c50 nodoInicial: n22 nodoFinal:n21.

s5011 := SegmentoRedCalles new.
s5011 initialize:'seg50-de-119-118' calle:c50 nodoInicial: n23 nodoFinal:n22.

s5012 := SegmentoRedCalles new.
s5012 initialize:'seg50-de-120-119' calle:c50 nodoInicial: n24 nodoFinal:n23.

"agregamos los segmentos para cada nodo"

(n18 getSegmentos) add:s501.
(n18 getSegmentos) add:s502.
(n19 getSegmentos) add:s503.
(n20 getSegmentos) add:s504.
(n20 getSegmentos) add:s508.
(n21 getSegmentos) add:s505.
(n21 getSegmentos) add:s509.
(n22 getSegmentos) add:s506.
(n22 getSegmentos) add:s5010.
(n23 getSegmentos) add:s507.
(n23 getSegmentos) add:s5011.
(n24 getSegmentos) add:s5012.

segmentos := OrderedCollection new.
segmentos add: s501.
segmentos add: s502.
segmentos add: s503.
segmentos add:s504.
segmentos add:s505.
segmentos add:s506.
segmentos add:s507.
segmentos add:s508.
segmentos add:s509.
segmentos add:s5010.
segmentos add:s5011.
segmentos add:s5012.

c50 setSegmentos:segmentos.

"Armamos calle51"

c51 := Calle new.
c51 initialize: 'calle 51'.

s511 := SegmentoRedCalles new.
s511 initialize:'seg51-de-2-1' calle:c51 nodoInicial: n25 nodoFinal:n26.

s512 := SegmentoRedCalles new.
s512 initialize:'seg51-de-1-115' calle:c51 nodoInicial: n26 nodoFinal:n27.

s513 := SegmentoRedCalles new.
s513 initialize:'seg51-de-115-116' calle:c51 nodoInicial: n27 nodoFinal:n28.

s514 := SegmentoRedCalles new.
s514 initialize:'seg51-de-116-117' calle:c51 nodoInicial: n28 nodoFinal:n29.

s515 := SegmentoRedCalles new.
s515 initialize:'seg51-de-117-118' calle:c51 nodoInicial: n29 nodoFinal:n30.

s516 := SegmentoRedCalles new.
s516 initialize:'seg51-de-118-119' calle:c51 nodoInicial: n30 nodoFinal:n31.

s517 := SegmentoRedCalles new.
s517 initialize:'seg51-de-119-120' calle:c51 nodoInicial: n31 nodoFinal:n32.

"agregamos los segmentos para cada nodo"

(n25 getSegmentos) add:s511.
(n26 getSegmentos) add:s512.
(n27 getSegmentos) add:s513.
(n28 getSegmentos) add:s514.
(n29 getSegmentos) add:s515.
(n30 getSegmentos) add:s516.

segmentos := OrderedCollection new.
segmentos add: s511.
segmentos add: s512.
segmentos add: s513.
segmentos add:s514.
segmentos add:s515.
segmentos add:s516.
segmentos add:s517.

c51 setSegmentos:segmentos.

"*****Armamos calle52*****"

c52 := Calle new.
c52 initialize: 'calle 52'.

s521 := SegmentoRedCalles new.
s521 initialize:'seg52-de-1-2' calle:c52 nodoInicial: n34 nodoFinal:n33.

s522 := SegmentoRedCalles new.
s522 initialize:'seg52-de-115-1' calle:c52 nodoInicial: n35 nodoFinal:n34.

s523 := SegmentoRedCalles new.
s523 initialize:'seg52-de-116-115' calle:c52 nodoInicial: n36 nodoFinal:n35.

s524 := SegmentoRedCalles new.
s524 initialize:'seg52-de-117-116' calle:c52 nodoInicial: n37 nodoFinal:n36.

s525 := SegmentoRedCalles new.
s525 initialize:'seg52-de-118-117' calle:c52 nodoInicial: n38 nodoFinal:n37.

s526 := SegmentoRedCalles new.
s526 initialize:'seg52-de-119-118' calle:c52 nodoInicial: n39 nodoFinal:n38.

s527 := SegmentoRedCalles new.

s527 initialize:'seg52-de-120-119' calle:c52 nodoInicial: n40 nodoFinal:n39.

"agregamos los segmentos para cada nodo"

(n34 getSegmentos) add:s521.

(n35 getSegmentos) add:s522.

(n36 getSegmentos) add:s523.

(n37 getSegmentos) add:s524.

(n38 getSegmentos) add:s525.

(n39 getSegmentos) add:s526.

(n40 getSegmentos) add:s527.

segmentos := OrderedCollection new.

segmentos add: s521.

segmentos add: s522.

segmentos add: s523.

segmentos add:s524.

segmentos add:s525.

segmentos add:s526.

segmentos add:s527.

c52 setSegmentos:segmentos.

"*****Armamos calle 2*****"

c2 := Calle new.

c2 initialize: 'calle 2'.

s21 := SegmentoRedCalles new.

s21 initialize:'seg2-de48-49' calle:c2 nodoInicial: n1 nodoFinal:n9.

s22 := SegmentoRedCalles new.

s22 initialize:'seg2-de49-50' calle:c2 nodoInicial: n9 nodoFinal:n17.

s23 := SegmentoRedCalles new.

s23 initialize:'seg2-de-50-51' calle:c2 nodoInicial: n17 nodoFinal:n25.

s24 := SegmentoRedCalles new.

s24 initialize:'seg2-de51-52' calle:c2 nodoInicial: n25 nodoFinal:n33.

"agregamos los segmentos para cada nodo"

(n1 getSegmentos) add:s21.

(n9 getSegmentos) add:s22.

(n17 getSegmentos) add:s23.

(n25 getSegmentos) add:s24.

segmentos := OrderedCollection new.

segmentos add: s21.

segmentos add: s22.

segmentos add: s23.
segmentos add:s24.

c2 setSegmentos:segmentos.

"*****Armanos calle 1*****"

c1 := Calle new.
c1 initialize: 'calle 1'.

s11 := SegmentoRedCalles new.
s11 initialize:'seg1-de-48-49' calle:c1 nodoInicial: n2 nodoFinal:n10.

s12 := SegmentoRedCalles new.
s12 initialize:'seg1-de-49-50' calle:c1 nodoInicial: n10 nodoFinal:n18.

s13 := SegmentoRedCalles new.
s13 initialize:'seg1-de50-51' calle:c1 nodoInicial: n18 nodoFinal:n26.

s14 := SegmentoRedCalles new.
s14 initialize:'seg1-de-51-52' calle:c1 nodoInicial: n26 nodoFinal:n34.

s15 := SegmentoRedCalles new.
s15 initialize:'seg1-de52-51' calle:c1 nodoInicial: n34 nodoFinal:n26.

s16 := SegmentoRedCalles new.
s16 initialize:'seg1-de51-50' calle:c1 nodoInicial: n26 nodoFinal:n18.

s17 := SegmentoRedCalles new.
s17 initialize:'seg1-de50-49' calle:c1 nodoInicial: n18 nodoFinal:n10.

s18 := SegmentoRedCalles new.
s18 initialize:'seg1-de49-48' calle:c1 nodoInicial: n10 nodoFinal:n2.

"agregamos los segmentos para cada nodo"

(n2 getSegmentos) add:s11.
(n10 getSegmentos) add:s12.
(n10 getSegmentos) add:s18.
(n18 getSegmentos) add:s13.
(n18 getSegmentos) add:s17.
(n26 getSegmentos) add:s14.
(n26 getSegmentos) add:s16.
(n34 getSegmentos) add:s15.

segmentos1 := OrderedCollection new.
segmentos1 add: s11.
segmentos1 add: s12.
segmentos1 add: s13.
segmentos1 add: s14.
segmentos1 add: s15.

segmentos1 add: s16.
segmentos1 add: s17.
segmentos1 add: s18.

c1 setSegmentos:segmentos1.

"*****Armamos calle 115*****"

c115 := Calle new.
c115 initialize: 'calle 115'.

s1151 := SegmentoRedCalles new.
s1151 initialize:'seg115-de-49-48' calle:c115 nodoInicial: n11 nodoFinal:n3.

s1152 := SegmentoRedCalles new.
s1152 initialize:'seg115-de50-49' calle:c115 nodoInicial: n19 nodoFinal:n11.

s1153 := SegmentoRedCalles new.
s1153 initialize:'seg115-de-51-50' calle:c115 nodoInicial: n27 nodoFinal:n19.

s1154 := SegmentoRedCalles new.
s1154 initialize:'seg115-de-52-51' calle:c115 nodoInicial: n35 nodoFinal:n27.

"agregamos los segmentos para cada nodo"

(n11 getSegmentos) add:s1151.
(n19 getSegmentos) add:s1152.
(n27 getSegmentos) add:s1153.
(n35 getSegmentos) add:s1154.

segmentos115 := OrderedCollection new.
segmentos115 add: s1151.
segmentos115 add: s1152.
segmentos115 add: s1153.
segmentos115 add: s1154.

c115 setSegmentos:segmentos115.

"*****Armamos calle 116*****"

c116 := Calle new.
c116 initialize: 'calle 116'.

s1161 := SegmentoRedCalles new.
s1161 initialize:'seg116-de-48-49' calle:c116 nodoInicial: n4 nodoFinal:n12.

s1162 := SegmentoRedCalles new.
s1162 initialize:'seg116-de-49-50' calle:c116 nodoInicial: n12 nodoFinal:n20.

s1163 := SegmentoRedCalles new.
s1163 initialize:'seg116-de-50-51' calle:c116 nodoInicial: n20 nodoFinal:n28.

s1164 := SegmentoRedCalles new.
s1164 initialize:'seg116-de-51-52' calle:c116 nodoInicial: n28 nodoFinal:n36.

"agregamos los segmentos para cada nodo"
(n4 getSegmentos) add:s1161.
(n12 getSegmentos) add:s1162.
(n20 getSegmentos) add:s1163.
(n28 getSegmentos) add:s1164.

segmentos116 := OrderedCollection new.
segmentos116 add: s1161.
segmentos116 add: s1162.
segmentos116 add: s1163.
segmentos116 add: s1164.

c116 setSegmentos:segmentos116.

"*****Armamos calle 117*****"
c117 := Calle new.
c117 initialize: 'calle 117'.

s1171 := SegmentoRedCalles new.
s1171 initialize:'seg117-de-49-48' calle:c117 nodoInicial: n13 nodoFinal:n5.

s1172 := SegmentoRedCalles new.
s1172 initialize:'seg117-de-50-49' calle:c117 nodoInicial: n21 nodoFinal:n13.

s1173 := SegmentoRedCalles new.
s1173 initialize:'seg117-de-51-50' calle:c117 nodoInicial: n29 nodoFinal:n21.

s1174 := SegmentoRedCalles new.
s1174 initialize:'seg117-52-51' calle:c117 nodoInicial: n37 nodoFinal:n29.

"agregamos los segmentos para cada nodo"
(n37 getSegmentos) add:s1174.
(n29 getSegmentos) add:s1173.
(n21 getSegmentos) add:s1172.
(n13 getSegmentos) add:s1171.

segmentos117 := OrderedCollection new.
segmentos117 add: s1171.
segmentos117 add: s1172.
segmentos117 add: s1173.
segmentos117 add: s1174.

c117 setSegmentos:segmentos117.

"*****Armamos calle 118*****"

```
c118 := Calle new.
c118 initialize: 'calle 118'.

s1181 := SegmentoRedCalles new.
s1181 initialize:'seg118-de-48-49' calle:c118 nodoInicial: n6 nodoFinal:n14.

s1182 := SegmentoRedCalles new.
s1182 initialize:'seg118-de-49-50' calle:c118 nodoInicial: n14 nodoFinal:n22.

s1183 := SegmentoRedCalles new.
s1183 initialize:'seg118-de-50-51' calle:c118 nodoInicial: n22 nodoFinal:n30.

s1184 := SegmentoRedCalles new.
s1184 initialize:'seg118-de-51-52' calle:c118 nodoInicial: n30 nodoFinal:n38.

"agregamos los segmentos para cada nodo"
(n6 getSegmentos) add:s1181.
(n14 getSegmentos) add:s1182.
(n22 getSegmentos) add:s1183.
(n30 getSegmentos) add:s1184.

segmentos118 := OrderedCollection new.
segmentos118 add: s1181.
segmentos118 add: s1182.
segmentos118 add: s1183.
segmentos118 add: s1184.

c118 setSegmentos:segmentos118.

"*****Armanos calle 119*****"
c119 := Calle new.
c119 initialize: 'calle 119'.

s1191 := SegmentoRedCalles new.
s1191 initialize:'seg119-de-49-48' calle:c119 nodoInicial: n15 nodoFinal:n7.

s1192 := SegmentoRedCalles new.
s1192 initialize:'seg119-de-50-49' calle:c119 nodoInicial: n23 nodoFinal:n15.

s1193 := SegmentoRedCalles new.
s1193 initialize:'seg119-de-51-50' calle:c119 nodoInicial: n31 nodoFinal:n23.

s1194 := SegmentoRedCalles new.
s1194 initialize:'seg119-de52-51' calle:c119 nodoInicial: n39 nodoFinal:n31.

"agregamos los segmentos para cada nodo"

(n39 getSegmentos) add:s1194.
(n31 getSegmentos) add:s1193.
```

(n23 getSegmentos) add:s1192.
(n15 getSegmentos) add:s1191.

segmentos119 := OrderedCollection new.
segmentos119 add: s1191.
segmentos119 add: s1192.
segmentos119 add: s1193.
segmentos119 add: s1194.

c119 setSegmentos:segmentos119.

"*****Armamos calle 120 *****"

c120 := Calle new.
c120 initialize: 'calle 120'.

s1201 := SegmentoRedCalles new.
s1201 initialize:'seg120-de48-49' calle:c120 nodoInicial: n8 nodoFinal:n16.

s1202 := SegmentoRedCalles new.
s1202 initialize:'seg120-de-49-50' calle:c120 nodoInicial: n16 nodoFinal:n24.

s1203 := SegmentoRedCalles new.
s1203 initialize:'seg120-de50-51' calle:c120 nodoInicial: n24 nodoFinal:n32.

s1204 := SegmentoRedCalles new.
s1204 initialize:'seg120*de51-52' calle:c120 nodoInicial: n32 nodoFinal:n40.

"agregamos los segmentos para cada nodo"

(n8 getSegmentos) add:s1201.
(n16 getSegmentos) add:s1202.
(n24 getSegmentos) add:s1203.
(n32 getSegmentos) add:s1204.

segmentos120 := OrderedCollection new.
segmentos120 add: s1201.
segmentos120 add: s1202.
segmentos120 add: s1203.
segmentos120 add: s1204.
c120 setSegmentos:segmentos120.

calles := OrderedCollection new.

calles add: c120.
calles add: c118.
calles add: c119.
calles add: c50.
calles add: c51.
calles add: c52.
calles add: c48.
calles add: c116.

calles add: c117.
calles add: c49.
calles add: c115.
calles add: c1.
calles add: c2.

movRC := RedDeCalles new.
movRC setCalles:calles.
(ciudad getMovilidades) add: movRC.

"*****FACULTAD *****"

facultad := PIconMovilidadInterna new.
facultad setDescripcion: 'facultad'.

A := AreaInteresConcreta new.
A setNombre: 'Area I A'.
A setVisitado: false.

B := AreaInteresConcreta new.
B setNombre: 'Area I B'.
B setVisitado: false.

C := AreaInteresConcreta new.
C setNombre: 'Area I C'.
C setVisitado: false.

D := AreaInteresConcreta new.
D setNombre: 'Area I D'.
D setVisitado: false.

Pasillo := AreaInteresConcreta new.
Pasillo setNombre: 'Area I Pasillo'.
Pasillo setVisitado: false.

ANula := AreaInteresNula new.
ANula setNombre: 'Area Interes nula'.
ANula setVisitado: false.

c1:= ConectorAreaInteres new.
c1 initialize:'cB-Pasillo' Desde:B Hasta:Pasillo.

c11:= ConectorAreaInteres new.
c11 initialize:'cPasillo-B' Desde:Pasillo Hasta:B.

c2:= ConectorAreaInteres new.
c2 initialize:'cD-Pasillo' Desde:D Hasta:Pasillo.
c22:= ConectorAreaInteres new.
c22 initialize:'cPasillo-D' Desde:Pasillo Hasta:D.

```
c3:= ConectorAreaInteres new.
c3 initialize:'cPasillo-A' Desde:Pasillo Hasta:A.
c33:= ConectorAreaInteres new.
c33 initialize:'cA-Pasillo' Desde:A Hasta:Pasillo.

c4:= ConectorAreaInteres new.
c4 initialize:'cA-C' Desde:A Hasta:C.
c44:= ConectorAreaInteres new.
c44 initialize:'cC-A' Desde:C Hasta:A.

c5:= ConectorAreaInteres new.
c5 initialize:'cPasillo-Nula' Desde:Pasillo Hasta:ANula.
c55:= ConectorAreaInteres new.
c55 initialize:'cNula-Pasillo' Desde:ANula Hasta:Pasillo.
```

```
conectores:= OrderedCollection new.
```

```
conectores add: c1.
conectores add: c2.
conectores add: c3.
conectores add: c4.
conectores add: c11.
conectores add: c22.
conectores add: c33.
conectores add: c44.
```

```
movTAI := TransitarAreasInteres new.
movTAI setConectoresAreaInteres: conectores.
```

```
(facultad getMovilidades) add: movTAI.
```

```
"*****CASA*****"
```

```
casa := PIconMovilidadInterna new.
casa setDescription: 'casa'.
```

```
Cocina := AreaLogica new.
Cocina setName: 'Cocina'.
Cocina setVisitado: false.
```

```
Baño := AreaLogica new.
Baño setName: 'Baño'.
Baño setVisitado: false.
```

```
Comedor := AreaLogica new.
Comedor setName: 'Comedor'.
Comedor setVisitado: false.
```

```
Dormitorio := AreaLogica new.
Dormitorio setName: 'Dormitorio'.
```

Dormitorio setVisitado: false.

CasaPasillo := AreaLogica new.
CasaPasillo setNombre: 'Casa-Pasillo'.
CasaPasillo setVisitado: false.

AreaNula := AreaLogica new.
AreaNula setNombre: 'Area logica nula'.
AreaNula setVisitado: false.

puerta1:= ConectorAreaLogica new.
puerta1 initialize:'Comedor-Calle' Desde:Comedor Hasta:AreaNula.
puerta11:= ConectorAreaLogica new.
puerta11 initialize:'Calle-Comedor' Desde:AreaNula Hasta:Comedor.

puerta2:= ConectorAreaLogica new.
puerta2 initialize:'Comedor-Cocina' Desde:Comedor Hasta:Cocina.
puerta22:= ConectorAreaLogica new.
puerta22 initialize:'Cocina-Comedor' Desde:Cocina Hasta:Comedor.

puerta3:= ConectorAreaLogica new.
puerta3 initialize:'Comedor-Pasillo' Desde:Comedor Hasta:CasaPasillo.
puerta33:= ConectorAreaLogica new.
puerta33 initialize:'Pasillo-Comedor' Desde:CasaPasillo Hasta:Comedor.

puerta4:= ConectorAreaLogica new.
puerta4 initialize:'Pasillo-Dormitorio' Desde:CasaPasillo Hasta:Dormitorio.
puerta44:= ConectorAreaLogica new.
puerta44 initialize:'Dormitorio-Pasillo' Desde:Dormitorio Hasta:CasaPasillo.

puerta5:= ConectorAreaLogica new.
puerta5 initialize:'Pasillo-Baño' Desde:CasaPasillo Hasta:Baño.
puerta55:= ConectorAreaLogica new.
puerta55 initialize:'Baño-Pasillo' Desde:Baño Hasta:CasaPasillo.

conectores:= OrderedCollection new.

conectores add: puerta1.
conectores add: puerta11.
conectores add: puerta2.
conectores add: puerta22.
conectores add: puerta3.
conectores add: puerta33.
conectores add: puerta4.
conectores add: puerta44.
conectores add: puerta5.
conectores add: puerta55.

movTAL := TransitarAreasLogicas new.
movTAL setConectoresAreaLogica: conectores.

(casa getMovilidades) add: movTAL.

*****PUNTOS DE ACCESOS*****"

"punto de acceso que entra a la facultad"

paFacCalle:= PuntoDeAcceso new.
paFacCalle setPim:facultad.
paFacCalle setUnidadDesplazamiento:c55.
paFacCalle setMovilidad: movTAL.
paFacCalle setContenidoEn: ciudad.

"punto de acceso que sale de la facultad"

paCalleFac:= PuntoDeAcceso new.
paCalleFac setPim:facultad.
paCalleFac setUnidadDesplazamiento:c5 .
paCalleFac setMovilidad: movTAL.
paCalleFac setContenidoEn: ciudad.

(facultad getPuntosAcceso) add: paCalleFac.
(facultad getPuntosAcceso) add: paFacCalle.

"punto de acceso que entra a la casa"

paCalleCasa:= PuntoDeAcceso new.
paCalleCasa setPim:casa.
paCalleCasa setUnidadDesplazamiento: puerta11.
paCalleCasa setMovilidad: movTAL.
paCalleCasa setContenidoEn: ciudad.

"punto de acceso que sale de la casa"

paCasaCalle:= PuntoDeAcceso new.
paCasaCalle setPim:casa.
paCasaCalle setUnidadDesplazamiento: puerta1.
paCasaCalle setMovilidad: movTAL.
paCasaCalle setContenidoEn: ciudad.

(casa getPuntosAcceso) add: paCalleCasa.
(casa getPuntosAcceso) add: paCasaCalle.

"PUNTO DE ACCESO DE LA CIUDAD"

paCiudad1:= PuntoDeAcceso new.
paCiudad1 setPim:ciudad.
paCiudad1 setUnidadDesplazamiento:s21.
paCiudad1 setMovilidad: movRC.
paCiudad1 setContenidoEn: ciudad.

```

paCiudad2:= PuntoDeAcceso new.
paCiudad2 setPim:ciudad.
paCiudad2 setUnidadDesplazamiento:s507.
paCiudad2 setMovilidad: movRC.
paCiudad2 setContenidoEn: ciudad.

```

```

paCiudad3:= PuntoDeAcceso new.
paCiudad3 setPim:ciudad.
paCiudad3 setUnidadDesplazamiento:s5012.
paCiudad3 setMovilidad: movRC.
paCiudad3 setContenidoEn: ciudad.

```

```

(ciudad getPuntosAcceso) add: paCiudad1.
(ciudad getPuntosAcceso) add: paCiudad2.
(ciudad getPuntosAcceso) add: paCiudad3.

```

```

"*****puntos e Acceso a la Computadora*****"

```

```

paCompu:= PuntoDeAcceso new.
paCompu setPim:computadora.
paCompu setUnidadDesplazamiento:c4.
paCompu setMovilidad: movTAI.
paCompu setContenidoEn:facultad.

```

```

(computadora getPuntosAcceso) add: paCompu.

```

```

"*****PUNTOS CONEXION *****"

```

```

puntoC1 := PuntoConexion new.
puntoC1 setOrigen: paCasaCalle.
puntoC1 setDestino: paCiudad1.

```

```

puntoC2 := PuntoConexion new.
puntoC2 setOrigen: paCalleFac.
puntoC2 setDestino: paCiudad2.

```

```

puntoC3 := PuntoConexion new.
puntoC3 setOrigen: paCalleFac.
puntoC3 setDestino: paCiudad3.

```

```

(s getPuntosConexion) add: puntoC1.
(s getPuntosConexion) add: puntoC2.
(s getPuntosConexion) add: puntoC3.

```

```

"/*****CREACION USUARIO*****/"

```

```

u:= Usuario new.
ubicacion := UbicacionUsuario new.

```

ubicacion setPIMovilidad: casa.
 ubicacion setUnidadDesplazamiento: puerta1.
 u setUbicacionUsuario: ubicacion.

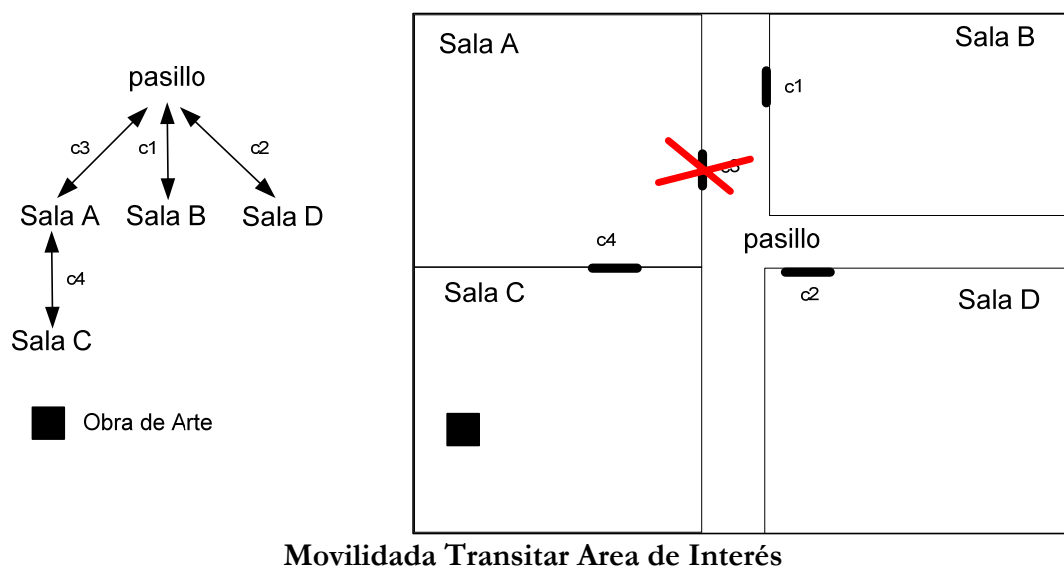
s setUsuario:u.

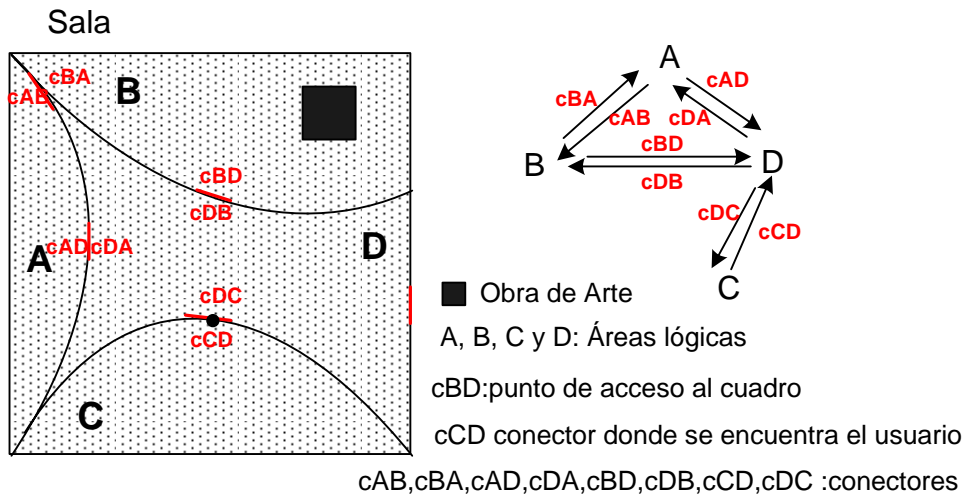
s calcularMejorCaminoPara:u Hasta:computadora.
 (u getCaminoSugerido) getTramos.

2. Ejemplo que involucra cambio de movilidad.

En este caso realizamos un ejemplo de una Sala con dos tipos de movilidades, Transitar Area de Interes y Transitar Area Lógica.

En la sala hay una obra de Arte, la cual es el destino deseado por el usuario. Este se encuentra en la sala y su ubicación tiene asociada la movilidad Transitar Areas de Interes. Debido a que no existe un camino, en este tipo de movilidad, se deberá repositicionar al usuario en la movilidad Transitar Area Lógica, para poder obtener un camino hasta la obra de Arte..





Movilidad Transitar Area Lógica

s:= Sistema new.

sala := PIconMovilidadInterna new.

cuadro:= PISinMovilidadInterna new.

sala setDescription:'sala'.

cuadro setDescription:'cuadro'.

"*****moviliadad Area Interes *****"

A := AreaInteresConcreta new.

A setNombre: 'Area I A'.

B := AreaInteresConcreta new.

B setNombre: 'Area I B'.

C := AreaInteresConcreta new.

C setNombre: 'Area I C'.

D := AreaInteresConcreta new.

D setNombre: 'Area I D'.

Pasillo := AreaInteresConcreta new.

Pasillo setNombre: 'Area I Pasillo'.

ANula := AreaInteresNula new.

ANula setNombre: 'Area Interes nula '.

c1:= ConectorAreaInteres new.

c1 initialize:'cB-Pasillo' Desde:B Hasta:Pasillo.

c11:= ConectorAreaInteres new.

c11 initialize:'cPasillo-B' Desde:Pasillo Hasta:B.

c2:= ConectorAreaInteres new.

c2 initialize:'cD-Pasillo' Desde:D Hasta:Pasillo.

c22:= ConectorAreaInteres new.

c22 initialize:'cPasillo-D' Desde:Pasillo Hasta:D.

c3:= ConectorAreaInteres new.
c3 initialize:'cPasillo-A' Desde:Pasillo Hasta:A.
c33:= ConectorAreaInteres new.
c33 initialize:'cA-Pasillo' Desde:A Hasta:Pasillo.

c4:= ConectorAreaInteres new.
c4 initialize:'cA-C' Desde:A Hasta:C.
c44:= ConectorAreaInteres new.
c44 initialize:'cC-A' Desde:C Hasta:A.

conectores:= OrderedCollection new.

conectores add: c1.
conectores add: c2.
"conectores add: c3." "suponemos que no esta disponible"
conectores add: c4.
conectores add: c11.
conectores add: c22.
conectores add: c33.
conectores add: c44.

movTAI := TransitarAreasInteres new.
movTAI setConectoresAreaInteres: conectores.

(sala getMovilidades) add: movTAI.

*****puntos de Acceso para la movilidad Transitar Areas Interes *****"

paAI1:= PuntoDeAcceso new.
paAI1 setPim:cuadro.
paAI1 setUnidadDesplazamiento:c4.
paAI1 setMovilidad: movTAI.
paAI1 setContenidoEn:sala.

(cuadro getPuntosAcceso) add: paAI1.

*****Movilidad Transitar Areas Logicas *****"

aA := AreaLogica new.
aA setNombre: 'areaA'.
aB := AreaLogica new.
aB setNombre: 'areaB'.
aC := AreaLogica new.
aC setNombre: 'areaC'.
aD := AreaLogica new.
aD setNombre: 'areaD'.
aNula:= AreaLogicaNula new.

cAB:= ConectorAreaLogica new.
cAB initialize:'de-A-B' Desde:aA Hasta:aB.
cBA:= ConectorAreaLogica new.
cBA initialize:'de-B-A' Desde:aB Hasta:aA.
cAD:= ConectorAreaLogica new.
cAD initialize:'de-A-D' Desde:aA Hasta:aD.
cDA:= ConectorAreaLogica new.
cDA initialize:'de-D-A' Desde:aD Hasta:aA.
cBD:= ConectorAreaLogica new.
cBD initialize:'de-B-D' Desde:aB Hasta:aD.
cDB:= ConectorAreaLogica new.
cDB initialize:'de-D-B' Desde:aD Hasta:aB.
cCD:= ConectorAreaLogica new.
cCD initialize:'de-C-D' Desde:aC Hasta:aD.
cDC:= ConectorAreaLogica new.
cDC initialize:'deD-C' Desde:aD Hasta:aC.

conectores:= OrderedCollection new.

conectores add: cAB.
conectores add: cBA.
conectores add: cAD.
conectores add: cDA.
conectores add: cBD.
conectores add: cDB.
conectores add: cCD.
conectores add: cDC.

movTAL := TransitarAreasLogicas new.
movTAL setConectoresAreaLogica: conectores.

paAL1:= PuntoDeAcceso new.
paAL1 setPim:cuadro.
paAL1 setUnidadDesplazamiento:cBD.
paAL1 setMovilidad: movTAL.
paAL1 setContenidoEn: sala.

(sala getMovilidades) add: movTAL.
(cuadro getPuntosAcceso) add: paAL1.

“*****Creacion del usuario y su ubicación *****”

u:= Usuario new.
ubicacion := UbicacionUsuario new.
ubicacion setPIMovilidad: sala.

ubicacion setUnidadDesplazamiento: c2.
(ubicacion getMovilidades) at: movTAL put: cCD.
u setUbicacionUsuario: ubicacion.
s setUsuario:u.

s calcularMejorCaminoPara:u Hasta:cuadro.
(u getCaminoSugerido) getTramos.