



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

**Departamento de Computación**

**Mecanismos para la localización de los recursos del  
entorno cercano a un colaborador**

Tesis que presenta

**Tanibet Pérez de los Santos Mondragón**

para obtener el Grado de

**Maestro en Ciencias en Computación**

Directora de tesis

**Dra. Sonia Guadalupe Mendoza Chapa**



## Resumen

Este trabajo de investigación se inscribe en las áreas del Trabajo Colaborativo Asistido por Computadora (TCAC) y del Cómputo Ubicuo (CU). El área de TCAC se encarga de estudiar la colaboración, la coordinación y la comunicación de un grupo de personas y de facilitar estas actividades mediante tecnologías computacionales. Por otro lado, el área de CU se enfoca en simplificar la interacción entre el usuario y la tecnología para que el primero la use sin siquiera pensar en ella. De esta manera, es posible obtener una conectividad de tipo “en todo momento y en todo lugar”.

Cuando una persona llega a una organización que desconoce, le puede resultar difícil encontrar alguna oficina o recurso que necesita debido a que, la mayoría de las veces, los edificios no cuentan con los señalamientos necesarios para guiar a sus visitantes. Además resulta imposible (en términos de espacio, costo y beneficio) concentrar todos los recursos físicos de la organización en un sólo espacio. De hecho, dichos recursos generalmente están distribuidos en los diferentes edificios de la organización con el fin de para soportar la delegación de tareas y la especialización.

En caso de que varios colaboradores necesiten interactuar con un recurso al mismo tiempo, la coordinación de todos ellos puede volverse una tarea ardua. Sin embargo, con el fin de resolver estos problemas comunes, esta tarea puede automatizarse mediante la utilización de tecnología. La creación de un soporte informático para determinar la ubicación de recursos constituye la base de un entorno ubicuo, que es indispensable para hacer más eficiente el trabajo de los miembros de la organización.

Este trabajo de tesis propone diseñar e implementar un sistema para facilitar la localización de recursos distribuidos, inmersos en un ambiente cooperativo ubicuo. Para representar un edificio, se diseñó un modelo topológico del mismo que consiste en: a) un grafo cuyos nodos representan celdas y cuyas aristas denotan la distancia que existe entre dichos nodos y b) celdas que a su vez representan cuartos o corredores. La distancia entre un usuario y un recurso es calculada por el método de la distancia de recorrido, el cual determina la distancia real del recorrido en vez de la distancia Euclidiana.

Finalmente, debido a que diversos recursos pueden satisfacer una solicitud, se propone un mecanismo para determinar el recurso más cercano al colaborador solicitante. Además, con el fin de que este pueda tener una mejor interpretación de la ubicación del recurso, la información correspondiente es desplegada mediante mapas en formato vectorial accesibles desde un navegador Web.



## Abstract

The present research work falls within the realm of the Computer-Supported Cooperative Work (CSCW) and Ubiquitous Computing (UC) fields. The CSCW field is in charge of studying the collaboration, coordination and communication of a group of persons and facilitating these activities by means of computational technologies. On the other hand, the UC field is focused on simplifying the interaction between the user and the technology so that he/she uses it without thinking about it. In this way, it is possible to obtain a connectivity of type “every moment everywhere”.

When a person arrives at an organization he/she does not know, it could be difficult for him/her to find some office or resource he/she needs because, most of the time, the buildings do not have the needed signals to guide its visitors. In addition, it results impossible (in terms of space, cost and benefit) to concentrate all the organization physical resources into a unique place. In fact, such resources are generally distributed on the different buildings of the organization in order to support task delegation and specialization.

If several collaborators need to interact with a resource at the same time, the coordination among them can become an arduous task. However, in order to solve these common problems, this task can be automated by using the technology. Creating a computational support to determine the location of resources constitutes the basis of an ubiquitous environment, which is indispensable to make the organization staff’s work more efficient.

This thesis work propose to design and implement a system to facilitate the location of distributed resources, immersed in a ubiquitous cooperative environment. To represent a building, we design a topological model of it that consists in: a) a graph whose nodes represent cells and whose edges denote the distance among such nodes and b) cells which in turn represent rooms or corridors. The distance between a user and a resource is calculated by the walking distance method, which determines the actual walking distance instead of the Euclidean distance.

Finally, due to several resources can satisfy a request, we provide a mechanism to determine the resource closest to the requesting collaborator. Moreover, in order to provide her/him with a better interpretation of the resource location, the corresponding information is displayed using vector maps accessible from a Web browser.



## Agradecimientos

# Al CINVESTAV-IPN y al CONACYT

Por brindarme la oportunidad y los recursos para poder estudiar un posgrado.

## A mis maestros

Gracias por  
todos los consejos,  
secretos de profesión  
compartidos, por los regaños  
y por todo el conocimiento  
compartido. Es por ustedes  
que en la actualidad soy  
una gran profesionista  
y una gran persona.

## A la Dra. Sonia

Us-  
ted  
inaugura  
una catego-  
ría que denomino  
"maestro fascinante"  
porque tiene la sensibilidad  
de hablar a los corazones  
de sus alumnos, la dedicación  
de explotar nuevas capacidades  
en los estudiantes  
para poder realizar  
un excelente  
trabajo  
de tesis

## A mi familia

Los créditos de finalizar  
 una maestría teniendo dos niños, no  
 son del todo míos. Mi familia siempre me ha  
 apoyado, siempre ha estado pendiente de mi y de  
 mis hijos. Gracias mamá y papá por creer en mi y por  
 su apoyo constante, este logro lo podemos compartir  
 todos, ya que sin ustedes no hubiera sido posible. A  
 mis hermanos Ara, Miguel Angel y Mauricio por  
 su compañía y su apoyo y por los momentos tan  
 buenos que hemos pasado juntos. A mis hijos  
 Alejandro y Elena ustedes son mi fuerza de  
 voluntad, el motor que hace que mi vi-  
 da funcione, ustedes son la razón  
 de querer seguir adelante  
 y superarme como  
 persona  
 ♥

## A mis amigos

Por  
 to-  
 dos  
 los  
 buenos  
 momentos,  
 palabras de apoyo y compañerismo mostrado.  
 Siempre es bueno saber que hay perso-  
 nas con las que puedes contar  
 para reír, para platicar  
 y más que nada  
 para echarle montón  
 a las interminables  
 tareas y tra-  
 bajos de  
 la es-  
 cue- la.



# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto de investigación	1
1.1.1	Trabajo Cooperativo Asistido por Computadora	1
1.1.2	Cómputo Ubicuo	3
1.2	Planteamiento del problema	6
1.3	Objetivos del proyecto	7
1.4	Organización del documento	8
<b>2</b>	<b>Marco teórico</b>	<b>11</b>
2.1	Grafos	11
2.2	Sistemas de Información Geográfica	13
2.2.1	Estructura de la información en un SIG	14
2.2.2	Estandarización de los SIG	15
2.3	Camino más corto: algoritmo de Dijkstra	17
2.3.1	Representación del camino más corto	19
2.3.2	Algoritmo de Dijkstra	20
<b>3</b>	<b>Trabajos relacionados</b>	<b>23</b>
3.1	Servicios basados en localización	23
3.2	Ubicación geográfica de un dispositivo móvil	25
3.2.1	Red telefónica celular	26
3.2.2	Satélites	27
3.2.3	Estaciones de corto alcance	27
3.3	Trabajos relacionados	28
3.3.1	NearMe	29
3.3.2	Sistema <i>Active Badge</i>	29
3.3.3	Sistema <i>The Bat Ultrasonic Location</i>	30
3.3.4	Sistema Cricket	31
3.3.5	Sistema HiBall tracker	33
3.3.6	Sistema RADAR	34
3.3.7	Sistema WHISPER	35

3.4	Comparación de trabajos relacionados	36
<b>4</b>	<b>Análisis y diseño del sistema NOTU</b>	<b>39</b>
4.1	Análisis	40
4.1.1	Análisis del problema	40
4.1.2	Análisis del modelo topológico	41
4.1.3	Análisis del problema al calcular las distancias	41
4.2	Diseño	43
4.2.1	Diseño del modelo topológico	43
4.2.2	Diseño de mapas	44
4.3	Arquitectura del sistema NOTU	46
4.4	Requerimientos del sistema NOTU	47
4.4.1	Descripción de módulos del sistema NOTU	49
4.4.2	Módulo de publicación	49
4.4.3	Módulo de búsqueda	50
4.4.4	Módulo de modificación de un recurso	53
4.4.5	Módulo de búsqueda del recurso más cercano	54
<b>5</b>	<b>Implementación del sistema NOTU</b>	<b>57</b>
5.1	Herramientas para la generación de mapas	57
5.2	Herramientas para el servidor de mapas	60
5.2.1	Interacción con el mapa	61
5.2.2	Gestión de la base de datos	62
5.2.3	Soporte de encaminamiento en la base de datos	64
5.2.4	Implementación del algoritmo de Dijkstra	67
5.3	Interacción de las herramientas	68
5.4	Diagrama entidad-relación	71
5.4.1	Entidad Geometry_Columns	71
5.4.2	Entidad Puntos_Alta	73
5.4.3	Entidad Puntos_Baja	73
5.4.4	Entidad Celdas_Alta	73
5.4.5	Entidad Celdas_Baja	74
5.4.6	Entidad Distancias_Alta	74
5.4.7	Entidad Recursos	74
5.4.8	Entidad Grafo	75
5.4.9	Entidad Celdas_Nodos_baja	75
5.5	Descripción del ambiente gráfico	75
5.5.1	Interfaz de usuario	76
5.5.2	Visualización de mapas	77
5.5.3	Publicación de un recurso	79
5.5.4	Modificación de un recurso	82

## ÍNDICE GENERAL

xi

5.5.5	Búsqueda de un recurso	84
5.6	Pruebas	87
5.6.1	Caso de estudio	88
5.6.2	Prueba del sistema NOTU	88
<b>6</b>	<b>Conclusiones y trabajo a futuro</b>	<b>93</b>
6.1	Conclusiones	93
6.2	Trabajo a futuro	96



# Índice de figuras

1.1	Matriz tiempo-espacio de los sistemas colaborativos . . . . .	2
1.2	Evolución del Cómputo Distribuido, Móvil y Ubicuo . . . . .	3
1.3	Relación del Cómputo Móvil y Ubicuo con otras áreas . . . . .	5
1.4	Organización del manuscrito de tesis . . . . .	9
2.1	Ejemplo de un grafo multi-arista . . . . .	12
2.2	Definición de Sistema de Información Geográfica (SIG) . . . . .	13
2.3	Ejemplo de un mapa representado con el modelo <i>raster</i> . . . . .	15
2.4	Ejemplo de un mapa con capas superpuestas . . . . .	16
2.5	Intersecciones para encontrar el camino más corto . . . . .	18
2.6	Ruta más corta del punto A al punto B . . . . .	19
2.7	Ejecución del algoritmo de Dijkstra . . . . .	22
3.1	Sistema Bat . . . . .	31
3.2	Sistema Cricket . . . . .	33
3.3	Sistema HiBall . . . . .	34
4.1	Ejemplo de la distancia más corta contra la distancia de recorrido . . . . .	42
4.2	Plano de la planta baja del edificio del Departamento de Computación . . . . .	44
4.3	Planta baja del edificio del Departamento de Computación con el grafo. . . . .	45
4.4	Mapa vectorial de la planta baja del Departamento de Computación . . . . .	46
4.5	Arquitectura del sistema . . . . .	48
4.6	Diagrama de casos de uso de la publicación de un recurso . . . . .	50
4.7	Diagrama de secuencia de la publicación de un recurso . . . . .	51
4.8	Diagrama de casos de uso de la búsqueda de un recurso . . . . .	52
4.9	Diagrama de secuencia del módulo de búsqueda. . . . .	53
4.10	Diagrama de casos de uso de la modificación de un recurso . . . . .	54
4.11	Diagrama de secuencias de la modificación de un recurso . . . . .	55
5.1	Mapa vectorial de la planta baja del Departamento de Computación . . . . .	58
5.2	Ejemplo del comando <code>shp2psql</code> . . . . .	59
5.3	Ejemplo de un archivo SQL . . . . .	59

5.4	Ejemplo de un archivo <code>mapfile</code> . . . . .	61
5.5	Ejemplo de la librería <code>OpenLayers</code> . . . . .	63
5.6	Comandos para agregar soporte espacial a una base de datos . . . . .	64
5.7	Tabla <code>celdas_alta</code> . . . . .	64
5.8	Pasos para añadir los atributos origen, destino y distancia . . . . .	66
5.9	Creación de la topología de datos . . . . .	67
5.10	Resultado de la consulta usando la función <code>shortest_path</code> . . . . .	69
5.11	Interacción de las herramientas . . . . .	70
5.12	Diagrama entidad-relación de la base de datos <code>routing2</code> . . . . .	72
5.13	Vista general de la interfáz gráfica del sistema NOTU . . . . .	76
5.14	Control que muestra el estado de las capas . . . . .	78
5.15	Control <code>minimapa</code> . . . . .	79
5.16	Controles que facilitan la interacción con el mapa . . . . .	80
5.17	Pantalla de publicación de un recurso . . . . .	81
5.18	Pantalla de modificación de un recurso . . . . .	83
5.19	Cuadro de texto con información de un recurso . . . . .	84
5.20	Cuadro de texto para modificar un recurso . . . . .	85
5.21	Pantalla de búsqueda de un recurso . . . . .	86
5.22	Resultado de la búsqueda del recurso más cercano . . . . .	87
5.23	Resultado 1 de la búsqueda del recurso más cercano . . . . .	89
5.24	Resultado 2 de la búsqueda del recurso más cercano . . . . .	90

# Índice de tablas

3.1	Tabla comparativa de trabajos relacionados . . . . .	37
5.1	Categorías de recursos . . . . .	91





# Capítulo 1

## Introducción

### 1.1 Contexto de investigación

En la sección 1.1.1 se describe el área del Trabajo Cooperativo Asistido por Computadora, así como los objetivos y las consideraciones de tiempo y espacio de los sistemas colaborativos. En la sección 1.1.2 se describe el área del Cómputo Ubicuo y se analiza su proceso evolutivo y su relación con otras áreas de las ciencias computacionales.

#### 1.1.1 Trabajo Cooperativo Asistido por Computadora

Las personas interactúan en nuevas y diversas maneras, las cuales surgen acorde con los avances tecnológicos de las computadoras y de las diversas formas de comunicación electrónica, e.g., sistemas que integran el procesamiento de información compartida y las actividades de comunicación de un grupo. El estudio de estos sistemas es una parte del campo de estudio multidisciplinario denominado Trabajo Colaborativo Asistido por Computadora (TCAC) [Greif, 1988].

El término “sistema colaborativo” (*“groupware”*) es usado frecuentemente como un sinónimo de las tecnologías de TCAC. Según Ellis et al. [Ellis et al., 1991] TCAC se encarga de estudiar cómo trabaja un grupo de colaboradores y de descubrir cómo la tecnología (especialmente computacional) puede ayudarlos.

El objetivo de los sistemas colaborativos es ayudar a un grupo de personas a comunicar, colaborar y coordinar sus actividades. La comunicación es un proceso mediante el cual las personas interactúan entre sí usando el intercambio de mensajes. Por su parte, la colaboración es un punto importante en las actividades grupales, ya que demanda que las personas compartan información. La eficacia de la comunicación y de la colaboración puede ser mejorada si las actividades del grupo son coordinadas. De esta manera, la coordinación puede verse como una actividad que debe ser efectuada cuando varios

participantes realizan una tarea común.

No existe una división estricta entre los sistemas que son considerados como colaborativos y los que no lo son. Por tal motivo, se puede clasificar un sistema colaborativo conforme a la manera en que resuelve los problemas en un entorno compartido. La mayoría de las veces, los sistemas pueden migrar de un punto a otro de la clasificación, ya que en la actualidad las tecnologías han evolucionado y las interfaces del usuario han ampliado dicha clasificación.

Un sistema colaborativo puede ayudar a grupos que interactúan cara a cara o que están distribuidos en varios lugares. Además, un sistema colaborativo puede mejorar la comunicación y la colaboración entre los usuarios, aún cuando su interacción sea en diferentes momentos (asíncrona). Estas consideraciones de tiempo y espacio sugieren cuatro categorías de sistemas colaborativos que se muestran en la figura 1.1. Un sistema colaborativo global debe cubrir las necesidades de todos los cuadrantes.

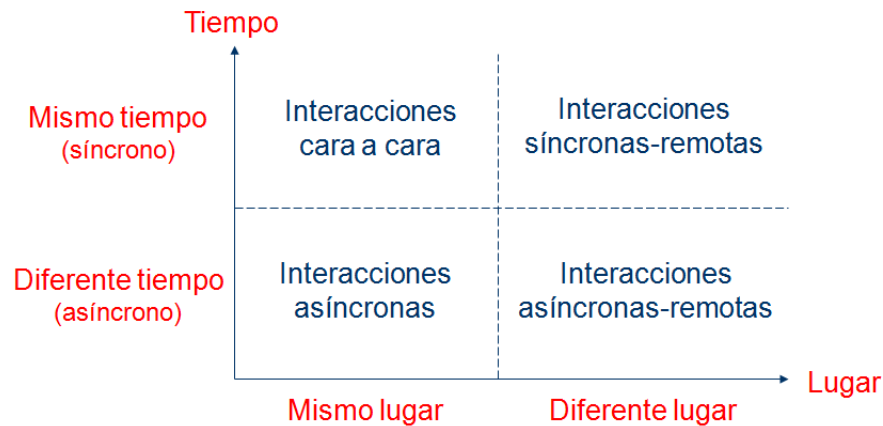


Figura 1.1: Matriz tiempo-espacio de los sistemas colaborativos

Algunas de las aplicaciones más conocidas de los sistemas colaborativos son: los sistemas de mensajes (e.g., AlphaMail [Kay, 2007]), los editores multiusuario (e.g. AllianceWeb [Decouchant et al., 2010]), los sistemas de soporte de decisión grupal (e.g., OpenSpace-online [Ender, 2009]), los espacios de media (e.g., JamSpace [Gurevich, 2006]), los sistemas de flujo de trabajo (e.g., Kepler/Pegasus [Mandal et.al, 2007]) y las herramientas de gestión de proyectos (e.g., P3M3 [Maqsood and Javed, 2007]).

### 1.1.2 Cómputo Ubicuo

El término Ubicuo (u omnipresente) [Weiser, 1993] puede ser definido como “algo que existe o está en todas partes al mismo tiempo, que puede ser encontrado constantemente y que es de uso cotidiano”. Cuando este término es utilizado en el ámbito computacional, implica que la tecnología se encuentra en todas partes y que puede ser utilizada en todo momento. Debido a la penetrabilidad de esta tecnología, se puede utilizar sin pensar en ella, i.e., un usuario se enfoca en la tarea a realizar de tal manera que la tecnología se vuelva realmente invisible. La tecnología ubicua es a menudo inalámbrica, móvil o conectada por medio de una red cableada, creando una mejor conexión entre los usuarios y el mundo alrededor de ellos.

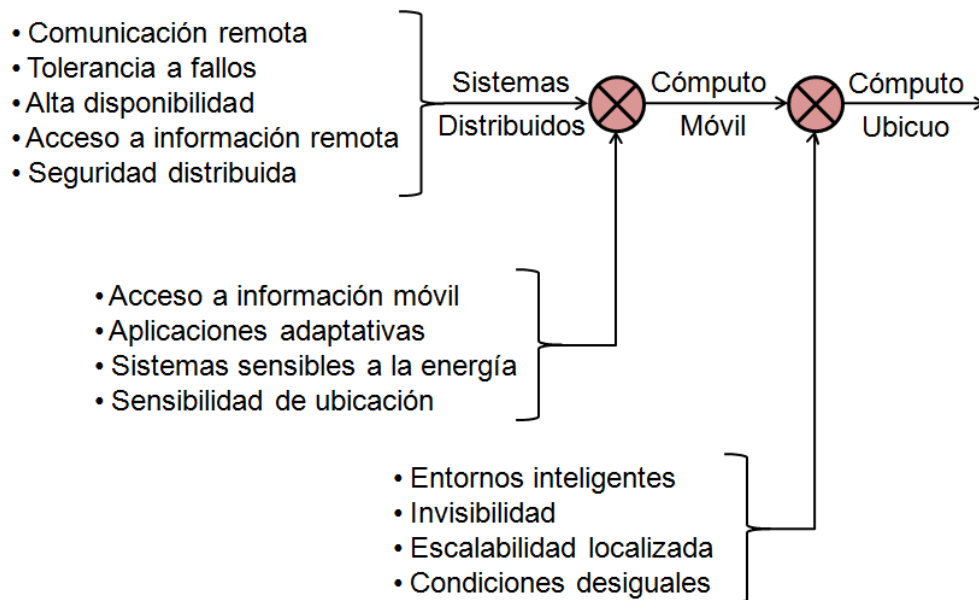


Figura 1.2: Evolución del Cómputo Distribuido, Móvil y Ubicuo

En la figura 1.2 se puede observar la evolución de las problemáticas que resuelven los Sistemas Distribuidos. La complejidad de dichas problemáticas aumenta de izquierda a derecha, i.e., la resolución de problemas anteriormente encontrados resulta más compleja cuando un área de investigación se inclina a otra. En la figura 1.2 se observan símbolos de cruz, estos representan la tendencia de los Sistemas Distribuidos hacia el Cómputo Móvil y Ubicuo e indican que el aumento de la complejidad es multiplicativo.

El Cómputo Móvil y Ubicuo ha dado los mayores pasos evolutivos en la línea de investigación de los Sistemas Distribuidos que comienza a mediados de los años 70s. En

dicha evolución se ha encontrado nuevos problemas que resultan más complejos, ya que es más difícil diseñar e implementar un sistema de *Cómputo Móvil* que un sistema distribuido, comparando el grado de robustez y madurez que necesita cada uno. Un sistema de *Cómputo Ubicuo* es por lo tanto aún más desafiante.

Los *Sistemas Distribuidos*, desde sus inicios, han dado solución a los problemas de acceso a información y recursos remotos, tolerancia a fallos, alta disponibilidad y seguridad. Cuando los *Sistemas Distribuidos* se integran al *Cómputo Móvil* se obtiene, de manera transparente, acceso a información móvil, aplicaciones adaptativas, sistemas sensibles a la energía y sensibilidad de ubicación. Estas tecnologías cambian la naturaleza general del *Cómputo Móvil*, permitiendo que la mayoría de los objetos que se encuentran en la vida diaria sean “concientes” de las necesidades del usuario.

En el año 2000, las investigaciones en el *Cómputo Móvil* comenzaron a abordar problemáticas que se pueden identificar dentro del campo de acción del *Cómputo Ubicuo*. La esencia del *Cómputo Ubicuo* es la creación de entornos saturados de comunicación.

El *Cómputo Ubicuo* pretende hacer nuestras vidas más simples a través de ambientes digitales que son sensitivos, adaptativos y responden a las necesidades humanas. Al integrar soporte *Ubicuo* en el *Cómputo Móvil* se obtiene una conectividad del tipo “en todo momento y en todas partes” [Saha and Mukherje, 2003], además de resolver problemáticas como acceso a la información móvil, aplicaciones adaptativas, sistemas sensibles a la energía y sensibilidad de ubicación.

En hardware se cuenta con dispositivos móviles, sensores e incluso utensilios inteligentes que ofrecen una plataforma *ubicua*. También, se puede adoptar tecnologías de software que incluyan procesamiento digital de señales y programación orientada a objetos para facilitar la implementación de *Sistemas Ubicuos*.

Las propiedades que se deben cubrir para que las aplicaciones puedan “desaparecer” en el entorno son la escalabilidad, la heterogeneidad, la integración de los componentes en una sola plataforma y una intervención mínima de los humanos. También, los sistemas y dispositivos deben percibir el entorno y poder usar esta información de manera efectiva.

En la figura 1.2 se pueden observar las problemáticas que surgen cuando los *Sistemas Distribuidos* y el *Cómputo Móvil* se inclinan hacia el *Cómputo Ubicuo*. A continuación se detallan dichas problemáticas:

- **Entornos inteligentes:** existen cuando las computadoras se incrustan en la infraestructura de un edificio; mediante esta fusión, se puede detectar y controlar estos dos mundos.

- **Invisibilidad:** se crea cuando un ambiente de Cómputo Ubicuo desaparece totalmente del conocimiento del usuario; cuando el ambiente cumple las expectativas del usuario y pocas veces ofrece sorpresas, entonces permite una interacción a un nivel casi subconsciente.
- **Escalabilidad localizada:** la escalabilidad en la detección es un problema crítico del Cómputo Ubicuo; los sistemas bien diseñados deben ser escalables, reduciendo rigurosamente la interacción entre entidades distantes.
- **Eliminar las condiciones desiguales:** existen grandes diferencias entre el nivel de “inteligencia” de diferentes ambientes ubicuos; al momento de integrarlos se debe reducir estas variaciones de “inteligencia”, compensando los ambientes ubicuos menos “inteligentes”.

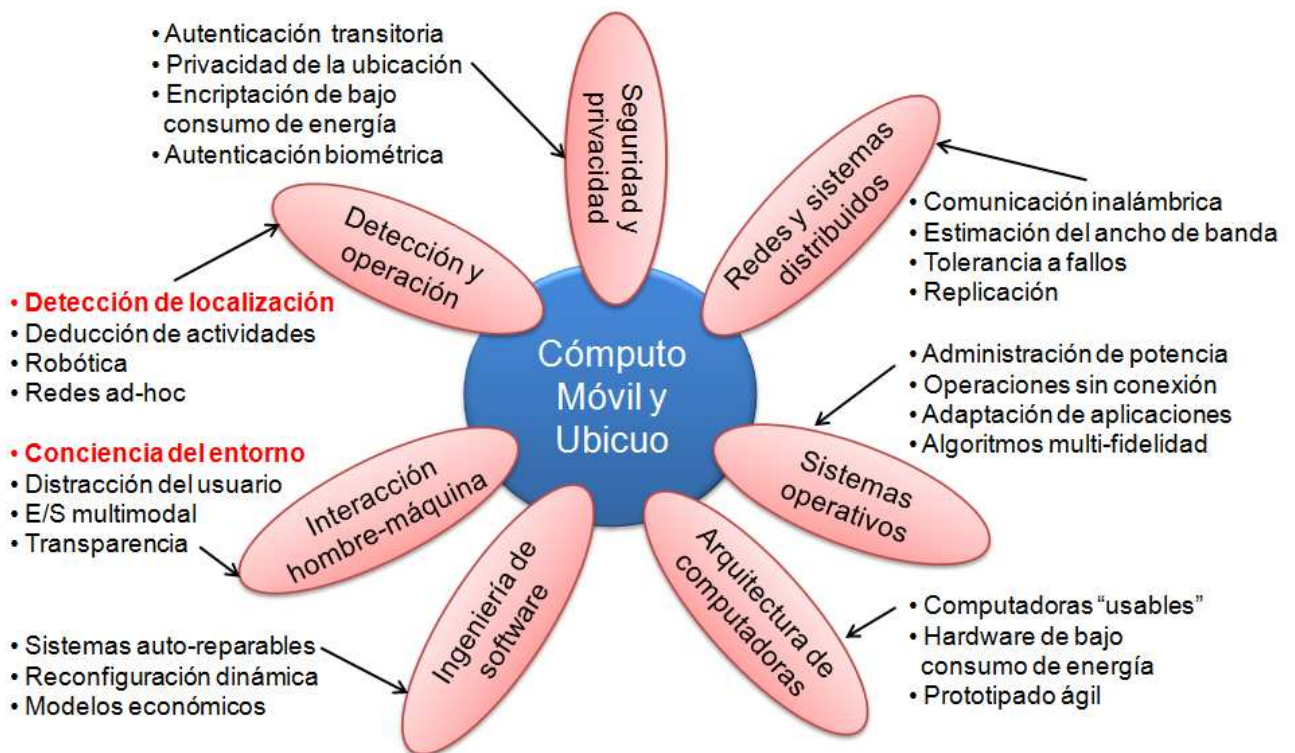


Figura 1.3: Relación del Cómputo Móvil y Ubicuo con otras áreas

La figura 1.3 presenta las problemáticas que surgen cuando el Cómputo Móvil y Ubicuo interactúa con otras áreas de investigación, algunos de ellos son:

- **Cómputo Ubicuo libre de distracciones:** se necesita explorar el uso de métodos proactivos y autoajustables para reducir la distracción del usuario; el carácter proactivo se refiere a la habilidad de una capa del sistema para anticipar peticiones de una capa superior; en tanto que el carácter autoajutable se refiere a la habilidad del sistema para observar las demandas y, en respuesta, ajustar su rendimiento y el uso de sus recursos.
- **Movilidad transparente usando hardware Ubicuo:** se basa en el principio suspender-continuar aplicado a la movilidad en hardware; el objetivo es suspender las tareas que se realizan en una computadora y guardar la información de dichas tareas en algún sitio de Internet, viajar a algún otro lugar y después, de manera transparente, continuar trabajando en otra computadora.
- **Computadoras usables:** se obtienen cuando se integran múltiples sensores a dispositivos de cómputo y también cuentan con aplicaciones que crean un entorno social, con la finalidad de anticipar las necesidades de los usuarios e incrementar la interacción entre ellos.
- **Configuración de una llave segura entre los dispositivos Ubicuos:** la proliferación de teléfonos móviles equipados con cámaras constituye una plataforma poderosa, que puede ser utilizada para proveer autenticación entre dispositivos sin necesidad de una autoridad certificadora.

## 1.2 Planteamiento del problema

Cuando una persona entra a una organización que desconoce resulta una tarea difícil ubicar un lugar, encontrar algún colega o un recurso que necesite. La mayoría de las veces los edificios no cuentan con los señalamientos correspondientes en las oficinas, pasillos o salones. Además resulta imposible (en espacio y costo) concentrar todos los recursos físicos en un sólo punto, por lo que es necesario distribuirlos por toda la organización.

En consecuencia, las grandes organizaciones (varios edificios) requieren un soporte informático para determinar la ubicación de recursos físicos costosos (e.g., impresoras y salas de juntas) y su cercanía relativa a un usuario solicitante. Mediante la creación de un soporte informático para facilitar la ubicación de recursos, se construye la base de entorno Ubicuo que es indispensable para:

- hacer más eficiente el trabajo de los miembros de una organización, e.g., trabajo de oficina;
- explotar los recursos subutilizados de la organización;
- conocer todos los recursos que existen en una organización; y

- saber rápidamente la ubicación precisa de cada recurso.

Cuando un usuario hace una petición, particularmente de recursos físicos, se puede encontrar más de un recurso que satisfaga las características técnicas especificadas en su petición. Para determinar cuál de estos recursos se encuentra más cerca del usuario, se propone implementar un sistema de localización. El sistema de localización del recurso más cercano toma en cuenta:

- la ubicación de los recursos que satisfacen los requerimientos técnicos de una petición;
- la ubicación del usuario que realiza la petición; y
- la información sobre las distancias que existen entre las oficinas de la organización.

La información sobre la ubicación de los recursos es proporcionada por los usuarios cuando publican sus recursos en el sistema. Por su parte, la información sobre la ubicación del usuario se obtiene de un sistema de reconocimiento de caras [García, 2009]. Por último, la información sobre las distancias entre las oficinas de la organización se conoce desde que se instala el sistema en dicha organización.

Los servicios de localización ayudan a un usuario a encontrar: lugares, servicios, productos, recursos y miembros de la organización. Estos servicios están enfocados en determinar la posición actual del usuario, en poder localizar el punto inicial del recorrido y en dar una guía de cómo llegar al recurso solicitado. Pero esta localización debe ser precisa ya que si existe un error en la ubicación, e.g., 10 metros, entonces el sistema localizará a un recurso en otro salón o incluso en otro edificio.

## 1.3 Objetivos del proyecto

### General

Diseñar mecanismos para determinar la ubicación del recurso solicitado, el cual debe ser el más cercano de la posición actual del colaborador solicitante y debe estar disponible.

### Particulares

Diseñar y desarrollar:

- mecanismos para determinar la ruta más corta a recursos físicos informáticos y no informáticos,

- un sistema que facilite la definición de la ubicación de recursos en mapas,
- un modelo topológico del edificio del Departamento de Computación del CINVESTAV-IPN para poder representar los recursos y las oficinas,
- un sistema Web que permita visualizar el sistema tanto en PC como en dispositivos móviles.

Otro objetivo particular es realizar una interfaz entre el sistema propuesto y el sistema de reconocimiento de caras, la integración de dichos sistemas provee de manera automática la ubicación del usuario debido a que el sistema de reconocimiento de caras permite saber en todo momento la ubicación de un usuario.

## 1.4 Organización del documento

El trabajo de tesis está compuesto por seis capítulos (figura 1.4).

En el capítulo 2 se da un marco teórico general de los temas que comprende este trabajo de tesis. Básicamente se abarcan los conceptos de Grafo, Sistema de Información Geográfica (SIG) así como el algoritmo de Dijkstra que se usa para resolver el problema del camino más corto.

El capítulo 3 aborda las investigaciones relacionadas con este trabajo de tesis. Cuenta con un análisis de los trabajos, mecanismos y sistemas más populares que se relacionan con el problema que se está abordando. Posteriormente, se realiza un análisis comparativo de todos los trabajos antes estudiados.

El capítulo 4 trata del análisis y diseño de sistema propuesto en esta tesis. El problema se divide en dos partes: 1) poder distinguir un recurso de otro cuando están en pisos diferentes y 2) evitar que un usuario atraviese un muro. Para resolver estos problemas se analizan mecanismos y técnicas como el análisis del modelo topológico. Además se describe el diseño del sistema propuesto así como el de sus módulos.

En el capítulo cinco 5 describe: 1) los pasos para la implementación del sistema propuesto, 2) las herramientas utilizadas así como sus respectivas configuraciones, 3) la base de datos geográfica mediante un diagrama entidad-relación y 4) la interfaz de usuario contemplando todas las actividades que puede hacer un usuario.

El capítulo 6 se refiere a las conclusiones y al trabajo futuro.





Figura 1.4: Organización del manuscrito de tesis



# Capítulo 2

## Marco teórico

En este capítulo se abarcan los diferentes conceptos que se usan a lo largo de este trabajo de tesis. Primero se describe el término grafo, sus propiedades, características y atributos. Posteriormente se menciona la definición de Sistemas de Información Geográfica, la estructura que manejan así como sus estándares. Por último se define el problema de la ruta más corta y el algoritmo de Dijkstra que pretende solucionar dicho problema. También se describe una implementación del algoritmo de Dijkstra.

### 2.1 Grafos

Muchas estructuras relacionadas con situaciones del mundo real pueden ser convenientemente representadas en papel, mediante un diagrama compuesto por un conjunto de puntos (usualmente dibujados como pequeños círculos o puntos) junto con líneas o curvas que unen pares de puntos. Por ejemplo, en un diagrama los puntos pueden representar diferentes ciudades de un país y una línea que une dos puntos  $A$  y  $B$  pero no pasa a través de un tercer punto  $C$ , puede indicar la existencia de un servicio aéreo directo entre dos ciudades.

En algunos casos ocurre que hay un servicio aéreo directo de  $A$  hacia  $B$  pero no de  $B$  hacia  $A$ . En tales situaciones una flecha (una línea o curva dirigida) es dibujada de  $A$  hacia  $B$ , de tal modo que la línea que une  $A$  y  $B$  se vuelve orientada o dirigida. Una abstracción matemática de estas estructuras, que involucran puntos y líneas, conduce al concepto de grafo.

Un **grafo**  $G$  [Balakrishnan, 1997] es un conjunto  $V$  de **vértices** (o **nodos**) y una colección  $E$  de pares no ordenados de vértices llamada **aristas** (o **lados**). Un grafo es representado simbólicamente como  $G = (V, E)$ . El **orden** de un grafo es el número de sus vértices y su **tamaño** es el número de sus aristas.

Si  $u$  y  $v$  son dos nodos de un grafo y si el par no ordenado  $\{u, v\}$  es una arista denominada  $e$ , se puede decir que  $e$  une a  $u$  y  $v$  o que es una arista entre  $u$  y  $v$ .

Un **multi-grafo** [Gross et al., 2004] es un grafo que está facultado para tener **multi-arista**, i.e., aristas que relacionan los mismos nodos. De esta forma, dos nodos pueden estar conectados por más de una arista. Algunos autores también denominan multi-grafo a un grafo que no tiene ciclos.

Un **ciclo** es una arista que tiene el mismo vértice repetido, i.e., que une un par no ordenado en el cual los dos elementos son iguales.

Un **grafo simple** es un grafo que no tiene aristas paralelas ni ciclos.

Ejemplo de un multi-grafo: la figura 2.1 es una línea dibujada que representa un grafo  $G = (V, E)$ . El conjunto de los vértices está definido por  $V = \{1, \dots, 7\}$  y el conjunto de aristas se define por  $E = \{a, b, c, d, e, f\}$ . Como se puede apreciar el conjunto  $\{a, b\}$  es denominado multi-arista. La arista  $c$  representa un ciclo.

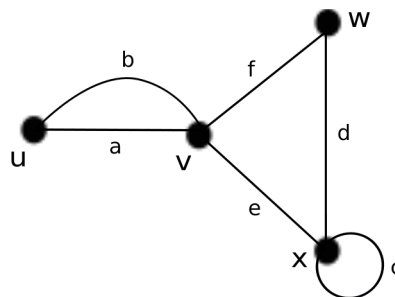


Figura 2.1: Ejemplo de un grafo multi-arista

Un grafo puede tener atributos adicionales a los vértices y a las aristas, lo cual resulta conveniente para representar modelos matemáticos que son usados en el proceso de creación de aplicaciones. Dos de los atributos adicionales más comunes de las aristas son la dirección y el peso. Otro atributo tanto para nodos como para aristas es el color. Un atributo de vértice es una función del conjunto de los vértices a algún conjunto de valores. Análogamente, un atributo de arista es una función de un conjunto de aristas a algún conjunto de valores.

Un **digrafo** (o grafo dirigido) es un grafo cuyas aristas son dirigidas, i.e., cada arista posee un vértice inicial y uno final. Existe cuando una arista entre dos vértices crea una conexión en varios sentidos a la vez. La asignación de una dirección a una arista crea

un sentido de “hacia adelante” y otro “hacia atrás”. Una **arista dirigida** es una arista que posee un nodo final que está designado como la cola, en tanto que un nodo inicial es denominado como la cabeza [Wallis, 2007].

## 2.2 Sistemas de Información Geográfica

Los Sistemas de Información Geográfica, conocidos como SIG o GIS <sup>1</sup>, están cada vez más presentes en la sociedad; aplicaciones como Google Maps son un claro ejemplo.

Deunker y Kjerne [Deuker and Kjerne, 1989] a finales de los años 90 dan una definición: un SIG es un sistema de hardware, software, personas, organizaciones, almacenaje, análisis y distribución de información de territorios de la tierra.

Posteriormente en el reporte técnico de Egrnhofer y otros [Egenhofer et al., 1991] se define SIG como un sistema de hardware, software y procedimientos elaborados para facilitar la obtención, gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados para resolver problemas complejos de planificación y gestión.

El término Sistema de Información Geográfica (SIG) según el Observatorio Regional de la Sociedad de la Información (ORSI) [map, 2009] define un SIG como un conjunto de hardware, software y datos que permite manejar, capturar, gestionar, implementar, visualizar y analizar información relacionada con la geografía (ver figura 2.2).

<b>S</b>	Sistema	Conjunto de hardware, software y datos
<b>I</b>	Información	Captura, gestión, interpretación, visualización y análisis
<b>G</b>	Geográfica	Información geográfica y topológica

Figura 2.2: Definición de Sistema de Información Geográfica (SIG)

<sup>1</sup>por su acrónimo en inglés: *Geographic Information System*

### 2.2.1 Estructura de la información en un SIG

La base de un SIG es la información que administra, la cual se encuentra almacenada necesariamente en una o varias bases de datos y se compone de una serie de caracteres alfanuméricos que representan la información geográfica.

Para definir el término de **información geográfica** se debe tomar en cuenta la definición de Geografía que es la ciencia que estudia la Tierra como astro, i.e., los fenómenos físicos, biológicos y humanos que ocurren en la superficie terrestre, así como sus causas y consecuencias. Por lo tanto, la información geográfica no se limita a un punto sobre la superficie terrestre más bien, dicha información puede ser climatológica, demográfica, topológica, medioambiental, económica y orográfica, asociada a fenómenos naturales, sociales, orográfica, etc. Entonces un SIG es capaz de administrar y representar todo tipo de información geográfica.

Toda esta información debe contener una referencia geográfica explícita, i.e., una coordenada, ya sea en un sistema universal (e.g., latitud y longitud) o bien en un sistema implícito (e.g., código postal o nombre de una calle). Esta referencia geográfica es necesaria para poder ubicar esta información en un punto del mapa.

Los SIG funcionan con base en dos modelos fundamentales para representar la información geográfica:

- el modelo *raster* y
- el modelo vectorial

El modelo *raster* consiste en representar la información dividiendo un mapa en celdas con un valor determinado, que puede ser un número, una imagen, un color, etc. (ver figura 2.3).

En el modelo vectorial, toda la información se codifica y almacena en un sistema de coordenadas (X,Y,Z) manteniendo las propiedades geométricas de las figuras. En este caso, sólo se podrán representar puntos, líneas y polígonos. La coordenada Z corresponde a la tercera dimensión y representa la altura mediante curvas de nivel sobrepuestas en el plano X-Y.

En sistemas comerciales, el modelo vectorial es el más utilizado debido a que es capaz de representar información con más precisión. El modelo *raster* sólo representa polígonos equivalentes a cada una de las celdas en las que se divide, aunque es un modelo muy útil para representaciones que no requieran de una gran precisión.

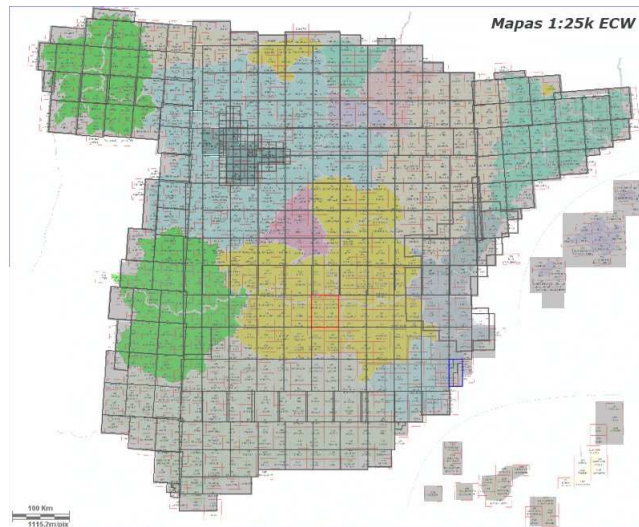


Figura 2.3: Ejemplo de un mapa representado con el modelo *raster*

En un sistema digital que abarca la cartografía de una zona determinada, se superponen diferentes planos o capas de información temática. Una capa es un nivel temático de datos geográficos digitalizados, i.e., contiene información geográfica sobre una temática y un nivel de información determinado.

Los SIG utilizan el análisis por superposición de capas. Este proceso de superposición permite visualizar y analizar varias capas simultáneamente sobre un mismo mapa. Es posible superponer en un mapa todas las capas que sean necesarias, lo que infiere una gran versatilidad y flexibilidad a los SIG. Las capas pueden estar representadas en un modelo *raster* o un modelo vectorial, e incluso en ambos modelos sin presentar ningún problema en la superposición de capas (ver figura 2.4).

Las aplicaciones actuales basadas en SIG permiten al usuario realizar un filtrado de información por capas. De esta forma es posible filtrar la visualización de mapas en función de la información que se quiera ver.

### 2.2.2 Estandarización de los SIG

En 1994 nace el *Open Geographic Consortium* (OGC) cuyo objetivo es el de definir estándares abiertos e interoperables para los SIG. Estos estándares buscan homogeneizar toda la información geográfica. Entre los estándares más importantes que ha definido el OGC se destacan tres:

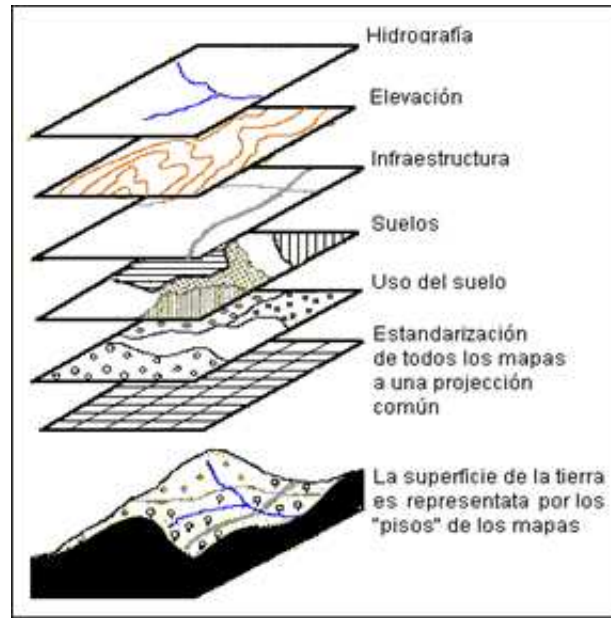


Figura 2.4: Ejemplo de un mapa con capas superpuestas

- GML (*Geographic Markup Language*)
- WMS (*Web Map Server*)
- WFS (*Web Feature Services*)

El estándar GML es en realidad un sublenguaje del lenguaje XML para modelar, almacenar e intercambiar información geográfica, i.e., GML define datos geográficos y su modo de almacenamiento en bases de datos para que, posteriormente, cualquier aplicación software sea capaz de trabajar con esa información.

El estándar WMS es un servicio de producción de mapas a través de la Web. Define un mapa como una representación de datos cartográficos para que puedan ser visualizados mediante de un navegador Web. Los mapas ofrecidos por WMS en realidad son imágenes con formato PING, GIF o JPEG. Actualmente, también se ofrecen en formatos vectoriales como SVG.

El estándar WFS es un interfaz de comunicación que permite interactuar con mapas WMS. Se trata de un servicio Web utilizado para la realización de consultas y recuperaciones de datos vectoriales con su información alfanumérica correspondiente.



## 2.3 Camino más corto: algoritmo de Dijkstra

Existen otros algoritmos que resuelven el problema de la ruta más corta, en este trabajo de tesis se eligió el algoritmo de Dijkstra debido a que el tiempo de ejecución es menor que del algoritmo de Bellman-Ford [Cormen et al., 2002], además en el artículo [Cherkassky et al., 1993] se realiza un estudio de los algoritmos del camino más cercano en el que se identifican las ventajas y desventajas de los siguientes algoritmos:

1. Bellman-Ford
2. Dijkstra
3. Grafo incremental
4. Umbral
5. Ordenamiento topológico
6. Cuadrícula
7. Cuadrícula amplia

En el artículo de Cherkassky et al. [Cherkassky et al., 1993] el al. se concluye que el algoritmo de Dijkstra es robusto y apropiado para problemas que contengan pesos no negativos.

Considere un motociclista que desea encontrar la ruta más corta de Chicago a Boston. Dado un mapa de carreteras de los Estados Unidos en el cual se marca la distancia entre cada par de intersecciones adyacentes ¿Cómo puede determinar la ruta más corta? Una solución es enumerar todas las rutas posibles de Chicago a Boston, añadir su distancia a cada ruta y seleccionar la más corta. Esta solución parece sencilla sin embargo, aún sin tomar en cuenta las rutas que contienen ciclos, existen millones de posibilidades de las cuales la mayoría no vale la pena considerar.

Para resolver el problema del camino más corto, se parte de un grafo ponderado  $G = (V, E)$ , con una función de peso  $w: E \rightarrow R$  que asigna a las aristas valores reales de ponderación [Diestel, 2000]. El peso de un camino  $p = (v_0, v_1, \dots, v_k)$  es la suma de los pesos de las aristas que lo constituyen:

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

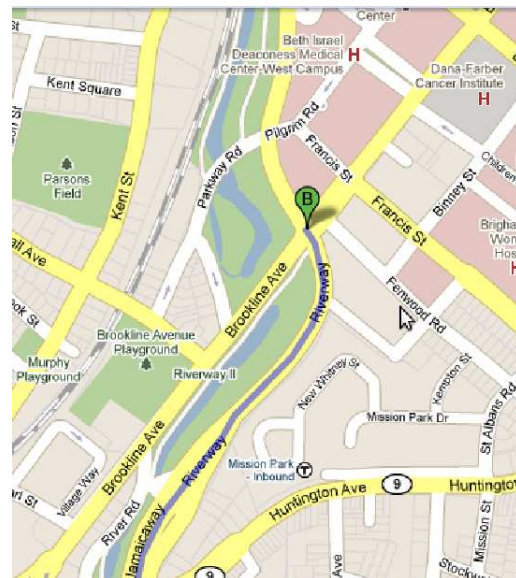
El peso del camino más corto de  $u$  a  $v$  se define por

$$\delta(u, v) = \begin{cases} \min\{w(p) : u \rightsquigarrow^p v\} & \text{Si hay un camino de } u \text{ a } v \\ \infty & \text{En cualquier otro caso} \end{cases}$$

En el ejemplo de la ruta más corta de Chicago a Boston se puede modelar el mapa de carreteras como un grafo, cuyos vértices representan las intersecciones y cuyos ejes denotan los segmentos de los caminos entre las intersecciones. El peso de una arista representa la distancia del camino. El objetivo es encontrar el camino más corto de una intersección en Chicago (e.g. *Clark St.* y *Addison Ave.*) a una intersección en Boston (e.g. *Brookline Ave.* y *Riverway*).



(a) Punto A. Intersección en Chicago (*Clark St.* y *Addison Ave.*)



(b) Punto B. Intersección en Boston (*Brookline Ave.* y *Riverway*)

Figura 2.5: Intersecciones para encontrar el camino más corto

Los pesos de las aristas pueden ser utilizados como indicadores además de las distancias. Los pesos son usados para representar tiempo, costo, penalizaciones, pérdidas o cualquier otra cantidad acumulable linealmente a lo largo del camino que se quiere minimizar.

Entonces el problema de la ruta más corta entre dos puntos se define como: encontrar el camino más corto de  $u$  a  $v$  dado un vértice origen  $u$  y un vértice destino  $v$  [Cormen et al., 2002]. Los algoritmos de ruta más corta normalmente se basan en la siguiente propiedad: el camino más corto entre dos vértices contiene otras rutas más cortas.



Figura 2.6: Ruta más corta del punto A al punto B

El camino más corto no puede contener un ciclo positivo de peso, ya que la eliminación del ciclo de la ruta produce una ruta con los mismos vértices origen y destino, pero con una trayectoria que tiene un peso más bajo.

El algoritmo de Dijkstra resuelve el problema del camino más corto en un grafo dirigido con pesos en sus aristas  $G = (V, E)$  para el caso en el que todos los pesos sean no negativos.

### 2.3.1 Representación del camino más corto

Dado un grafo  $G = (V, E)$  se mantiene por cada vértice  $v \in V$  un predecesor denominado  $\pi[v]$  que es un vértice diferente o un valor nulo. Se denomina  $d[v]$ , a una estimación del camino más corto.

La inicialización del grafo se realiza con el siguiente procedimiento: por cada vértice  $v \in V$  se almacena en la variable  $d[v]$  un valor infinito y en la variable  $\pi[v]$  un valor nulo (ver Pseudocódigo 1)

#### Pseudocódigo 1: Inicialización del grafo

```

INICIALIZA-ORIGEN ( $G, s$ )
1  for each vertex  $v \in V[G]$ 
2      do  $d[v] \leftarrow \infty$ 
3          $\pi[v] \leftarrow \text{NIL}$ 
4   $d[s] \leftarrow 0$ 
  
```

El proceso de relajación de una arista  $(u, v)$  consiste en comprobar si es posible mejorar el camino más corto encontrado hasta ahora. Dicho camino finaliza en el vértice  $v$  y pasa a través de  $u$ , si el camino se puede mejorar, se actualizan los valores contenidos en  $d[v]$  y  $\pi[v]$ . El código para realizar la relajación de la arista  $(u, v)$  se muestra en el pseudocódigo 2.

### Pseudocódigo 2: Relajación de una arista

```

RELAJA ( $u, v, w$ )
1   if  $d[v] > d[u] + w(u, v)$ 
2       then  $d[v] \leftarrow d[u] + w(u, v)$ 
3            $\pi[v] \leftarrow u$ 

```

El algoritmo de Dijkstra usa la técnica de **relajación**, por cada vértice  $u \in V - S$  se mantiene un atributo  $d[v]$  que es el límite superior del peso del camino más corto desde el origen  $s$  a el vértice  $v$ .

### 2.3.2 Algoritmo de Dijkstra

El algoritmo de Dijkstra mantiene un conjunto  $S$  de vértices cuyos pesos finales del camino más corto han sido determinados con base en el vértice origen  $s$ . El algoritmo selecciona repetidamente el vertice  $u \in V - S$  con el estimado mínimo de la ruta más corta, añade  $u$  a  $S$  y utiliza la técnica de relajación para todas las aristas sin modificar el valor de  $u$ . En el pseudocódigo 3 se usa una cola de prioridad mínima  $Q$  de vértices definidos en  $d[]$  donde  $d[]$ , como se mencionó anteriormente, contiene una estimación del camino más corto.

### Pseudocódigo 3: Algoritmo de Dijkstra

```

DIJKSTRA ( $G, w, s$ )
1 INICIALIZA-ORIGEN ( $G, s$ )
2  $S \leftarrow \emptyset$ 
3  $Q \leftarrow V[G]$ 
4   while  $Q \neq \emptyset$ 
5       do  $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
6            $S \leftarrow S \cup u$ 
7           for each vertex  $v \in \text{Adj}[u]$ 
8               do RELAJA( $u, v, w$ )

```

La línea 1 corresponde al proceso de inicialización del grafo mencionada en el Pseudocódigo 1, en tanto que la línea 2 inicializa el conjunto  $S$  con el conjunto vacío. El algoritmo

mantiene la invariante de que  $Q = V - S$  al inicio de cada iteración del ciclo **while** de las líneas 4 - 8. La línea 3 inicializa la cola de prioridad mínima  $Q$  para que contenga todos los vértices de  $V$ ; como  $S = \emptyset$  en ese momento, la invariante es verdadera después de la línea 3. En cada ciclo **while** de las líneas 4-8, un vértice  $u$  es extraído de  $Q$  y añadido al conjunto  $S$  de tal modo que se mantiene la invariante. Por lo tanto, el vértice  $u$  maneja el estimado menor de la ruta más corta de cualquier vértice en  $Q$ . Entonces las líneas 7 - 8 relajan cada eje  $(u, v)$  dejando a  $u$ . Así se actualiza el estimado  $d[v]$  y el predecesor  $\pi[v]$  si el camino más cercano a  $v$  puede ser mejorado a través de  $u$ .

Como el algoritmo de Dijkstra siempre elige el vértice más cercano (o el más ligero) en  $V - S$  para añadirlo al conjunto  $S$ , se puede decir que usa una estrategia codiciosa.

El algoritmo de Dijkstra relaja los ejes como se muestra en la figura 2.7. El origen  $s$  es el vértice que se encuentra más a la izquierda. Las estimaciones del camino más corto se muestran dentro de los vértices en tanto que los bordes sombreados indican valores predecesor.

Los vértices negros representan el conjunto  $S$  y los vértices blancos son la cola de prioridad mínima  $Q = V - S$ . El inciso (a) de la figura muestra la situación justo antes de la primera iteración del ciclo **while** de las líneas 4-8. El vértice sombreado tiene el valor mínimo  $d$  y es elegido como el vértice  $u$  en la línea 5 de la función Dijkstra. Los incisos (b) al (f) muestran la situación después de cada iteración del ciclo **while**. Los vértices sombreados en cada parte son elegidos como el vértice  $u$  de la línea 5 de la siguiente iteración dentro de la función Dijkstra. Los valores  $d$  y  $\pi$  se muestran en el inciso (f) como los valores finales.

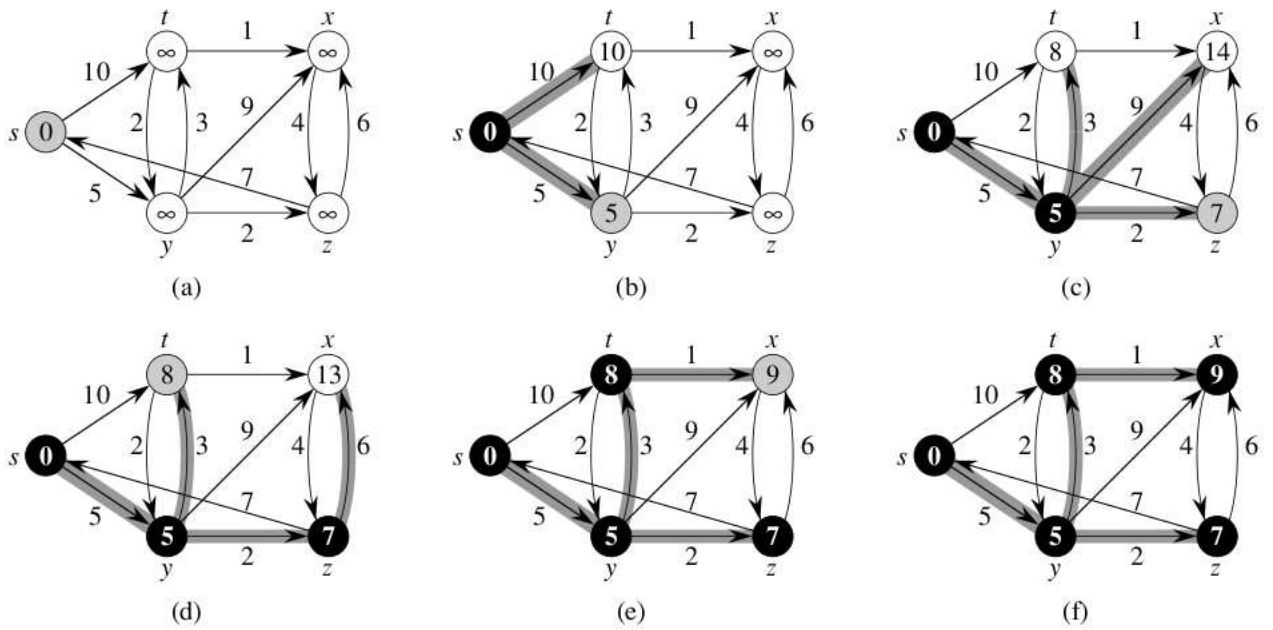


Figura 2.7: Ejecución del algoritmo de Dijkstra

# Capítulo 3

## Trabajos relacionados

En este capítulo se analizan los proyectos relacionados con el sistema propuesto en la presente tesis. En la sección 3.1, se estudia los principios de los sistemas basados en localización. Particularmente, se describe la detección de proximidad y separación, así como su aplicación en algunos sistemas relevantes. También se aborda el principio de ubicación geográfica, el cual se refiere a la determinación de la posición de un usuario en un sistema basado en localización. En la sección 3.2 se estudia la forma en la que un sistema descubre la ubicación de un dispositivo y se describe algunas técnicas de ubicación. En particular, se describen las técnicas que utilizan las redes telefónicas celulares, redes satelitales y estaciones de corto alcance. En la sección 3.3, se pone en evidencia las características, ventajas y desventajas de los sistemas basados en localización que están relacionados con el sistema propuesto en esta tesis. Por último, en la sección 3.4 se expone una tabla comparativa con las propiedades generales que de dichos sistemas.

### 3.1 Servicios basados en localización

Para proveer al usuario servicios de información conscientes de su ubicación, un sistema debe “sentir” la ubicación actual del usuario. Esta ubicación debe ser precisa para que la información manejada por el sistema sea lo más real posible. De otra manera, el usuario puede ser ubicado incorrectamente en el entorno, e.g. si existe un error de un metro, el usuario puede ser ubicado en otro cuarto o pasillo.

Los servicios basados en localización (*Location-Based Services*) ofrecen servicios personalizados a usuarios de dispositivos móviles, tomando en cuenta su ubicación. En otras palabras, ayudan a los usuarios a determinar su posición, localizar un objetivo (servicios, productos, recursos y miembros de una organización) y recibir una guía de cómo llegar a él.

Los servicios basados en localización responden a las siguientes preguntas: ¿Dónde estoy? ¿Qué está alrededor de mí? y ¿Cómo llego ahí? Proveen servicios y aplicaciones



personalizadas usando la ubicación del usuario en conjunto con información adicional. Para determinar la ubicación del usuario los servicios basados en localización pueden utilizar varias técnicas.

Por ejemplo, considérese un servicio inalámbrico de emergencia, el cual determina automáticamente la ubicación de la persona que llama. Este servicio sería extremadamente útil ya que el usuario evita problemas de marcación (números erróneos). Algunos ejemplos clásicos de servicios basados en localización son: los avisos de tráfico, la asistencia en carreteras y la ayuda de navegación que incluye mapas y direcciones.

Otros servicios basados en localización pueden combinar la ubicación actual del usuario con información referente a sus preferencias para ayudarlo a encontrar restaurantes, hospedaje o entretenimiento, adaptándose a sus gustos y a su presupuesto.

Según Snsekkens [Snekkens, 2001] existen dos tipos de tecnologías usadas en la localización de usuarios: 1) el rastreo y 2) el posicionamiento. Cuando se usa la tecnología de rastreo, la ubicación del usuario es obtenida por alguna entidad externa (e.g., el servidor o el operador de la red). En contraste, cuando se utiliza la tecnología de posicionamiento, el usuario con ayuda de algún dispositivo genera su propia ubicación (e.g., el cliente o GPS).

Los servicios basados en localización toman en cuenta la posición actual del usuario, con el fin de entregar una lista de los puntos de interés más cercanos, tales como salones o recursos.

### Detección de proximidad entre dispositivos.

La detección de proximidad es definida por Kjaerdaard et al. [Kjaerdaard et al., 2008] como “la capacidad de un servicio basado en localización para detectar cuando dos dispositivos móviles han excedido una distancia de cercanía preestablecida”. Análogamente, la detección de separación descubre cuando dos dispositivos se han alejado más allá de una distancia de separación previamente definida.

En otros trabajos, los mecanismos de proximidad y detección de separación están basados en el *middleware* TRAX [Azizyan et al., 2009]. Estos mecanismos controlan el posicionamiento de los dispositivos móviles sin necesidad de utilizar GPS y coordinan la transferencia de dicho posicionamiento a un servidor central de localización para comprobar la proximidad o separación con respecto a otros dispositivos.

La proximidad y la separación pueden estar basadas en la línea de observación o en la distancia Euclidiana, la cual puede ser calculada mediante la posición geográfica del dispositivo involucrado. La comprobación de la proximidad o de la separación por medio de distancias Euclidianas no tiene mucho sentido en interiores, debido a que los disposi-



tivos pueden estar localizados unos encima de otros, como en el caso de encontrarse en diferentes pisos de un edificio.

## 3.2 Ubicación geográfica de un dispositivo móvil

La localización precisa de un dispositivo móvil es provista por medio de geocodificación, que es el proceso de asignar coordenadas geográficas (e.g., latitud y longitud) a puntos de un mapa (e.g., salones y edificios). Las coordenadas geográficas pueden ser utilizadas para localizar el punto correspondiente del mapa en un Sistema de Información Geográfica. Para procesar la ubicación de un dispositivo móvil, un servicio basado en localización rastrea y mide las radioseñales emitidas por dicho dispositivo (e.g., la ubicación de la celda colindante o del satélite en órbita).

Las bases de datos espaciales almacenan representaciones de objetos como pueden ser países, ciudades, caminos, así como información espacial que pueda ser representada en un mapa. Una representación está asociada a un método de indexación espacial que consiste en usar "llaves espaciales" para buscar la información relacionada con un objeto, permitiendo la correlación entre un objeto y su posición actual.

Mientras que un usuario móvil no sea parte de los datos geográficos espaciales, cada registro del usuario puede incluir una llave espacial, la cual indica su posición actual. Además, las bases de datos espaciales puede asociar cada llave de objeto espacial con datos no geográficos, i.e., se puede identificar a los dispositivos móviles que están dentro de los límites del mapa espacial [Pashtan, 2005].

Para descubrir la ubicación de un dispositivo, los servicios basados en localización usan métodos de posicionamiento en tiempo real. La precisión de esta ubicación depende del método utilizado. La ubicación del dispositivo puede expresarse en términos espaciales o dr descripciones textuales. Una ubicación espacial puede ser representada en el sistema de coordenadas latitud-longitud-altitud.

La latitud es la distancia angular entre el Ecuador y un punto del planeta. Se mide de  $0^\circ$  a  $90^\circ$ . Para representar puntos al norte del Ecuador se usan valores positivos, en tanto que para representar puntos al sur se emplean valores negativos.

La longitud expresa la distancia angular entre algún punto de la superficie terrestre y el meridiano  $0^\circ$ , también denominado meridiano de Greenwich. Se mide de  $0^\circ$  a  $180^\circ$ . Para representar puntos al este se usan valores positivos y para los puntos al sur se emplean valores negativos. La altitud es la distancia vertical de un punto de la tierra respecto al nivel del mar.

Existe otro tipo de representación geográfica denominado descripciones textuales, las cuales generalmente se expresan como direcciones de calles, que incluyen ciudad, código postal, etc.

Un gran número de investigaciones sobre servicios basados en localización y sistemas de localización comerciales ha sido desarrollado en las últimas dos décadas [Hazas et al., 2004]. En general, estos sistemas cumplen alguno de los siguientes objetivos: 1) proveer un alto grado de precisión al momento de estimar la localización dentro de un área pequeña (interiores) y 2) proveer una precisión menor en una zona de cobertura amplia (exteriores). Los sistemas que se enfocan en la precisión (interiores) generalmente requieren una extensa infraestructura de sensores relativamente caros.

Los métodos más populares para implementar aplicaciones que usan servicios basados en localización son: la red telefónica celular, los satélites y las estaciones de corto alcance posicionadas en la infraestructura de un edificio. En las siguientes sub-secciones se describen algunos trabajos y tecnologías que usan estos métodos.

### 3.2.1 Red telefónica celular

Los sistemas de localización para exteriores más populares son GPS y GSM. La tecnología GSM (*“Groupe Special Mobile”*) es provista por las empresas operadoras de telefonía móvil y permite determinar, con poca precisión, dónde se encuentra físicamente un dispositivo móvil. La técnica más utilizada es la “celda ID”, según la cual se realiza una petición, a las torres celulares de las compañías telefónicas, para saber en qué celda se encuentra la persona o el dispositivo móvil que se desea localizar. El teléfono celular es la plataforma de cómputo más ubicua y un excelente candidato para los servicios basados en localización.

Los problemas del método de la localización basada en teléfonos celulares son la precisión, la privacidad y el costo [Borriello et al., 2005]. Primero, la precisión que obtiene este método es de 100 m, la cual puede ser ineficiente para una aplicación basada en localización de cualquier tipo. Segundo, los datos son transmitidos y procesados por la infraestructura de las compañías proveedoras y retransmitidos al teléfono del usuario, lo cual implica que el usuario debe “confiar” en que el proveedor de servicios proteja la información de su ubicación. Por último, los proveedores no proporcionan la información de la ubicación del usuario gratis, hay una tarifa de un dolar por cada petición de localización.

### 3.2.2 Satélites

Otro tipo de sistema de localización para exteriores es GPS (*“Global Positioning System”*) el cual representa el método de ubicación física más popular para dispositivos móviles. GPS es controlado por el Departamento de Defensa de Estados Unidos y usa 24 satélites que orbitan la tierra. GPS determina la posición de un dispositivo usando el principio matemático de triangulación de señales satelitales. La señal de GPS es codificada, por tal motivo los dispositivos móviles deben ser equipados con un receptor de GPS.

Este método es probablemente el más exacto (entre 4 y 40 m si el receptor GPS tiene una clara vista al cielo) pero tiene algunas desventajas: el hardware extra puede ser costoso, consume energía mientras está en uso y requiere mucho tiempo para fijar los satélites visibles. En ciudades donde la señal es intermitente, se genera el “efecto cañón”, i.e., la señal no puede verse correctamente en los dispositivos, quedando en blanco o con información des-sincronizada. Otra característica no deseada de GPS es que requiere una comunicación directa y alineada con varios satélites. Por lo tanto, no funciona correctamente dentro de edificios, en áreas con mucha vegetación o en lugares con muchos edificios altos.

### 3.2.3 Estaciones de corto alcance

En los últimos años se ha desarrollado diferentes esquemas de posicionamiento en interiores, los cuales se pueden clasificar en: 1) el tipo de señales utilizadas (e.g., infrarrojos, Radiofrecuencia y ultrasónicas), 2) el tipo de medición de dichas señales (e.g., el tiempo de llegada y la intensidad de la señal recibida) y 3) los métodos matemáticos (e.g., huella dactilar, ángulo de llegada y triangulación). Estos esquemas son usados para obtener como resultado la determinación de la posición de un objeto con base en las mediciones realizadas.

El esquema de posicionamiento denominado "ángulo de llegada" (*Angle of Arrival, AoA* o *Direction of Arrival, DoA*) utiliza varias antenas o puntos de acceso alineados para determinar el ángulo de la señal incidente. La posición del usuario se calcula mediante la triangulación de los ángulos obtenidos. La efectividad de este esquema radica en el número de puntos de acceso ya que si se implementa con uno o dos no será muy preciso, i.e., esta técnica es efectiva cuando existe un arreglo de antenas o puntos de acceso.

El esquema de "tiempo de llegada" (*Time of Arrival o ToA*) se basa en el hecho de que una señal electromagnética tarda un tiempo en llegar del emisor al receptor. Hay una relación lineal entre el tiempo de propagación y la distancia recorrida. Mediante técnicas de triangulación se puede calcular la distancia entre el emisor y el receptor.

El esquema de la intensidad de la señal recibida (*received signal strength* o *RSS*) usa la intensidad de la señal recibida para determinar la localización del emisor. Con base en la información obtenida se puede estimar la distancia que existe entre el emisor y el receptor. Esta técnica no es muy precisa cuando existen multitrayectos, reflexiones o difracciones en la señal recibida.

Uno de los esquemas más populares es denominado “ubicación de huellas” (LF, “*location fingerprinting*”), el cual estima la posición de un objeto a partir de las medidas de la intensidad de las señales recibidas por un emisor. La localización de los emisores es determinada asociando los valores medidos por patrones RSS, los cuales son llamados huellas. Cada huella tiene una posición pre-registrada y bien definida que está almacenada en una base de datos [Kjaergaard et al., 2008].

Hasta ahora se han examinado varios esquemas de ubicación física pero, según Azizyan et al. [Azizyan et al., 2009], dichos esquemas pueden no ser adecuadas para identificar con precisión localidades lógicas (e.g., cuartos, salones y pasillos). Esta limitación se debe a que dos localidades lógicas pueden estar separadas por una pared, por lo tanto un margen de error de 5 m puede ubicar al usuario en el lado incorrecto del muro.

Para distinguir las localidades adyacentes, algunos proyectos han instalado transmisores especiales en cada cuarto, e.g., el sistema Cricket [Priyantha et al., 2001] ha logrado una escala de localización en centímetros combinando la radiofrecuencia, señales ultrasónicas y sus respectivos receptores instalados en los alrededores.

En la sección 3.6 se describen algunos sistemas de localización para interiores, los cuales confían en un método de localización específico (e.g., triangulación, trilateración y distancia euclidiana), utilizan sensores (e.g., infrarrojos, ópticos y medallas ultrasónicas) así como tecnologías de comunicación (e.g., Bluetooth IEEE 802.11 y redes ad-hoc) que proveen una precisión adecuada para trabajar en interiores.

### 3.3 Trabajos relacionados

Resumiendo, en los últimos años se ha desarrollado varios servicios basados en localización que pueden clasificarse de la siguiente manera: 1) detección de proximidad y separación, 2) localización en exteriores (se basa en cubrir un área muy amplia pero sacrificando exactitud de localización) y 2) localización en interiores (se basa en radio de corto alcance y sensores). En esta sección se describen los sistemas de servicios basados en localización para interiores más populares. Los dos primeros tipos de sistemas están

fuera del alcance de la presente tesis.

### 3.3.1 NearMe

NearMe [Krumm and Hinkley, 2004] está conformado por un servidor, algoritmos e interfaces de programación de aplicaciones para clientes que están equipados con redes IEEE 802.11 (Wi-Fi) que en conjunto son utilizados por NearMe para detectar a las personas y objetos que se encuentren físicamente cerca del usuario solicitante. Con esa información se genera una lista de clientes. NearMe compara la lista de clientes de un punto de acceso Wi-Fi con la intensidad de la señal recibida del objeto a detectar para calcular la proximidad de los dispositivos.

NearMe soporta detección de proximidad en distancias cortas y toma en cuenta las mediciones RSSI (intensidad de la señal recibida) y la distancia Euclidiana. NearMe sigue un modelo cliente-servidor que maneja actualizaciones periódicas de RSSI, lo cual provoca sobrecargas significativas cuando los objetivos no se mueven en un periodo de tiempo.

### 3.3.2 Sistema *Active Badge*

El sistema de localización *Active Badge* [Hopper et al., 1992] fue desarrollado en *Olivetti Research Laboratory*. Es probablemente el primer prototipo de un sistema de localización en interiores el cual consiste en un sistema de proximidad celular que usa la tecnología de sensores infrarrojos.

El sistema *Active Badge* provee la ubicación de individuos dentro de un edificio, determinando la localización de sus “medallas activas”. Los usuarios usan estos pequeños dispositivos (que tienen forma de medalla) para transmitir una señal infrarroja única cada 10 o 15 segundos. El dispositivo fue diseñado en un paquete de aproximadamente  $55 \times 55 \times 57$  mm y pesa 40 gr.

A su vez, cada oficina está equipada con uno o más sensores en red, que detectan las transmisiones de los dispositivos móviles. Una estación central, que también se encuentra conectada a la red, sondea los sensores en red buscando los dispositivos en forma de medallas, procesa la información recabada y la pone a disposición de los clientes de forma gráfica.

Las señales infrarrojas se usan para la comunicación entre el dispositivo móvil (medalla activa) y las estaciones receptoras (sensores en red). Debido a que los dispositivos son muy pequeños y su costo de fabricación es mínimo, se pueden colocar varios sensores en una habitación y son reflejadas por particiones, i.e., se pueden colocar varios sensores que no necesitan de una instalación precisa cuando se usan dentro de una habitación pequeña.

Estas señales pueden operar en un rango de 6 m

Una medalla activa consume energía. Tomando en cuenta que la unidad sólo envía un pulso cada 15 segundos, consume un bajo nivel de energía dando como consecuencia que la batería pueda durar hasta un año. Además, las medallas poseen un sensor de luz que, cuando detecta poca o nada de luz, deja de emitir pulsos.

Un requerimiento indispensable de los servicios basados en localización es que se pueda detectar un número grande de personas en la misma localidad. En el caso del sistema *Active Badge* este requerimiento no se cumple debido a que la duración de la señal, emitida por la medalla, es de una décima de segundo, entonces hay una probabilidad de  $2/150$  de que dos señales colisionen cuando dos medallas están situadas en la misma posición, i.e., para un grupo grande de personas, hay una pequeña probabilidad de que todas sean detectadas.

Otra desventaja es que el sistema no puede detectar una señal poco frecuente. Si el usuario o la medalla activa se traslada muy rápido, los sensores capturan señales poco frecuentes ya que censan con un margen de 15 segundos, esto ocasiona que un usuario no sea detectado.

### 3.3.3 Sistema *The Bat Ultrasonic Location*

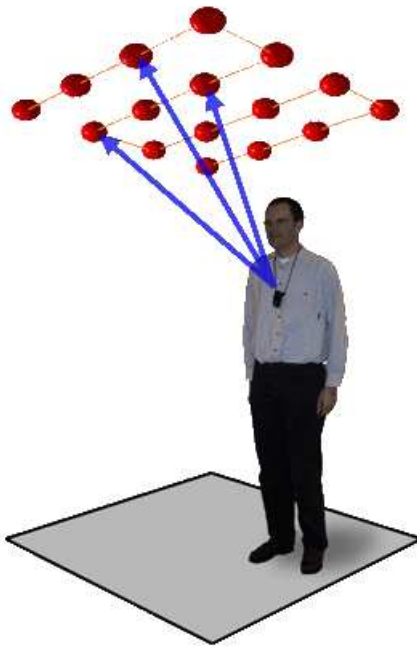
AT&T desarrolló el sistema de localización denominado *The Bat Ultrasonic Location System* [Harter et al., 1999], el cuál usa una técnica ultrasónica denominada “tiempo de recorrido” que provee una mejor precisión física que el Sistema *Active Badges*. El sistema *The Bat Ultrasonic Location* funciona de la siguiente manera: un pulso ultrasónico es emitido por un transmisor llamado “Bat”, el cuál está unido al objeto a localizar. A continuación se mide el “tiempo de vuelo” del pulso a los receptores que se encuentran montados en el techo, en puntos conocidos.

La velocidad con la que el sonido se dispersa en el aire es conocida, por lo tanto se puede calcular las distancias del transmisor Bat a cada receptor. Teniendo al menos tres distancias se tiene suficiente información para determinar la posición 3D del objeto. El sistema puede localizar objetos con una exactitud de 9 cm para el 95 % de las mediciones. Las desventajas de este sistema son la escalabilidad, la dificultad de implementación y el costo de instalación.

La figura 3.1a muestra un transmisor Bat el cual está unido a un usuario y emite un sonido que es recibido por varias estaciones que se encuentran montadas en el techo. A continuación se enlistan las características del dispositivo:

1. mide mide 7.5 cm × 3.5 cm × 1.5 cm (ver figura 3.1b)

2. es alimentado por una pila de Litio Cloruro de Tionilo de 3.6 V, la cual tiene un tiempo de vida de 15 meses
3. tiene un código único de 48 bits y está ligado con la infraestructura de sistema de localización fija
4. usa un enlace bidireccional de radio a 433MHz para comunicarse
5. tiene dos botones, dos LEDs y un altavoz, lo cual permite que sea usado como un dispositivo ubicuo de entrada y salida de datos
6. cuenta con un medidor de voltaje que permite consultar remotamente el estado de la batería.



(a) Funcionamiento de un Bat



(b) Transmisor Bat

Figura 3.1: Sistema Bat

### 3.3.4 Sistema Cricket

Cricket [Priyantha et al., 2001] combina radiofrecuencia (RF) y señales ultrasónicas para proveer información de la ubicación, de un usuario u objeto, a los dispositivos servidores

conectados.

Estos dispositivos se ubican montados en las paredes y en el techo de los edificios y transmiten una señal dirigida que se encarga de publicar información por un canal de RF. En cada aviso de RF los servidores transmiten concurrentemente un pulso ultrasónico.

Los receptores, que son dispositivos móviles, buscan las señales de RF y cuando reciben algunos bits escuchan el pulso ultrasónico. Cuando este pulso llega, el receptor obtiene la distancia estimada de la estación correspondiente, tomando en cuenta la diferencia de la velocidad de propagación que existe entre la señal ultrasónica (velocidad del sonido) y la señal de RF (señal de la luz). Las estaciones usan un algoritmo descentralizado para que las transmisiones sean aleatorias, consiguiendo que las colisiones e interferencia entre ellas sean minimizadas.

Cricket usa las estaciones para diseminar la información, acerca del espacio geográfico, a los receptores. Una estación es un dispositivo pequeño que manda mensajes dentro del espacio geográfico donde se encuentra colocado. Básicamente, las estaciones transmiten cadenas cortas como por ejemplo el nombre de alguna computadora. Las estaciones que maneja el sistema cricket no son caras por tal motivo pueden usarse varias estaciones, dando como resultado una mejor cobertura.

Para obtener información acerca de un espacio, cada usuario, estático o móvil, tiene un receptor unido a él. Un receptor es un pequeño dispositivo que recibe mensajes de las estaciones y los usa para detectar en qué espacio se encuentra. Los receptores están provistos de un API que permite que los usuarios aprendan donde están, así ellos pueden usar dicha información para interactuar con un servicio de descubrimiento de recursos. Los receptores tienen dos sensores un sensor de RF y un sensor ultrasónico

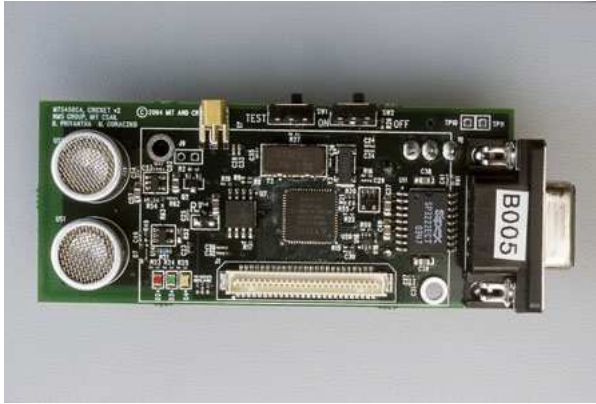
Para determinar la ubicación de un objeto se usa una combinación de RF y hardware ultrasónico, así un receptor puede calcular la distancia a la que se encuentra de la estación. Primero se mide el tiempo de propagación de una señal ultrasónica emitida por la estación, conociendo el hecho de que la velocidad del sonido en el aire (aproximadamente 3.444 m/s en condiciones normales) es mucho más pequeña que la velocidad de la luz (RF).

En cada transmisión una estación emite, concurrentemente, información acerca del espacio usando señales de RF y un pulso ultrasónico. Cuando el receptor escucha la señal de RF usa los primeros bits como información de aprendizaje y enciende el sensor ultrasónico. El receptor usa el tiempo de diferencia, entre la recepción del primer bit de información RF y la señal ultrasónica, para determinar la distancia a la estación.

La figura 3.2a muestra un dispositivo que funciona como estación en el sistema Cricket. Este dispositivo emite información de su entorno, e.g., emite el mensaje “Servidor”, así



los receptores pueden saber qué objetos se encuentran cerca. Estos dispositivos están colocados en la infraestructura del edificio cómo se ilustra en la figura 3.2b.



(a) Estación del sistema Cricket



(b) Infraestructura del sistema Cricket

Figura 3.2: Sistema Cricket

### 3.3.5 Sistema HiBall tracker

El sistema HiBall tracker [Welch and Bishop, 1997] incorpora seis sensores ópticos y seis lentes con filtros infrarrojos en un dispositivo del tamaño de una pelota de golf (ver figura 3.3a). Este dispositivo puede ser usado en la mano o en la cabeza del usuario denominado HiBall.

HiBall tracker obtiene la posición del usuario de la siguiente manera: el dispositivo usa los lentes, que están colocados “hacia arriba”, para buscar las señales luminosas que emiten los diodos (LEDs) instalados, con una gran precisión, en paneles montados en el techo del cuarto (ver figura 3.3b). La posición de la cabeza del usuario se obtiene mediante el uso de técnicas de fotogrametría. La orientación del usuario se obtiene basándose en el ángulo que forma la proyección de los LEDs en el lente del dispositivo.

Para determinar la posición del dispositivo HiBall se implementan reportes de rastreo. Cada vez que el dispositivo mide un nuevo nivel que esté por debajo de los reportados para formar una colección de observaciones. Este mecanismo resulta más conveniente que tener una sola observación ya que limita la solución matemática. La investigación está basada en un principio de “*single-constraint-at-a-time*” o rastreo SCAAT. El objetivo es que observaciones sencillas provean información acerca del estado del usuario que pueda ser usada para mejorar una estimación previa. Este principio se aplica recursivamente, la nueva información que es arrojada por los dispositivos tan pronto como es medida. Con esta estrategia se puede generar mejores estimaciones y más frecuentes con menor latencia

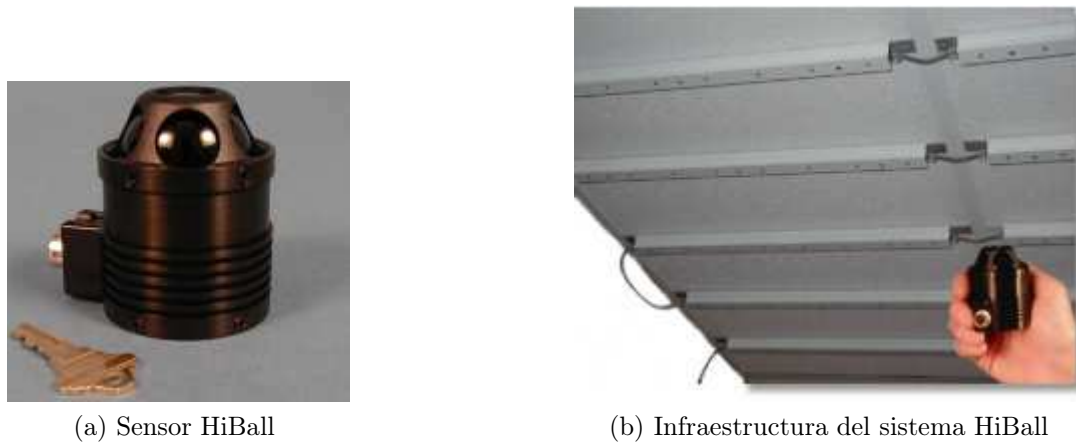


Figura 3.3: Sistema HiBall

y mayor exactitud.

### 3.3.6 Sistema RADAR

El sistema RADAR es una implementación que usa la tecnología de redes inalámbricas, fue desarrollada por Microsoft Research Group. RADAR mide la intensidad y el ruido de la señal enviada por un dispositivo inalámbrico y procesa su posición 2D dentro de un edificio. RADAR usa la información, de la intensidad de señal, que es recopilada por múltiples estaciones base para determinar las coordenadas de un usuario.

Las estaciones base están colocadas estratégicamente para proveer cobertura en un área de interés. La triangulación se realiza combinando las mediciones empíricas con el modelo de propagación de señales para determinar la localización del usuario. Cada estación está equipada con una interfaz de red basada en la tecnología WaveLAN RF LAN de la compañía Lucent. El alcance de la red es de 200, 50 y 25 m para ambientes abiertos, semi-abiertos y cerrados, respectivamente. Esta clasificación varía dependiendo de la densidad de los materiales de construcción y de la obstrucción que existe entre el emisor y el receptor.

Este sistema usa la información de la señal (intensidad y ruido) para construir y validar un modelo de propagación de señales, el cual es usado para inferir la ubicación del usuario en tiempo real. Para recabar esta información se utilizó la tecnología WaveLAN la cual, además de la información convencional, permite saber la intensidad de la señal (*signal strength*, SS) y la proporción señal-ruido (*signal-to-noise ratio*, SNR).

La SS es reportada en unidades de dBm y la SNR en dB. Esta información se obtiene mediante la siguiente analogía: una intensidad de señal de 1 watt es equivalente a 30 dBm, además si el ruido de la señal es 0.1 watt el SNR será de 10 dB. El software FreeBSD 3.0 WaveLan extrae el SS y el SNR cada vez que un paquete del tipo *broadcast* es recibido.

El proceso de captura de datos opera de la siguiente manera: primero los relojes de los dispositivos inalámbricos y de las estaciones base son sincronizados; a continuación el dispositivo inicia un *broadcast* de paquetes UDP en un rango de 4 paquetes por segundo.

Cada estación base (bs) almacena la medición de la intensidad de la señal (SS) junto con una estampa de tiempo (t), entonces se almacena una tupla (t, bs, SS). Para determinar la ubicación exacta de un dispositivo se

Mediante esta información se puede realizar una búsqueda para determinar la ubicación más exacta. Con la ayuda de un algoritmo de búsqueda lineal se determina la ubicación del dispositivo.

### 3.3.7 Sistema WHISPER

El sistema WHISPER [Vallidis, 2002] es un sistema acústico de rastreo de usuarios, tiene como objetivo localizar la cabeza y las manos de uno o más usuarios, usar la información de su ubicación y dibujar un avatar en un ambiente virtual.

El hardware de WISPER consiste en los siguientes elementos: una unidad de procesamiento, un altavoz y un micrófono. El procesador genera la señal, la convierte en analoga mediante un convertidor digital-analógico (DAC), amplifica el resultado y envía la señal al altavoz. A su vez, convierte la señal eléctrica en una señal acústica, la cual se propaga a través del aire hacia el micrófono. La salida del micrófono es enviada a través de un amplificador y un filtro a un convertidor analógico-digital (ADC) y regresa al elemento de procesamiento.

El sistema WISPER utiliza una señal de banda ancha para aprovechar la capacidad de la difracción del sonido, fenómeno que consiste en el curvado y esparcido de las ondas (en este sistema las ondas de sonido son de baja frecuencia) cuando encuentran un obstáculo.

El enfoque principal del sistema es realizar un seguimiento, de forma recursiva, de la correlación que existe entre las versiones transmitidas y recibidas de una señal acústica de banda ancha pseudo-aleatoria. La diferencia del valor de correlación máximo corresponde al retraso, el cual a su vez corresponde a la distancia que existe entre el micrófono y el altavoz.

La correlación es costosa, en términos computacionales, pero el sistema WHISPER re-

duce el cómputo necesario al restringir el retraso que se genera en el espacio de búsqueda y al usar un filtro Kalman para predecir el retraso de la secuencia actual de pseudo-ruido.

### 3.4 Comparación de trabajos relacionados

A continuación se expone una comparación, de trabajos mencionados anteriormente, con las propiedades generales que deben cumplir los sistemas que usan servicios basados en localización.

Generalmente los sistemas de localización comparan localidades absolutas, las cuales requieren una adecuada calibración y configuración, por el contrario NearMe trabaja con información de proximidad de usuarios, por lo tanto no necesita calibración geométrica ni tampoco una configuración mínima. NearMe tiene un rango limitado de 30-100 metros, el cual depende del alcance de los puntos de acceso y de la forma en la que están conectados. Los puntos de acceso se encuentran inter-conectados de tal manera que las señales se superponen, formando una topología compleja y costosa. Además, todos los clientes del sistema NearMe, ya sean usuarios o recursos estáticos (como impresoras o salones para conferencias) deben de registrarse en el sistema para que todos ellos conformen la lista que contiene la información de proximidad. Este requerimiento provoca que el sistema tenga un número limitado de clientes, i.e., si existe una alta concentración de clientes en un punto, la respuesta del sistema será lenta debido a la alta saturación de información y la carga de trabajo.

El dispositivo (o medalla) que usa el sistema Active Badge para localizar objetos es muy pequeño (mide  $55 \times 55 \times 57$  mm y pesa 40 gr) lo cual hace que sea portable y económico. Puede detectar tanto a usuarios como a recursos estáticos. Sin embargo, en grupos grandes de personas se corre el riesgo de que las señales colisionen, i.e., el sistema Active Badge sólo garantiza la detección de personas en grupos pequeños, ya que la duración de la señal emitida es de 15 segundos. Además, las señales infrarrojas que emite el sistema Active Badge para localizar a los dispositivos no pueden “atravesar” objetos. Si existe un obstáculo entre el emisor y el receptor infrarrojo, la señal no podrá ser detectada. Además la señal tiene un rango de 6 m lo cual dificulta la localización de un usuario que se encuentra alejado del receptor.

Los dispositivos conocidos como Bat, en el sistema Bat Ultrasonic, funcionan como emisores de señales ultrasónicas y, al ser inalámbricos, necesitan usar una fuente de energía, para enviar y recibir información. Este dispositivo cuenta con una batería de litio de alta duración, la cual tiene un tiempo de vida de 18 meses. Además, este sistema puede determinar la posición (en tres dimensiones) de 75 objetos cada segundo con una exactitud

Tabla 3.1: Tabla comparativa de trabajos relacionados

Parámetro \ Sistema	<i>Active Badge</i>	<i>HiBall tracker</i>	<i>Bat Ultrasonic</i>	<i>Cricket</i>	<i>RADAR</i>	<i>WHISPER</i>	<i>NearMe</i>
Método de localización	Proximidad celular	SCAAT	Trilateración	Correlación	Triangulación	"tiempo de llegada"	Distancia euclidiana
Tecnología de comunicación	Sensores infrarrojos	Sensores ópticos	Transmisores infrarrojos	RF y ultrasónica	SS y SNR	Espectro de audio	RRSI
Calibración geométrica	Si	Si	Si	Si	Si	No	No
Rango de alcance	6 m.	Un cuarto de 4.5 x 8.5 cm	No específica	No específica	200, 50 y 25 m.	No específica	30-100 m
Detección de recursos estáticos	Si	Si	Si	Si	Si	Si	Si
Duración de batería	No específica	No específica	15 meses	No específica	No usa	No usa	No usa
Costo de instalación	Económico	No específica	Caro	Económico	No específica	No específica	No específica
Dificultad de Escalabilidad	Media	Compleja	Compleja	Media	Media	Media	Media
Exactitud de detección	No específica	0.5 mm	3 cm	No específica	No específica	No específica	No específica
Capacidad de atravesar obstáculos	No	No	Si	Si	Si	Si	Si
Implementación en varios pisos	No	No	No	No	No	No	No

de 3 cm y una precisión de 95 %. Sin embargo, la implementación de una red de sensores ultrasónicos, en el sistema Bat Ultrasonic, resulta muy costosa debido a que se debe adquirir un número considerable de sensores para crear una matriz, que será instalada en la infraestructura de los cuartos a monitorear. Además cada individuo u objeto que quiera

interactuar con el sistema debe contar con un sensor que actúa como emisor de señales ultrasónicas. El proceso de añadir más cuartos o edificios, al sistema Bat Ultrasonic, resulta un proceso complicado, debido a que se debe implementar e instalar una topología de red de sensores muy extensa y complicada. Por cada cuarto se debe instalar una matriz de sensores, que debe tener una calibración geométrica adecuada para lograr una buena exactitud.

El sistema HiBall tracker genera 2000 estimaciones de localización por segundo, con menos de un milisegundo de latencia. El ruido de orientación es menor a  $0.5\text{ mm}$  y  $0.02^\circ$  en un cuarto de  $4.5$  por  $8.5$  m. Sin embargo, las señales luminosas que emite el dispositivo HiBall para ser localizado por las estaciones no puede atravesar objetos, provocando que, si existe un obstáculo entre el dispositivo y las estaciones, estas no podrán detectar la señal, por lo tanto, no se podrá determinar la ubicación del dispositivo. Además, en cada cuarto en el que se instale el sistema HiBall tracker se debe colocar una red de sensores LEDs, que necesitan una calibración geométrica precisa para garantizar la exactitud de la medición del posicionamiento del usuario. Este proceso dificulta la escalabilidad del sistema.

El sistema RADAR requiere de algunas estaciones base y de una red inalámbrica dentro del edificio, lo cual facilita la instalación, y la escalabilidad del sistema. Además la señal inalámbrica puede atravesar muros y objetos, esto da como resultado una comunicación constante entre el emisor y el receptor dentro de un rango de  $25$  m. La desventaja del sistema RADAR involucra la dificultad de aplicar este sistema en un edificio de muchos pisos. La señal recibida es representada en el sistema como una línea o un valor que puede ir en cualquier dirección. Este enfoque no respeta ni paredes ni pisos, por lo que este sistema no puede distinguir en qué piso o en qué salón se encuentra un dispositivo debido a que lo único que maneja es la distancia euclidiana representada mediante la intensidad de la señal que recibe.

El sistema WHISPER puede identificar a un usuario incluso si existen obstáculos entre el emisor y el receptor. Puede calcular 1000 ubicaciones en 3D para dos objetivos en un segundo. Tiene una latencia de  $18$ - $49$  milisegundos, esto depende del ruido de la señal y de la distancia al objetivo. Hay algunos factores que alteran el rendimiento de WISPER. El primero es el ruido ya que se suma al pseudo-ruido que entra en la señal produciendo una señal de que no coincide exactamente con el código. El segundo factor que genera errores es la direccionalidad de los transductores. El sistema requiere de transductores que sean bidireccionales, pero esos dispositivos no existen por lo que se decrementa el rendimiento del sistema. El factor final son las condiciones atmosféricas: la variación de la temperatura y las corrientes de aire afectan la precisión del sistema.

# Capítulo 4

## Análisis y diseño del sistema NOTU

En este capítulo trata del diseño y modelado del sistema propuesto en este trabajo de tesis. La sección 4.1 contempla el análisis del problema, como esta constituido el modelo topológico, también se realiza un análisis de la técnica para calcular las distancias. En la sección 4.2 se describe el diseño de los mapas, el cual intenta solucionar dos problemas en particular. Primero, se ataca el problema de distinguir un recurso de otro cuando se encuentran en pisos diferentes. Para resolverlo se representa un modelo topológico con el cual se diseña un edificio de dos pisos.

El segundo problema se refiere a evitar que los usuarios sean "fantasmas", i.e., el sistema esta diseñado para generar una ruta que asemeje a la vida real por lo que dicha ruta no debe atravesar un muro o puerta. Por último, en la sección 4.3 se describe la arquitectura del sistema propuesto así como el funcionamiento de los módulos del dicho sistema Nearest los cuales son: Publicación de recursos, Modificación de recursos y Búsqueda de recursos.

### Sistema Nearest Object To User (NOTU)

Cuando una persona llega a una organización que desconoce, le puede resultar difícil encontrar alguna oficina o recurso que necesita, ya que la mayoría de las veces los edificios no cuentan con los señalamientos necesarios para guiar a las personas. Además resulta imposible (en espacio y costo) concentrar todos los recursos físicos en un sólo punto, por lo que es necesario distribuirlos entre los edificios de la organización.

Cuando varios colaboradores necesitan interactuar con un recurso al mismo tiempo, la coordinación de todos ellos puede volverse una tarea ardua. No obstante, estas tareas pueden automatizarse aplicando tecnología computacional. Al crear un soporte informático para determinar la ubicación de recursos se crea la base de un entorno ubicuo, que es indispensable para hacer más eficiente el trabajo de los miembros de la organización.

El sistema NOTU está diseñado para definir y localizar recursos distribuidos inmersos en un ambiente cooperativo con carácter ubicuo. El sistema localiza los recursos que satisfacen las necesidades del usuario, determina el recurso más cercano a la posición actual del usuario y genera la ruta más corta a dicho recurso.

Para representar un edificio se diseñó un modelo topológico del mismo, el cual consiste en:

- un grafo cuyos los nodos representan las celdas y cuyas aristas denotan la distancia que existe entre dichas celdas y
- las celdas que representan un cuarto o un corredor.

La distancia del usuario al recurso más cercano, será calculada por el método de distancias recorridas, el cual calcula la distancia del recorrido y no en la distancia euclidiana.

Finalmente, se implementó el algoritmo de Dijkstra para determinar la distancia mínima entre el usuario y el recurso. Si existe más de un recurso que satisfaga las necesidades del usuario, el sistema calcula la distancia mínima para cada recurso y elige la menor. Para que el usuario pueda tener una mejor interpretación de la ubicación de los recursos, la información será desplegada mediante mapas en formato vectorial. Por ser un sistema distribuido, se implementó un servidor de mapas en la Web para que un usuario pueda interactuar con el sistema vía PC de escritorio o con algún dispositivo móvil (*tablets, smartphones, etc*).

## 4.1 Análisis

### 4.1.1 Análisis del problema

Al manejar varios niveles (o pisos) resulta complicado relacionar cada recurso con cada nivel. Además, se dificulta la representación gráfica en mapas vectoriales. Por estos motivos se eligió utilizar un Sistema de Información Geográfica (SIG).

Un SIG permite hacer una abstracción de la realidad, ya que tiene la capacidad de reconocer y analizar las relaciones espaciales que existen en la información geográfica almacenada. Estas relaciones topológicas permiten realizar modelizaciones y análisis espaciales complejos.

Para entender el modelo topológico utilizado, primero es necesario mencionar que la topología es el estudio de aquellas propiedades de los cuerpos geométricos que pertenecen inalteradas por transformaciones continuas. En otras palabras, dos objetos serán



equivalentes mientras podamos transformar uno en otro mediante isometrías (rotaciones, traslaciones, reflexiones, etc) o mediante transformaciones que conservan las medidas de ángulo, longitud, área, volumen y otras.

Entonces los modelos topológicos se usan para transformar un objeto en otro usando isometrías. Por ejemplo, los SIG transforman el planeta tierra, que tiene forma de geoide, en un plano cuya representación contiene información adicional a la geometría y tiene una escala, e.g., ríos, ciudades, países, etc.

### 4.1.2 Análisis del modelo topológico

Retomando el problema al manejar varios niveles (o pisos) resulta complicado relacionar cada recurso con cada nivel y calcular la distancia que existe entre los recursos o los pasillos. Por tal motivo se diseñó un modelo topológico del edificio del Departamento de Computación del CINVESTAV-IPN, el cual es el escenario de pruebas.

En este modelo topológico la unidad fundamental se denomina celda, la cual representa un cuarto, oficina o bodega. También puede representar algún pasillo, corredor o escaleras. Un piso ó nivel está representado por un conjunto de celdas aledañas que se encuentran en el mismo nivel o a la misma altura en un edificio.

Cada celda tiene un nombre, el cual corresponde al plano del edificio del Departamento de Computación del CINVESTAV-IPN, la nomenclatura de las celdas está compuesta de la siguiente manera:

[numero\_de\_piso] [seccion] [numero\_de\_oficina]

donde:

**numero\_de\_piso:** representa el número del piso en donde se encuentra la celda

**seccion:** El departamento de Computación se encuentra dividido en secciones que están representadas por letras mayúsculas

**numero\_de\_oficina:** Cada sección asigna un número a las oficinas con las que cuenta

### 4.1.3 Análisis del problema al calcular las distancias

Se puede pensar que la mejor ruta a proponer para llegar a un lugar es la distancia más corta. Sin embargo, la distancia más corta entre dos puntos está determinada por una recta, la cual no puede seguirse en un escenario real que contiene cuartos aledaños.

Para este análisis, se estudia un edificio de dos pisos con espacios cerrados (e.g., oficinas y salones de clase) pero que contienen interconexiones entre ellos (e.g., puertas y pasillos).

Así que, en lugar de obtener la distancia euclidiana entre dos puntos, se propone utilizar la técnica de “distancia de recorrido”, la cual toma en cuenta el trayecto que un usuario recorre para llegar a su destino. Dicho trayecto implica desplazarse en el edificio realizando acciones como salir de una oficina, caminar en un pasillo, subir escaleras, etc. Para calcular la “distancia de recorrido” es necesario ampliar el modelo topológico del edificio del Departamento de Computación añadiendo rutas que representan el recorrido de un usuario.

Para aclarar un poco la problemática, se describe la siguiente situación: el usuario Juan Pérez desea desplazarse de su oficina al área de impresoras, para hacerlo debe salir de la oficina B3, caminar por el pasillo 2, rodear la oficina central, caminar por el pasillo 10 y entrar en el laboratorio. El recorrido de Juan Pérez es la ruta más corta, pero si el recorrido se calcula usando la distancia euclidiana entre dos puntos Juan Pérez tendría que atravesar el muro de su oficina, algunas oficinas y el muro del laboratorio.

La figura 4.1 muestra los dos posibles recorridos de Juan Pérez. Con la línea continua se representa el camino generado usando la técnica de distancia de recorrido, en la Figura 4.1 se representa con el número 1. La línea punteada denota la distancia más corta (euclidiana) entre la oficina de Juan Pérez y el laboratorio, en la figura 4.1 se representa con el número 2.

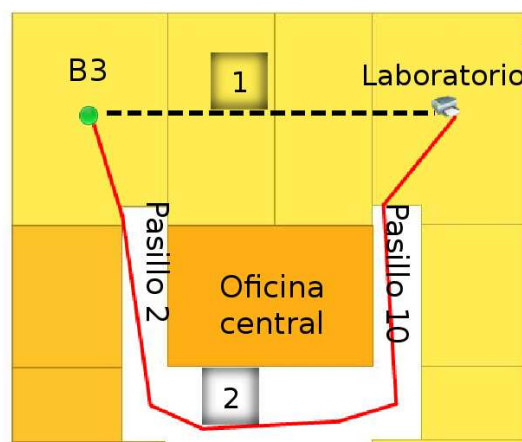


Figura 4.1: Ejemplo de la distancia más corta contra la distancia de recorrido

## 4.2 Diseño

### 4.2.1 Diseño del modelo topológico

Un edificio puede ser definido como un conjunto de elementos (oficinas, corredores, escaleras, etc) los cuales tienen una extensión espacial y una o más conexiones hacia los elementos aledaños.

Una celda es la unidad básica que se puede distinguir en el modelo, esto implica que la localización se realiza en términos de celdas en lugar de coordenadas. Una celda usualmente cubre pequeños cuartos o partes de un corredor. Para este ejemplo, una celda será una oficina o bodega.

Las interconexiones entre las celdas son definidas usando la técnica de distancia de recorrido la cual utiliza:

- puntos centrales que representan el centro de una celda y
- puntos de transición que son usados como puntos auxiliares para desplazarse entre las celdas.

Los puntos de transición son colocados para simular el recorrido real que un usuario debe realizar para desplazarse dentro de un edificio. En otras palabras, es imposible que un usuario atraviese paredes, por eso debe seguir la infraestructura del edificio para poder desplazarse de un lugar a otro

#### Diseño del grafo

Una vez identificados los puntos centrales y los de transición es posible dibujar el grafo, el cual representa el modelo topológico del edificio. Dicho grafo está definido por:

$$B = \{P, E\}$$

Donde:

$P$  es el conjunto de todos los puntos (centrales y de transición) y,

$E$  es el conjunto de distancias entre dos puntos (de transición o centrales).

El conjunto de puntos representan los nodos del grafo, en tanto que las distancias representan las aristas.

## 4.2.2 Diseño de mapas

Para diseñar los mapas se usó la técnica “distancia de recorrido” [Kjaergaard et al., 2008]. La distancia más corta entre dos puntos es una recta, pero, para nuestro ejemplo, el escenario que estudiaremos es un edificio de dos pisos y al momento de calcular la distancia entre dos puntos del mapa, se debe tomar en cuenta el trayecto que un usuario recorre para llegar a su destino. Esto implica salir de su oficina, caminar a través de un pasillo, subir las escaleras, etc. Entonces el usuario no recorre una línea recta, más bien pasa por una serie de puntos los cuales se denominan puntos de transición.

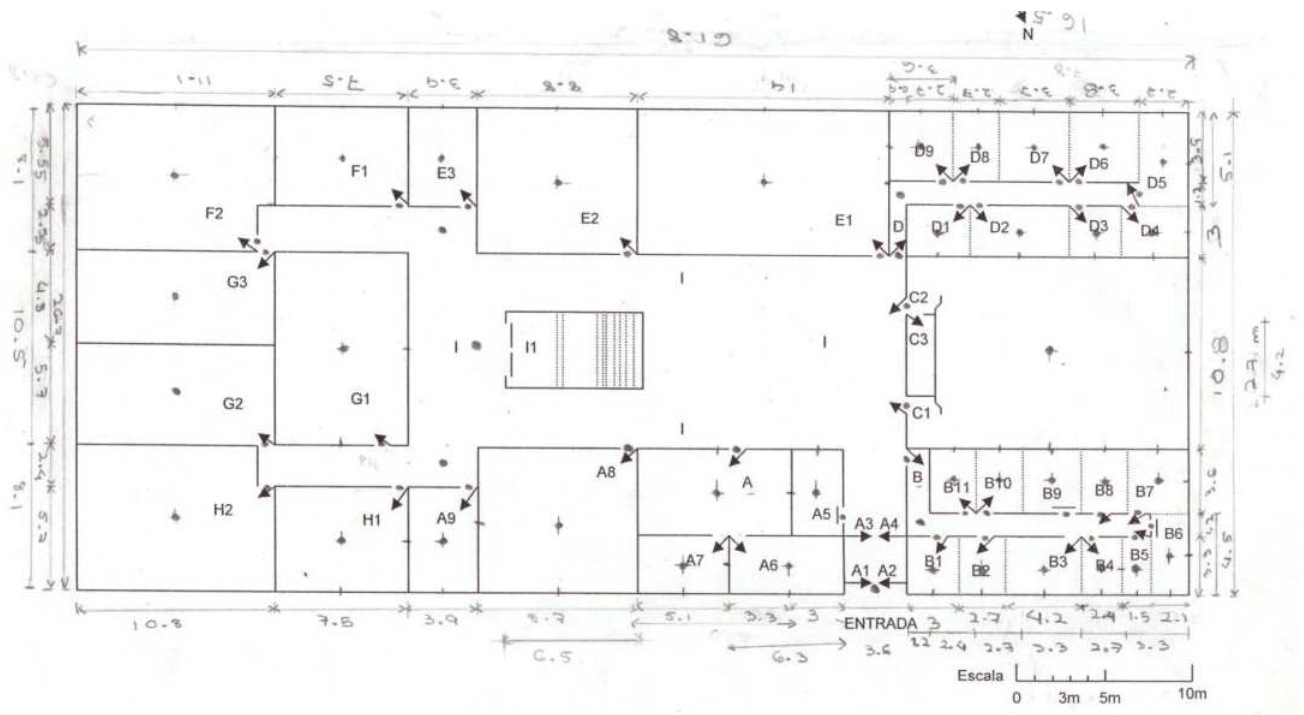


Figura 4.2: Plano de la planta baja del edificio del Departamento de Computación

En la figura 4.2 está representado el plano de la planta baja del edificio de computación, como se puede observar se han medido las oficinas, también se han colocado los puntos centrales de todas estas.

Un punto de transición son todos los puntos “auxiliares” que se colocaron en el mapa para poder calcular una distancia real del recorrido. Para conocer la posición de una oficina se dibujan puntos centrales los cuales fueron calculados usando la escala de los planos del edificio. Una vez identificados los puntos centrales y los de transición se puede dibujar

el grafo.

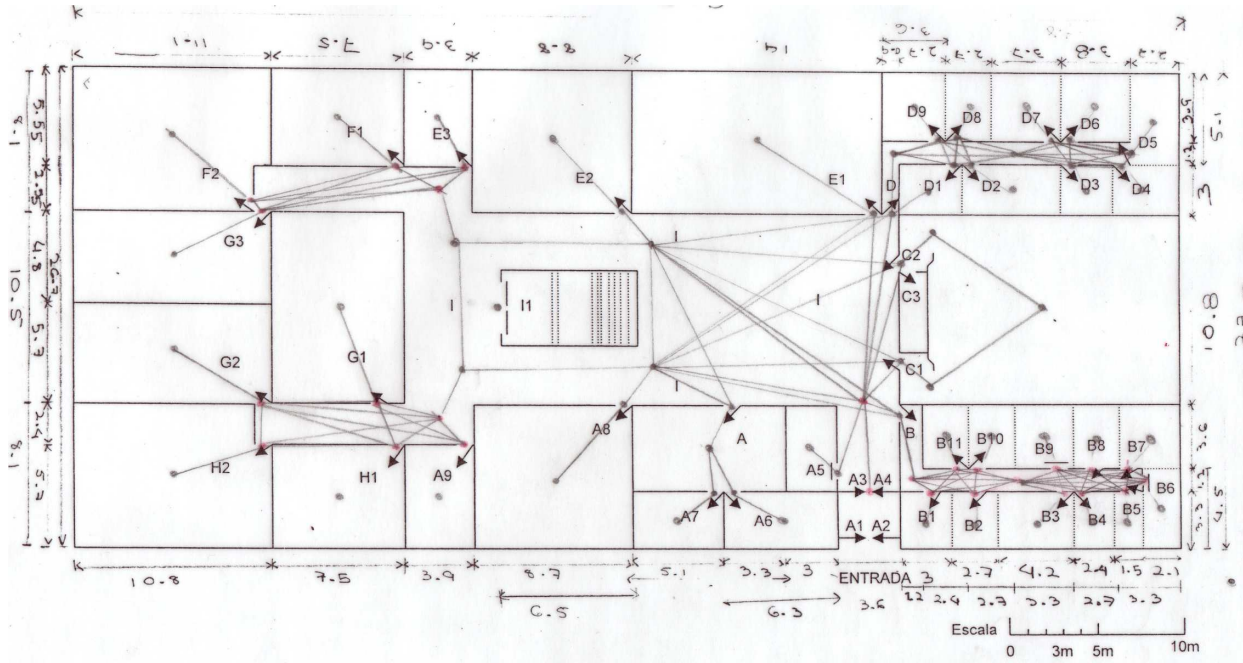


Figura 4.3: Planta baja del edificio del Departamento de Computación con el grafo.

En la figura 4.3 se pueden visualizar los puntos centrales de cada celda y los puntos de transición que fueron colocados para representar el recorrido real de un usuario. Se trazaron líneas entre cada punto central con sus respectivos puntos de transición para conectar todas las celdas. Los puntos representan los nodos del grafo, las líneas trazadas representan las aristas.

Para digitalizar esta información es necesario usar un software que sea capaz de manejar mapas en formato vectorial, ya que estos mapas pueden modelar digitalmente entidades del mundo real. Se utilizó la herramienta Quantim Gis (Q Gis) versión 1.0.2 la cual se distribuye con Licencia Publica General GNU.

Se crearon tres capas en el mapa vectorial, La primer capa corresponde a las celdas (edificios). A cada celda se le asigna un identificador y un nombre de acuerdo al planteado en el modelo topológico. La segunda capa concierne a los puntos centrales y de transición. La tercera capa atañe al grafo que representa las distancias de recorrido.

En la figura 4.4 se puede ver las capas que componen el mapa vectorial. La primera capa del mapa esta representada en la figura 4.4 (a) y corresponde a las celdas del edificio, en la figura 4.4 (b) se puede apreciar la capa de los puntos (centrales y de transición) que

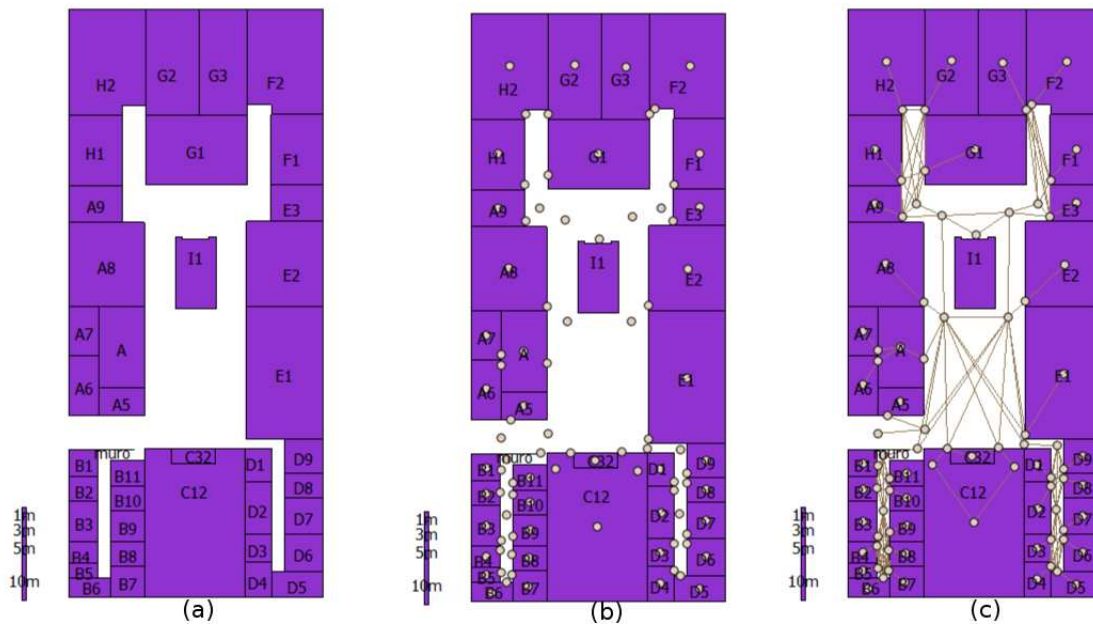


Figura 4.4: Mapa vectorial de la planta baja del Departamento de Computación

esta sobrepuesta en la capa de las celdas del edificio, por último, la figura 4.4 (c) muestra la capa que corresponde al grafo junto con las dos capas anteriores, esto genera un mapa vectorial con tres capas.

Al ser un mapa vectorial, cada capa cuenta con información relevante de su contenido. La capa de las celdas almacena la información del identificador de celda y su nombre; la capa de puntos contiene el identificador de cada punto, su tipo (central o de transición) y su nombre; por último la capa de grafo contiene el identificador de cada arista y la distancia que representa en metros.

### 4.3 Arquitectura del sistema NOTU

El sistema de localización del recurso más cercano está diseñado para que un usuario pueda saber el recorrido que debe seguir para llegar a su objetivo. Este sistema cuenta con varios módulos que le ayudan a realizar este proceso. A continuación se explica la interacción que existe entre dichos módulos. Primero se describe la interfaz de usuario cuya función es recibir las peticiones del usuario y desplegar sus resultados de una manera fácil y amigable. Básicamente la interfaz de usuario recibe tres tipos de peticiones: realizar una publicación, realizar una búsqueda y consultar la modificación de un recurso. Para manejar estas tareas se diseñó un módulo para cada una.

El módulo de publicación está encargado de permitir a un usuario publicar un recurso en el sistema, con la finalidad de que todos los usuarios autorizados puedan conocer las características del nuevo recurso. El módulo de búsqueda también permite que los usuarios puedan solicitar la ubicación algún recurso que está registrado en el sistema y, si existe este recurso, se muestra la ruta más corta para llegar a él.

El módulo de modificación se usa para cambiar la información de los recursos que están publicados; en particular, se pueden cambiar las características (tipo, capacidad, disponibilidad) y su ubicación.

Estos tres módulos están vinculados con otro módulo muy importante que es el de actualización y comunicación. Cuando se modifica la información de los recursos, ya sea por publicación o modificación, los usuarios autorizados que están conectados deben conocer que la información ha cambiado y en consecuencia, la información (el mapa) que ellos manejan se debe actualizar. La figura 4.5 muestra la arquitectura del sistema.

## 4.4 Requerimientos del sistema NOTU

Los requerimientos establecen las propiedades o comportamientos que un sistema debe tener. En la definición de estos requerimientos se deben detallar todas las características del sistema, de una forma clara y sencilla para que todos los integrantes del proyecto puedan entenderlos.

### 1. Requerimientos funcionales:

- (a) El sistema debe mostrar mapas con la ubicación de los recursos.
- (b) El sistema debe proveer información detallada de los recursos (disponibilidad, tipo, etc.).
- (c) El sistema debe desplegar automáticamente (en el mapa) el recurso más cercano que satisface la petición del usuario, tomando en cuenta la disponibilidad de los recursos y cómo llegar a él.
- (d) El sistema debe soportar los cambios en la disponibilidad de los recursos
- (e) El sistema debe ser capaz de actualizar la posición de los recursos, así como añadir nuevos.
- (f) El sistema debe saber en todo momento la posición de los usuarios que usarán el sistema.
- (g) El sistema debe responder a todas las peticiones de los usuarios.



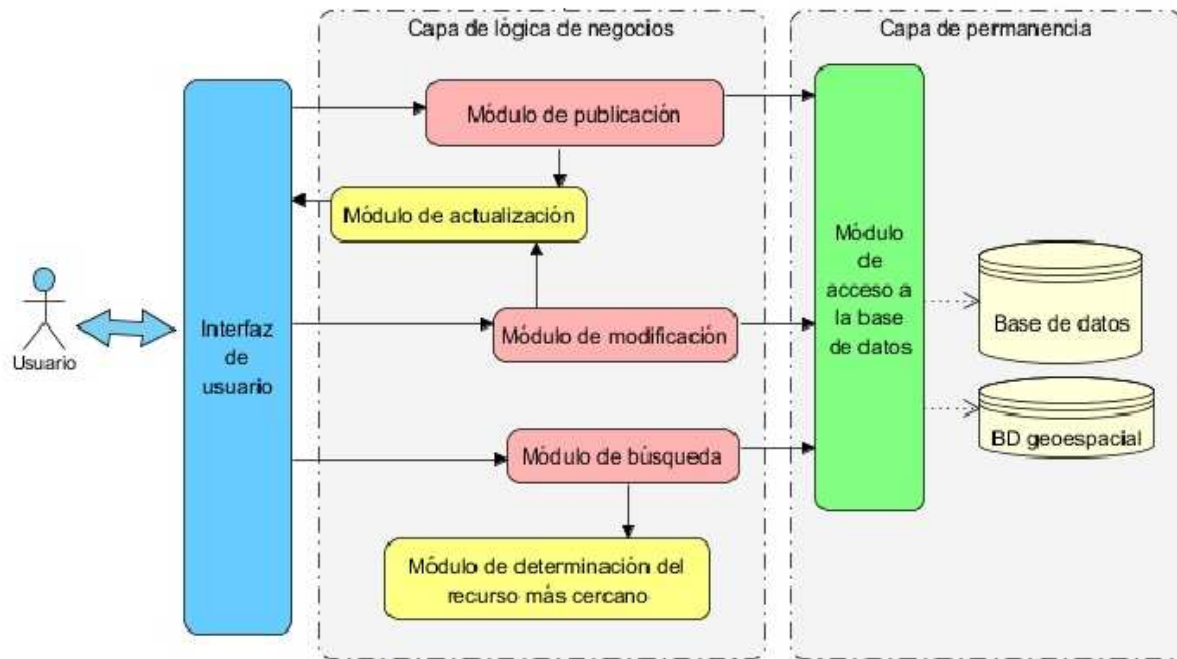


Figura 4.5: Arquitectura del sistema

- (h) El sistema debe almacenar la información de los recursos que maneja.
2. Requerimientos de rendimiento:
    - (a) El tiempo de respuesta del sistema debe ser eficiente (entre 1 y 2 segundos por petición).
  3. Requerimientos de confiabilidad:
    - (a) El sistema debe ser capaz de manejar sesiones de usuario simultáneas.
    - (b) Los cambios que realiza un usuario deben ser reflejados en las instancias de aplicación de todos los usuarios conectados.
    - (c) Las distancias que maneja el sistema deben coincidir con la realidad, incluyendo la distancia de recorrido que un usuario hace para desplazarse de una oficina a otra.
  4. Requerimientos de seguridad:



- (a) El sistema debe comprobar que sólo el usuario que publique un recurso sea el único que puede cambiar su ubicación o la información de este.

#### 5. Requerimientos de hardware:

- (a) La comunicación de los usuarios del sistema será mediante una red LAN ya sea cableada o inalámbrica.

### 4.4.1 Descripción de módulos del sistema NOTU

En esta sección se define el alcance de los módulos del sistema NOTU. El módulo de publicación se describe en la sección 4.4.2, está encargado de incluir nuevos recursos al sistema. El módulo de búsqueda se define en la sección 4.4.3, el cual ayuda a un usuario a realizar una búsqueda de algún recurso que necesite y despliega el más cercano a él. Por último el módulo de modificación se describe en la sección 4.4.4, este módulo despliega la información de los recursos disponibles y permite al usuario modificar su información.

### 4.4.2 Módulo de publicación

En este módulo un agente productor puede publicar un recurso en el sistema. Para publicarlo el usuario debe capturar la información del recurso (tipo, nombre y descripción). Esta información es almacenada en una base de datos.

Una vez que está capturada la información, el usuario debe especificar la ubicación del recurso. Para realizar este proceso el sistema NOTU desplegará, en la pantalla del usuario, el mapa del edificio donde se indicará la posición del recurso.

El sistema asocia el punto central más cercano con la ubicación del nuevo recurso, obteniendo una relación de 1:N. En otras palabras, un punto central puede estar asociado a varios recursos. Dicha asociación facilita la búsqueda de los recursos ya que disminuye el número de nodos.

En la figura 4.6 se puede observar el diagrama de casos de uso del módulo de publicación. En este escenario se representa la interacción que existe entre el agente productor (actor) y el sistema NOTU. Para publicar un recurso, es necesario capturar la información de dicho recurso en un formulario y posteriormente se debe especificar su ubicación en el mapa.

La figura 4.7 muestra el diagrama de secuencias del módulo de publicación en donde el agente productor solicita al sistema NOTU el inicio de una sesión. El servidor compara los datos que introduce el usuario con los que se encuentran almacenados en la base de datos para autorizar el acceso al sistema NOTU. El inicio de sesión es necesario para asignar

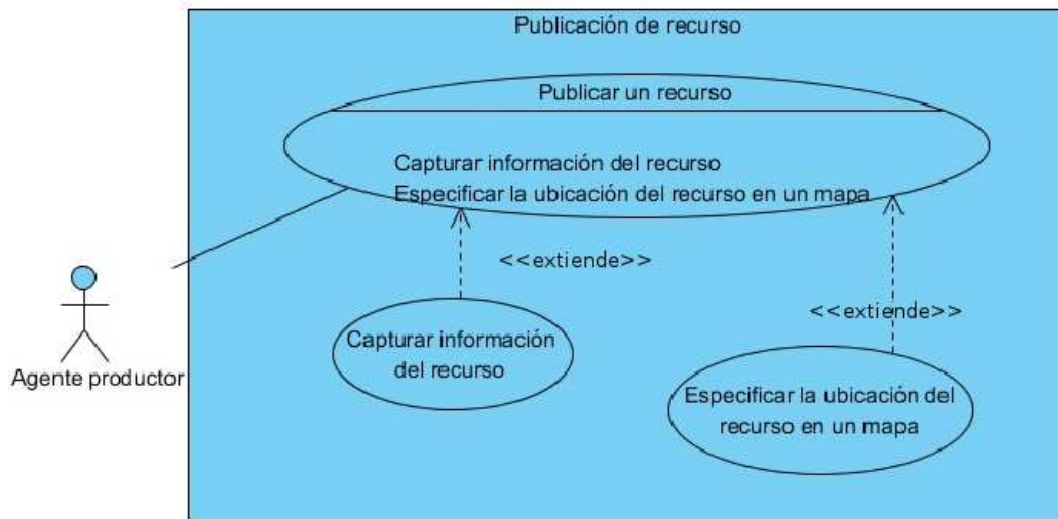


Figura 4.6: Diagrama de casos de uso de la publicación de un recurso

roles a los usuarios, e.g., el usuario 1234 tiene disponibles cinco impresoras.

Una vez que el agente productor accede al sistema, debe realizar una solicitud para publicar un recurso. Como se comentó anteriormente, para publicar un recurso se debe capturar la información mediante un formulario que proporciona el sistema NOTU. Este formulario es almacenado en la base de datos general de dicho sistema.

Para que el agente productor pueda especificar la ubicación del recurso, el sistema NOTU construye dinámicamente el mapa, con la información que se encuentra almacenada en la base de datos. Una vez construido el mapa, el sistema NOTU se lo envía al agente productor como imágenes en formato PNG, para que este pueda definir gráficamente la ubicación del nuevo recurso. Finalmente, el sistema NOTU recibe esta ubicación y la almacena en la base de datos.

### 4.4.3 Módulo de búsqueda

En este módulo el agente consumidor se suscribe a un recurso. Una suscripción es un modelo de negocios en donde un usuario especifica las características del recurso que necesita. En la Figura 4.8 se muestran todos los procesos que se deben realizar para que un agente consumidor realice una suscripción.

Primero el agente consumidor realiza una petición de un recurso, para lo cual es nece-

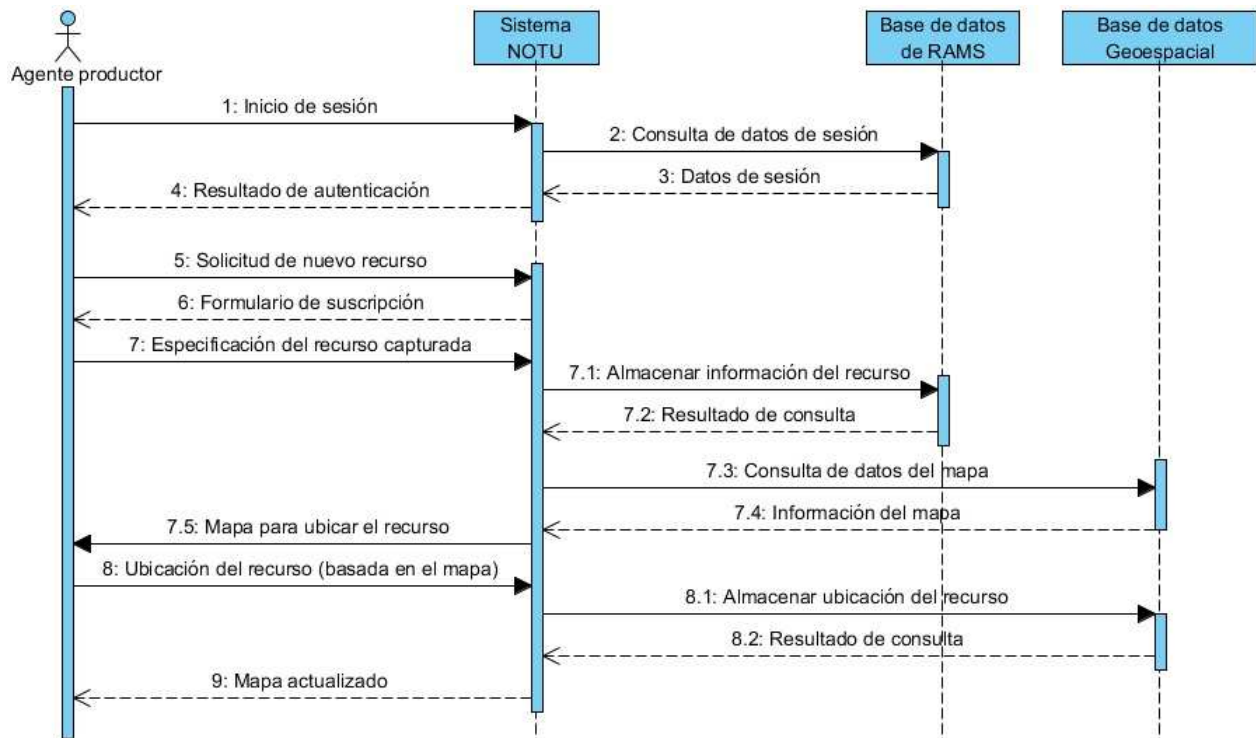


Figura 4.7: Diagrama de secuencia de la publicación de un recurso

sario que capture las especificaciones del recurso que solicita mediante un formulario. El sistema NOTU busca en la base de datos general algún recurso que cumpla las necesidades del agente consumidor usando la información que se proporcionó en el formulario. Si el sistema NOTU encuentra un recurso entonces consulta la base de datos geo-espacial para saber su posición actual.

También el usuario debe introducir su posición actual usando la interfaz de usuario del sistema NOTU. Una vez conocido el punto origen (la posición del agente consumidor) y los puntos destinos (los recursos que satisfacen la petición), el sistema NOTU despliega el recurso más cercano al agente consumidor usando un algoritmo para calcular la ruta más corta el cual se auxilia del grafo con las distancias recorridas. Cuando se encuentra el recurso más cercano, el sistema NOTU puede dibujar sobre el mapa la ruta que el agente consumidor debe seguir para llegar al recurso.

En la figura 4.9 se puede observar con más detalle el procedimiento para efectuar una suscripción. El agente consumidor realiza un proceso de autenticación para ingresar al sistema NOTU. Una vez autenticado, el agente consumidor emite una solicitud para realizar



Figura 4.8: Diagrama de casos de uso de la búsqueda de un recurso

una búsqueda; en respuesta el sistema NOTU envía al agente consumidor un formulario donde debe especificar la información del recurso que está solicitando.

A partir de este punto el sistema realiza tres consultas usando la información obtenida en el formulario:

1. el sistema NOTU consulta la base de datos general, para buscar el recurso que satisfaga las necesidades del agente consumidor,
2. si el recurso es encontrado, el sistema NOTU consulta la base de datos geo-espacial para buscar la ubicación de dicho recurso y
3. el sistema NOTU consulta la base de datos geo-espacial para buscar la ubicación del agente consumidor.

En este momento, el sistema NOTU tiene la información suficiente para poder determinar el recurso más cercano ya que cuenta con el punto origen (ubicación del agente consumidor) y los puntos destino (ubicación de los recursos que satisfacen la petición). Entonces el sistema NOTU hace uso del grafo mencionado anteriormente para determinar el recurso más cercano. Finalmente el sistema envía al agente consumidor el mapa con la

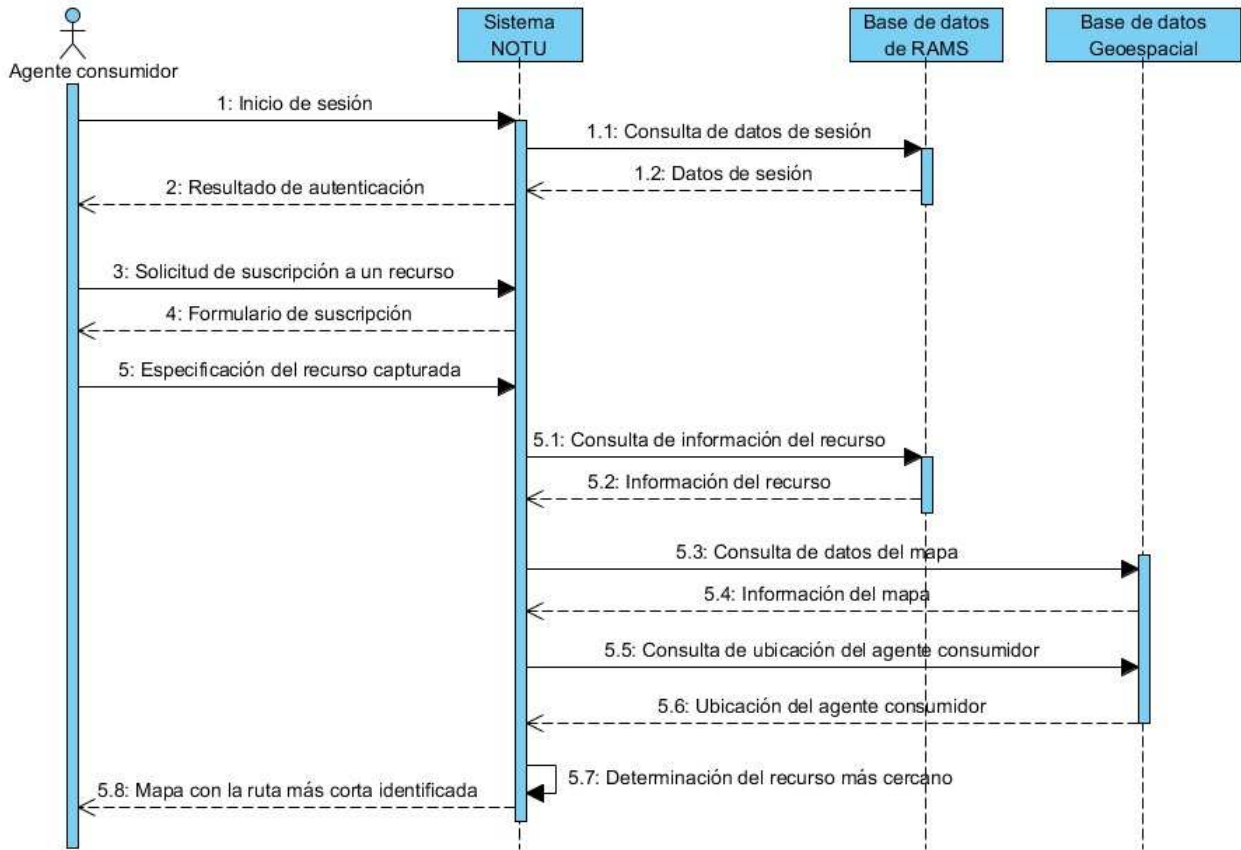


Figura 4.9: Diagrama de secuencia del módulo de búsqueda.

ruta identificada.

#### 4.4.4 Módulo de modificación de un recurso

El módulo de modificación sirve para que un usuario pueda cambiar la información del recurso, su disponibilidad o su ubicación. La figura 4.10 muestra el diagrama de casos de uso para la modificación de un recurso.

Para cambiar la información de un recurso el agente productor debe seguir dos procesos. El primer proceso se trata de modificar la ubicación del recurso, que consiste en: un agente productor realiza una petición de modificación al servidor, el cual busca la información almacenada en las bases de datos geo-espaciales, la despliega en el mapa para que el usuario pueda seleccionar el recurso que desea modificar. Cuando el agente productor escoge un recurso puede cambiar su ubicación, simplemente arrastrándolo en el mapa.

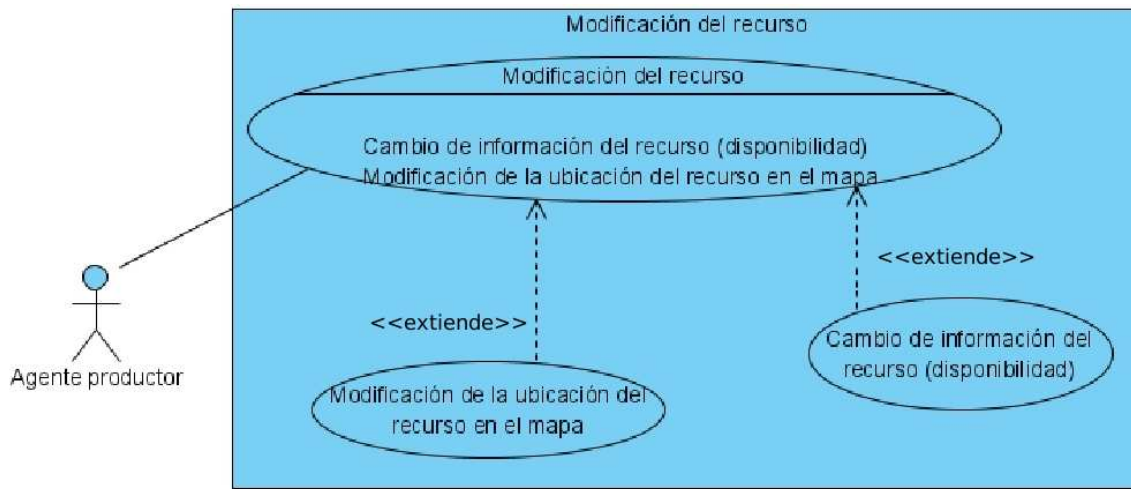


Figura 4.10: Diagrama de casos de uso de la modificación de un recurso

El segundo proceso es la modificación de la información del recurso: una vez que el agente productor modifica la ubicación del recurso, se despliega en pantalla la información del recurso para que el agente productor la modifique y el servidor la almacene en la base de datos general.

En la figura 4.11 se muestra el diagrama de secuencias para la modificación de un recurso. Este diagrama indica las acciones que el agente consumidor debe realizar para modificar un recurso. Al igual que en los módulos anteriores, un agente consumidor debe pasar un proceso de autenticación para ingresar al sistema.

#### 4.4.5 Módulo de búsqueda del recurso más cercano

Para que un usuario pueda saber la ruta más corta es necesario implementar un módulo que sea capaz de generar una ruta usando las distancias que se manejan en el modelo topológico del edificio. Para resolver este problema se usó *pgRouting Project* que es una herramienta para proveer la funcionalidad de encaminamiento al gestor de base de datos *PostGIS/PostgreSQL*; *pgRouting* es parte de *PostLBS*; el cual provee herramientas esenciales para los servicios basados en localización como software de código abierto.

Básicamente este software provee la funcionalidad para que se pueda: 1) crear una base de datos con encaminamiento, 2) crear la topología de los datos y 3) validar los datos. Para crear una base de datos con encaminamiento se necesita información geográfica para formar una red (de caminos) de encaminamiento. Para hacer esta red, se necesita una



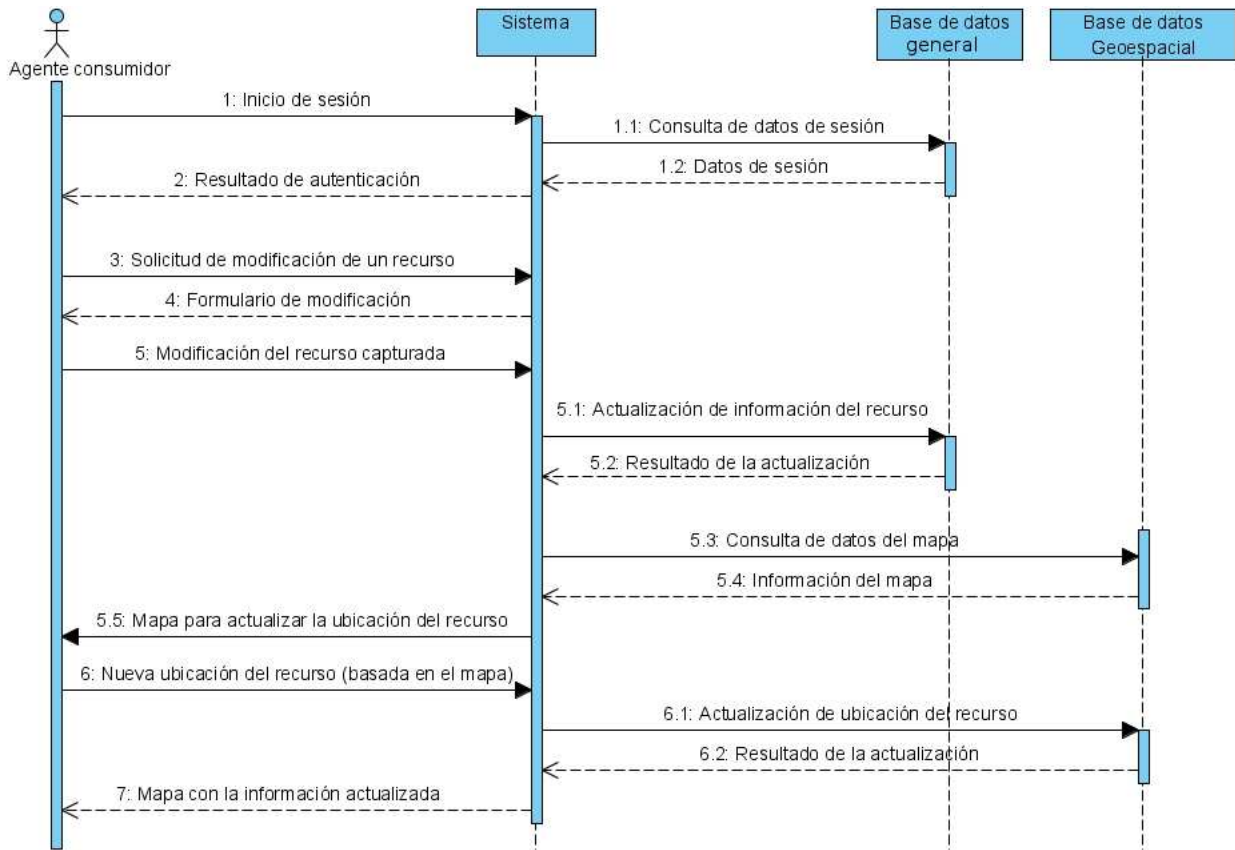


Figura 4.11: Diagrama de secuencias de la modificación de un recurso

topología.

Los motores de encaminamiento requieren los nodos origen y destino por cada línea para crear una búsqueda del camino más corto. Al crear esta información se genera una topología de la red de caminos. Además pgRouting maneja varios algoritmos para resolver el problema del camino más corto, los más importantes son:

- Problema del camino más corto, Dijkstra
- Problema del camino más corto, A\*
- Problema del agente viajero (TSP, *Traveling Sales person Problem*)

Para este módulo se usó el algoritmo de Dijkstra para resolver el problema del camino más corto.





# Capítulo 5

## Implementación del sistema NOTU

Para implementar el sistema denominado *Nearest Object To User* (NOTU) fue necesario instalar y configurar varias herramientas que básicamente se dividen en cinco grupos: 1) las herramientas para la generación de los mapas, 2) el servidor de mapas, 3) las herramientas para la interacción con los mapas, 4) el gestor de base de datos con soporte para información espacial y 5) la herramienta para buscar el recurso más cercano. A continuación se describen las herramientas que se utilizaron, así como su funcionamiento e interacción con las otras.

### 5.1 Herramientas para la generación de mapas

En esta sección se describen las herramientas utilizadas para la implementación del sistema NOTU.

#### Quantum Gis Versión 1.0.2

Para digitalizar la información geográfica y convertirla en un mapa, se utilizó la herramienta *Quantum Gis* versión 1.0.2 que maneja mapas en formato vectorial. Esta herramienta permite modelar digitalmente entidades del mundo real.

Un mapa vectorial está compuesto por tres capas. La primera capa dibuja las celdas, a cada una de las cuales se le asigna un identificador. La segunda capa representa los puntos centrales y de transición. Por último, la tercera capa corresponde a un grafo que representa los caminos formados por los puntos centrales y de transición. Estas capas se muestran en la figura 5.1. Particularmente la figura 5.1(a) corresponde a las oficinas del edificio del Departamento de Computación del CINVESTAV-IPN. En la figura 5.1(b) se observa la capa de los puntos centrales y de transición arriba de la capa de celdas. Finalmente, la figura 5.1(c) muestra el mapa vectorial completo, una vez que se añadió la capa del grafo.

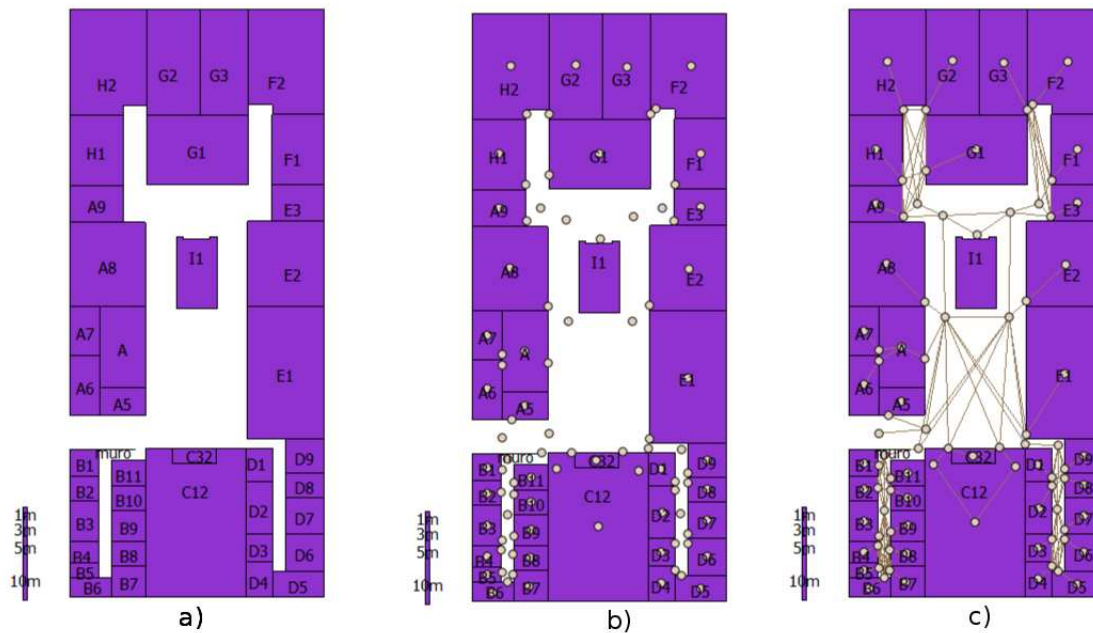


Figura 5.1: Mapa vectorial de la planta baja del Departamento de Computación

Cada capa del mapa vectorial cuenta con información relevante de su contenido. La capa de celdas almacena la información del identificador de cada celda y su nombre. La capa de puntos contiene el identificador de cada punto, su tipo (central o de transición) y su nombre. Por último, la capa del grafo contiene el identificador de cada arista y la distancia representada en metros.

### Comando shp2sql

Al crear un mapa vectorial se genera un archivo con la extensión **shp** (*shapefile*). Este archivo puede ser importado a una base de datos (con soporte geoespacial) usando el comando **shp2psql**. Este comando genera un archivo **SQL** por cada capa del mapa, por lo tanto se obtienen tres archivos **SQL**: el de las celdas, el de los puntos y el del grafo.

El comando **shp2psql** tiene la sintaxis:

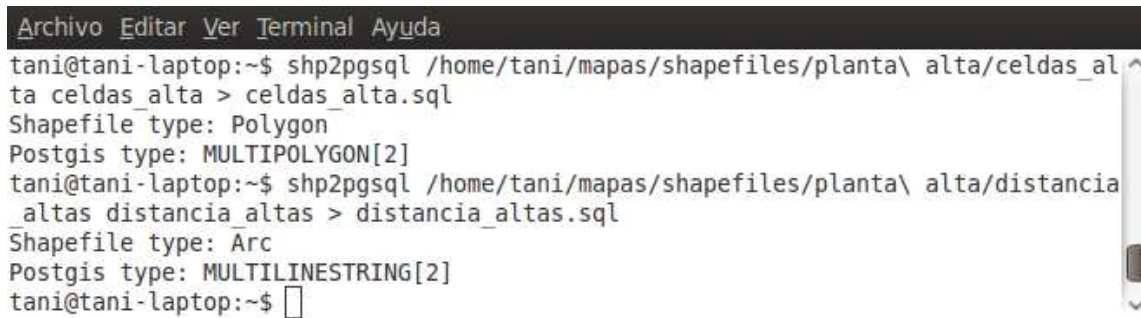
```
shp2psql [ruta_del_archivo_shp] [nombre_de_la_tabla] > [archivo_sql_de_salida]
```

en donde:

- `[ruta_del_archivo_shp]` especifica la ruta en donde se encuentra el archivo **shp** que se desea convertir

- [nombre\_de\_la\_tabla] es el nombre que se asigna a la tabla resultante y
- [archivo\_sql\_de\_salida] es el nombre del nuevo archivo SQL que se genera.

En la figura 5.2 se muestra un ejemplo de la conversión de los archivos `celdas_altas` y `distancias_altas` de tipo `shp` a `SQL`. Estos archivos corresponden al dibujo vectorial de las celdas y al grafo de la planta baja del edificio del Departamento de Computación.



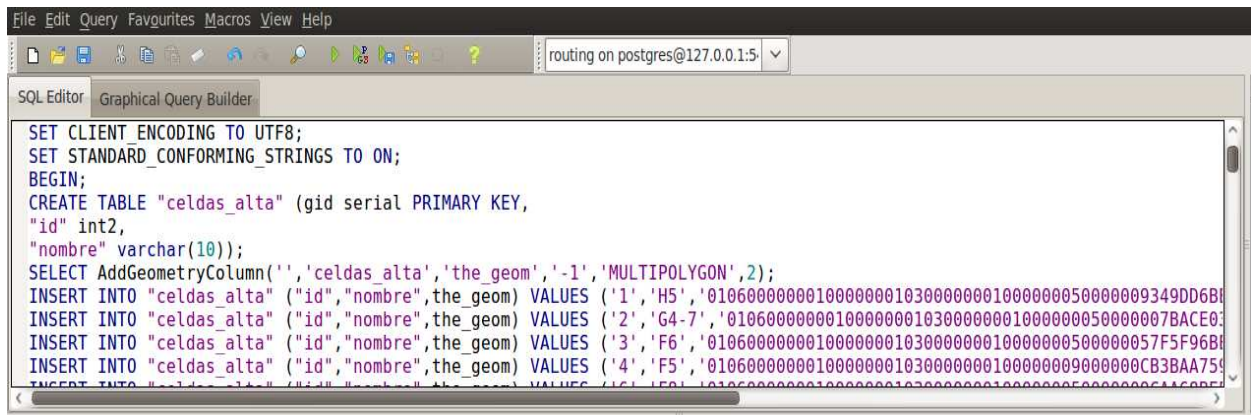
```

Archivo Editar Ver Terminal Ayuda
tani@tani-laptop:~$ shp2pgsql /home/tani/mapas/shapefiles/planta\ alta/celdas_alta
celdas_alta > celdas_alta.sql
Shapefile type: Polygon
Postgis type: MULTIPOLYGON[2]
tani@tani-laptop:~$ shp2pgsql /home/tani/mapas/shapefiles/planta\ alta/distancia
_distancias_altas > distancia_altas.sql
Shapefile type: Arc
Postgis type: MULTILINESTRING[2]
tani@tani-laptop:~$

```

Figura 5.2: Ejemplo del comando `shp2pgsql`

Parte del contenido del archivo `SQL` se muestra en la figura 5.3. El atributo `the_geom` tiene como valor una cadena alfanumérica muy larga que representa el soporte espacial del mapa. i.e., contiene la información referente al objeto espacial e.g., sus coordenadas (expresadas en latitud y longitud), su forma y datos propios del objeto (e.g. nombre, distancia, etc.).



```

SET CLIENT_ENCODING TO UTF8;
SET STANDARD_CONFORMING_STRINGS TO ON;
BEGIN;
CREATE TABLE "celdas_alta" (gid serial PRIMARY KEY,
"id" int2,
"nombre" varchar(10));
SELECT AddGeometryColumn('', 'celdas_alta', 'the_geom', '-1', 'MULTIPOLYGON', 2);
INSERT INTO "celdas_alta" ("id", "nombre", the_geom) VALUES ('1', 'H5', '010600000001000000010300000001000000050000009349DD6B...');
INSERT INTO "celdas_alta" ("id", "nombre", the_geom) VALUES ('2', 'G4-7', '010600000001000000010300000001000000050000007BACE0...');
INSERT INTO "celdas_alta" ("id", "nombre", the_geom) VALUES ('3', 'F6', '0106000000010000000103000000010000000500000057F5F96B...');
INSERT INTO "celdas_alta" ("id", "nombre", the_geom) VALUES ('4', 'F5', '01060000000100000001030000000100000009000000CB3BAA75...');

```

Figura 5.3: Ejemplo de un archivo `SQL`

## 5.2 Herramientas para el servidor de mapas

En esta sección se describen las herramientas utilizadas para la implementación de un servidor de mapas en el sistema NOTU; dicho servidor facilitará las siguientes tareas:

1. **visualización de mapas:** actúa como una interfaz entre los datos espaciales y el usuario. Por un lado, el servidor de mapas maneja la información almacenada en la base de datos (entrada de datos) y por el otro lado despliega esta información como mapas en formato de imagen PNG (salida de datos).
2. **publicación de mapas:** se puede configurar como un Servidor de Mapas Web (WMS por sus siglas en inglés *Web Map Server*)
3. **manejo de información dinámica** (mapas dinámicos): los mapas que despliega el sistema NOTU se generan dependiendo de las necesidades de los usuarios, e.g., se despliega una ruta diferente por cada petición.

### MapServer

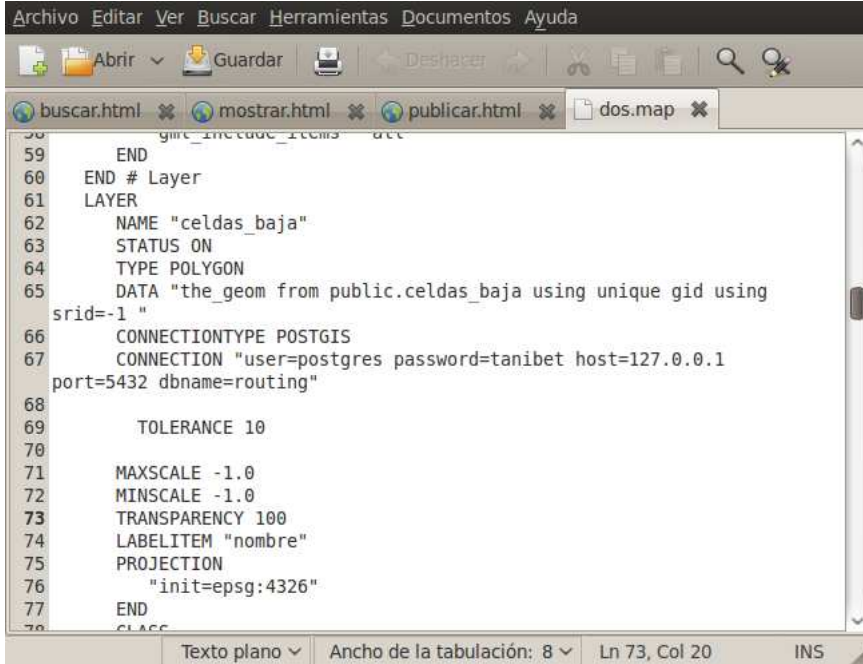
Para publicar la información geográfica e implementar el servidor de mapas se usa la plataforma de código abierto *MapServer*. Esta plataforma es software libre y sirve para publicar datos espaciales, así como para realizar aplicaciones de mapeo interactivo en la Web. En otras palabras, permite crear mapas geográficos dinámicos e interactivos.

Las características principales de *MapServer* son:

1. se ejecuta bajo las plataformas Linux/Apache y Windows;
2. soporta varios formatos vectoriales: ESRI shapefiles, *PostGIS*, ESRI ArcSDE, GML y otros muchos vía OGR;
3. soporta diversos formatos *raster*: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 y otros vía GDAL;
4. se puede configurar en tiempo de ejecución vía URL; y
5. proporciona una API, llamada MapScript, para poder acceder a las funcionalidades de *MapServer* mediante lenguajes de programación como PHP, Java, Perl, Python, Ruby o C#.

Para que *MapServer* pueda publicar los datos, es necesario generar un archivo con la extensión *map* (*mapfile*). Este archivo sirve para especificar el origen de los datos, los símbolos que contiene, el color del mapa, las capas que puede manejar, así como otros

parámetros necesarios para generar un mapa vectorial. La figura 5.4 muestra un ejemplo de un archivo `map` en el que se especifica el origen de los datos, en este caso es una base de datos.



```
59     END
60 END # Layer
61 LAYER
62     NAME "celdas_baja"
63     STATUS ON
64     TYPE POLYGON
65     DATA "the_geom from public.celdas_baja using unique gid using
srid=-1 "
66     CONNECTIONTYPE POSTGIS
67     CONNECTION "user=postgres password=tanibet host=127.0.0.1
port=5432 dbname=routing"
68
69     TOLERANCE 10
70
71     MAXSCALE -1.0
72     MINSSCALE -1.0
73     TRANSPARENCY 100
74     LABELITEM "nombre"
75     PROJECTION
76     "init=epsg:4326"
77     END
78 CLASS
```

Figura 5.4: Ejemplo de un archivo `mapfile`

*MapServer* usa los archivos `map` para generar los mapas, según las características que se especifiquen. Una vez que genera los mapas, *MapServer* los muestra al usuario como imágenes en formato PNG. Estos mapas cambian de acuerdo con las peticiones que realice el usuario y manejan la información contenida en la base de datos para generar la vista del mapa.

### 5.2.1 Interacción con el mapa

En esta sub-sección se describe la librería *OpenLayers* que facilita la interacción del usuario con el mapa de una forma interactiva y dinámica. *OpenLayers* es una librería JavaScript, gratuita y de código abierto.

## OpenLayers

Una vez que el servidor de mapas genera el mapa, se usa la librería *OpenLayers* que facilita la inserción de un mapa dinámico en una página Web. *OpenLayers* puede desplegar los títulos del mapa y los marcadores prácticamente desde cualquier origen de datos.

Básicamente *OpenLayers* facilita la interacción del usuario con los mapas, permitiendo realizar funciones como *zoom*, navegación, manejo de capas e inserción de minimapa entre otras funciones. También permite implementar ventanas *PopUp* con información del mapa.

*OpenLayers* maneja la información que genera *MapServer* para desplegarla dentro de una página Web. Otra ventaja de usar *OpenLayers* es que, al ser *JavaScript*, se ejecuta en algún navegador del cliente, aligerando la carga de trabajo del servidor. En la figura 5.5 se ejemplifica el uso de *OpenLayers*.

Primero se establece la ruta de los archivos de tipo `mapfile` para que *OpenLayers* pueda incrustar el mapa en una página Web. A continuación se crea un nuevo objeto `map` de tipo `OpenLayers Map`, en el que se especifican las dimensiones del mapa, los controles que usarán y las capas que se desplegarán.

Los controles que se insertaron usando *OpenLayers* son: *zoom*, navegación, manejo de capas, minimapa, información de la escala e información de la posición del ratón (coordenadas).

Por último se crean las capas que se mostrarán en el mapa. La variable `celda` contiene un objeto del tipo `OpenLayers.Layer.WMS`, i.e., una capa de tipo *Web Map Server*. También se debe especificar el origen de los datos (en este caso es el archivo `mapfile`), el nombre con el que se identifica y si es visible.

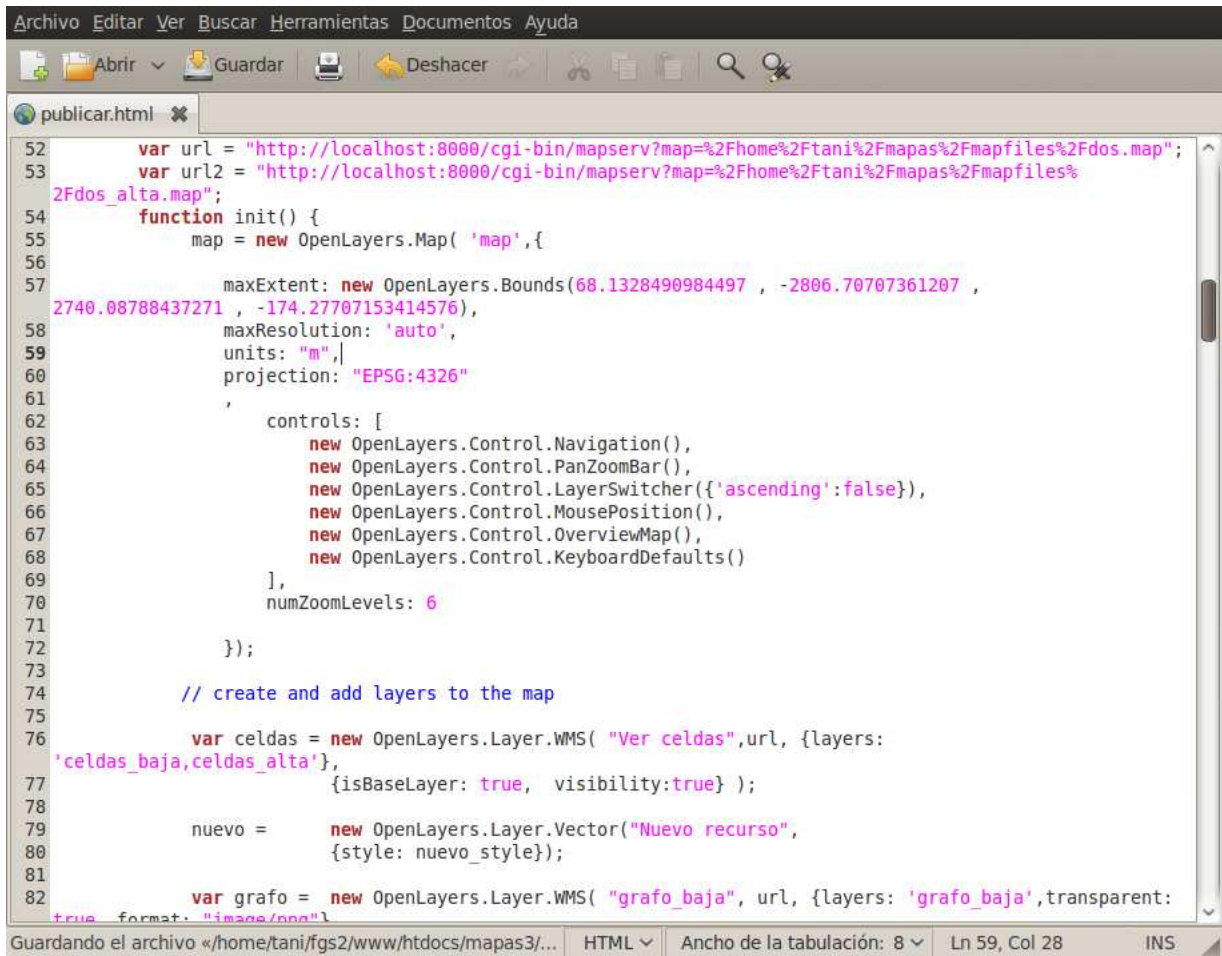
### 5.2.2 Gestión de la base de datos

En esta sub-sección se describe el gestor de base de datos utilizado para gestionar la información del sistema NOTU.

#### PostgreSQL con la extensión PostGIS

Para resolver el problema de la persistencia de datos se usó el gestor de base de datos *PostgreSQL* versión 8.4 con la extensión *PostGIS* para soportar objetos espaciales. *PostgreSQL* es un sistema gestor de base de datos objeto-relacional, de código abierto, multiplataforma y ofrece interfaces de programación para múltiples lenguajes de programación. *PostGIS* añade soporte para objetos espaciales a la base de datos *PostgreSQL*.





```

52     var url = "http://localhost:8000/cgi-bin/mapserv?map=%2Fhome%2Ftani%2Fmapas%2Fmapfiles%2Fdos.map";
53     var url2 = "http://localhost:8000/cgi-bin/mapserv?map=%2Fhome%2Ftani%2Fmapas%2Fmapfiles%
2Fdos_alta.map";
54     function init() {
55         map = new OpenLayers.Map( 'map',{
56
57             maxExtent: new OpenLayers.Bounds(68.1328490984497 , -2806.70707361207 ,
2740.08788437271 , -174.27707153414576),
58             maxResolution: 'auto',
59             units: "m",|
60             projection: "EPSG:4326"
61         },
62         controls: [
63             new OpenLayers.Control.Navigation(),
64             new OpenLayers.Control.PanZoomBar(),
65             new OpenLayers.Control.LayerSwitcher({'ascending':false}),
66             new OpenLayers.Control.MousePosition(),
67             new OpenLayers.Control.OverviewMap(),
68             new OpenLayers.Control.KeyboardDefaults()
69         ],
70         numZoomLevels: 6
71     });
72
73     // create and add layers to the map
74
75     var celdas = new OpenLayers.Layer.WMS( "Ver celdas",url, {layers:
'celdas_baja,celdas_alta'},
76         {isBaseLayer: true, visibility:true} );
77
78     nuevo = new OpenLayers.Layer.Vector("Nuevo recurso",
79         {style: nuevo_style});
80
81     var grafo = new OpenLayers.Layer.WMS( "grafo_baja", url, {layers: 'grafo_baja',transparent:
82         true, format: "image/png"}

```

Figura 5.5: Ejemplo de la librería *OpenLayers*

La figura 5.6 describe los pasos para crear una base de datos espacial. Primero, se crea una base de datos mediante el comando `createdb routing2`, donde `routing2` es el nombre de la base de datos. A continuación, se ejecuta el comando `createlang` para añadir un nuevo lenguaje de programación. Para este ejemplo se agregó el lenguaje `plpgsql` a la base de datos `routing2`. Por último, se ejecutan los *scripts* `postgis.sql` y `sys.sql` para crear las tablas necesarias que brindan soporte de datos espaciales.

Para añadir la información espacial a la base de datos se ejecutan los archivos SQL que se generaron con la herramienta `shp2psql`. Finalmente, en la figura 5.7 se muestra el contenido de la tabla `celdas_altas`.

```

Archivo Editar Ver Terminal Ayuda
tani@tani-laptop:~$ sudo su postgres
postgres@tani-laptop:/home/tani$ /usr/lib/postgresql/8.4/bin/createdb routing2
postgres@tani-laptop:/home/tani$ /usr/lib/postgresql/8.4/bin/createlang plpgsql routing2
postgres@tani-laptop:/home/tani$ /usr/lib/postgresql/8.4/bin/psql -f /usr/share/postgresql
/8.4/contrib/postgis-1.5/postgis.sql -d routing2
Archivo Editar Ver Terminal Ayuda
DROP FUNCTION
DROP FUNCTION
DROP FUNCTION
DROP FUNCTION
postgres@tani-laptop:/home/tani$ /usr/lib/postgresql/8.4/bin/psql -f /usr/share/postgresql
/8.4/contrib/postgis-1.5/spatial_ref_sys.sql -d routing2

```

Figura 5.6: Comandos para agregar soporte espacial a una base de datos

	gid [PK] seria	nombre character	the_geom geometry
1	1	H2	01060000000100000001030000000100000008000000E4
2	2	G2	0106000000010000000103000000010000000600000038
3	3	G3	01060000000100000001030000000100000006000000B8
4	4	F2	0106000000010000000103000000010000000800000040
5	5	F1	01060000000100000001030000000100000006000000BF
6	6	E3	01060000000100000001030000000100000006000000BE
7	7	E2	0106000000010000000103000000010000000600000006

Figura 5.7: Tabla celdas\_alta

### 5.2.3 Soporte de encaminamiento en la base de datos

Una vez que el sistema detecta uno o varios recursos que satisfacen las necesidades del usuario, es necesario determinar cual es el que se encuentra más cerca del usuario, i.e., dado un origen (usuario) y  $n$  destinos (recursos) se encuentra el destino más cercano al origen.

#### pgRouting

Para resolver este requerimiento se usó la herramienta *pgRouting*, la cual provee la



funcionalidad de encaminamiento al gestor de base de datos *PostGIS/PostgreSQL*. La herramienta *pgRouting* es una aplicación gratuita y de código abierto. Básicamente este software provee la funcionalidad de:

1. crear una base de datos con encaminamiento,
2. crear una topología de los datos y
3. validar dichos datos.

### **Creación de una base de datos con encaminamiento**

Para crear una base de datos con encaminamiento primero se necesita una topología de datos. Con base en la topología analizada y diseñada en el capítulo 3 (grafo) se creó la red de encaminamiento en la base de datos. Los motores de encaminamiento requieren nodos origen y destino por cada arista (o línea) del grafo para crear una búsqueda del camino más corto. Al crear esta información se genera una topología de la red de caminos, la cual constituye la representación del grafo.

La figura 5.8 muestra las tablas de la base de datos *routing* en donde se ejecutaron los archivos *celdas\_baja.sql*, *grafo\_baja.sql* y *puntos\_b.sql*. Además se añadieron tres atributos a la tabla: el origen, el destino y la longitud de cada arista. Estos atributos generan una base de datos con encaminamiento.

### **Creación de una topología de datos**

En esta sección se introduce el procedimiento para crear una topología de datos (grafo) usando la herramienta *pgRouting*. Al usar la función *assign\_vertex\_id()* se genera un grafo unidireccional, sin embargo si se desea construir un grafo dirigido se debe establecer el costo de avanzar (del punto A al punto B) y el costo de retroceder (del punto B al punto A). En el caso del sistema NOTU se diseñó un grafo unidireccional, donde el costo de cada arista representa la distancia geográfica de un punto a otro.

La herramienta *pgRouting* permite crear información de origen y destino en *PostgreSQL* usando la función *assign\_vertex\_id()* cuya sintaxis se muestra a continuación:

```
SELECT assign_vertex_id(nombre_tabla, rango_ajuste, columna_geometría,  
                        columna_ejes_id);
```

donde:

- *nombre\_tabla* es el nombre de la tabla que contiene los ejes,

```

Archivo Editar Ver Terminal Ayuda
Digite «help» para obtener ayuda.

postgres=# \c routing
psql (8.4.4)
Ahora está conectado a la base de datos «routing».
routing=# \d

                Listado de relaciones
Esquema |      Nombre      | Tipo   | Dueño
-----+-----+-----+-----
public  | celdas_baja      | tabla  | postgres
public  | celdas_baja_gid_seq | secuencia | postgres
public  | geography_columns | vista  | postgres
public  | geometry_columns | tabla  | postgres
public  | grafo_baja       | tabla  | postgres
public  | grafo_baja_gid_seq | secuencia | postgres
public  | puntos_b         | tabla  | postgres
public  | puntos_b_gid_seq | secuencia | postgres
public  | spatial_ref_sys  | tabla  | postgres
(9 filas)

routing=# ALTER TABLE grafo_baja ADD COLUMN source integer;
ALTER TABLE
routing=# ALTER TABLE grafo_baja ADD COLUMN target integer;
ALTER TABLE
routing=# ALTER TABLE grafo_baja ADD COLUMN length double precision;
ALTER TABLE
routing=# 

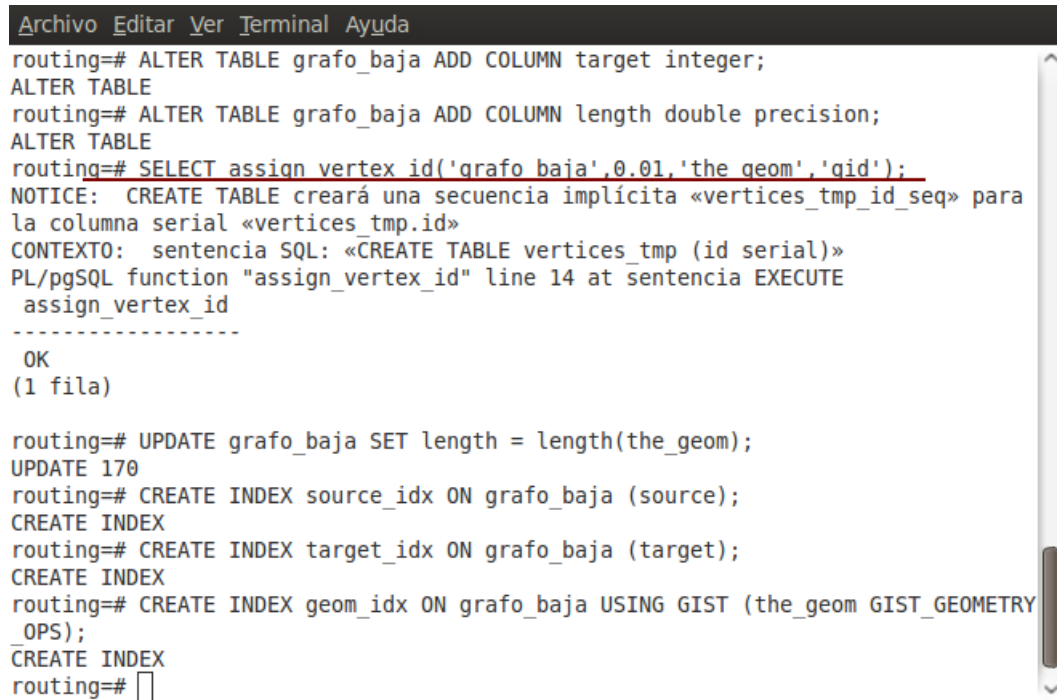
```

Figura 5.8: Pasos para añadir los atributos origen, destino y distancia

- `rango_ajuste` determina el ajuste que se le da a los nodos,
- `columna_geometría` es el nombre de la columna que contiene la información geométrica (usualmente se llama `the_geom`) y
- `columna_ejes_id` es el nombre de la columna que mantiene el identificador de los ejes (generalmente denominada `gid`).

En la figura 5.9 se muestra el uso de la función `assign_vertex_id()`. Esta función modifica la tabla `grafo_baja` (que contiene la información geográfica correspondiente a las distancias de recorrido) añadiendo información de origen-destino a cada nodo, i.e., de qué nodo viene y a qué nodo va.

Una vez realizado este proceso, se actualiza el atributo `length`, que contiene la distancia geométrica entre dos nodos del grafo. Por último se crean los índices `source_idx`, `target_idx` y `geom_idx` para agilizar la obtención de los datos de la tabla.



```

Archivo Editar Ver Terminal Ayuda
routing=# ALTER TABLE grafo_baja ADD COLUMN target integer;
ALTER TABLE
routing=# ALTER TABLE grafo_baja ADD COLUMN length double precision;
ALTER TABLE
routing=# SELECT assign_vertex_id('grafo_baja',0.01,'the_geom','gid');
NOTICE: CREATE TABLE creará una secuencia implícita «vertices_tmp_id_seq» para
la columna serial «vertices_tmp.id»
CONTEXTO: sentencia SQL: «CREATE TABLE vertices_tmp (id serial)»
PL/pgSQL function "assign_vertex_id" line 14 at sentencia EXECUTE
assign_vertex_id
-----
OK
(1 fila)

routing=# UPDATE grafo_baja SET length = length(the_geom);
UPDATE 170
routing=# CREATE INDEX source_idx ON grafo_baja (source);
CREATE INDEX
routing=# CREATE INDEX target_idx ON grafo_baja (target);
CREATE INDEX
routing=# CREATE INDEX geom_idx ON grafo_baja USING GIST (the_geom GIST_GEOMETRY
OPS);
CREATE INDEX
routing=# █

```

Figura 5.9: Creación de la topología de datos

### 5.2.4 Implementación del algoritmo de Dijkstra

Para obtener el recurso más cercano al usuario, primero se calcula la ruta más corta entre dos puntos. El primer punto es la ubicación del usuario y el segundo punto es la ubicación de cada recurso. Por lo tanto si hay  $n$  recursos que satisfacen la petición del usuario se deben calcular  $n$  rutas mínimas. Cuando se tienen las  $n$  distancias más cortas se clasifican para obtener la más cercana al usuario.

Para llevar a cabo este proceso, se usa el algoritmo de Dijkstra que realiza una búsqueda de la distancia mínima entre dos puntos. Este algoritmo es implementado usando la herramienta *pgRouting* la cual realiza una consulta a la base de datos mediante la función `shortest_path`. Esta función usa el algoritmo de Dijkstra para calcular la ruta más corta y su sintaxis es:

```

CREATE OR REPLACE FUNCTION shortest_path(sql_text, source_id integer,
target_id integer, directed boolean, reverse_cost boolean)
RETURNS SETOF path_result

```

Los argumentos son:

- `sql_text`: es una consulta SQL con la siguiente sintaxis:

```
SELECT id, source, target, cost FROM edge_table
```

y regresa un conjunto de elementos con las siguientes columnas:

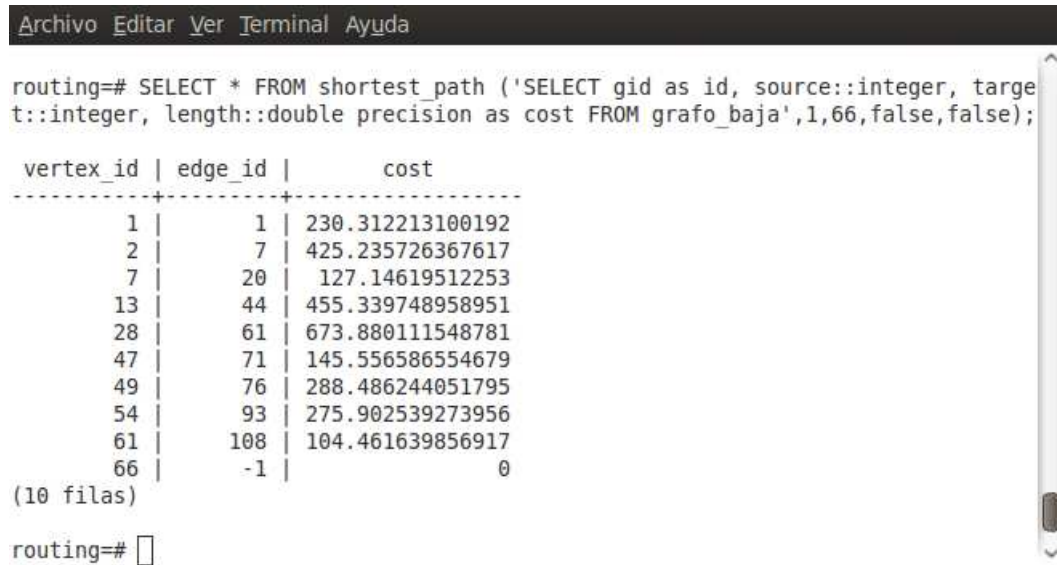
- `id`: un dato de tipo `int4` que es el identificador del eje
  - `source`: un dato de tipo `int4` que identifica el vértice origen
  - `target`: un dato de tipo `int4` que identifica el vértice destino
  - `cost`: un dato de tipo `float8` que es el costo transversal del eje
- `source_id`: un dato de tipo `int4` que es el identificador del punto de origen,
  - `directed`: si es `true` el grafo es dirigido,
  - `has_reverse_cost`: si es `true` el costo de regreso genera un conjunto de filas que serán usadas como el costo transversal del eje.

La figura 5.10 muestra el uso de la función `shortest_path` para obtener la distancia mínima del punto origen al destino. Para este ejemplo el punto origen es el 1 y el punto destino es el 66. Esta función regresa una tabla en donde el conjunto de filas corresponde a las aristas que constituyen la ruta más corta, en tanto que el conjunto de columnas esta formado por los siguientes elementos:

- `vertex_id`: identifica el punto origen de cada arista,
- `edge_id`: identifica el eje transitado,
- `cost`: es el costo asociado a cada eje, así el trayecto final puede ser calculado sumando el costo de todos los ejes.

### 5.3 Interacción de las herramientas

En esta sección se describe la interacción entre las herramientas utilizadas para la implementación del sistema NOTU, así como el flujo de información entre ellas. La interacción entre las herramientas que actúan como interfaz entre el usuario y la información geográfica está reflejada en la figura 5.11, la cual se divide en dos partes. El recuadro izquierdo representa las herramientas que emplea el cliente, mientras que el recuadro derecho engloba todas las herramientas que emplea el servidor.



```

Archivo  Editar  Ver  Terminal  Ayuda

routing=# SELECT * FROM shortest_path ('SELECT gid as id, source::integer, target::integer, length::double precision as cost FROM grafo_baja',1,66,false,false);

 vertex_id | edge_id |      cost
-----+-----+-----
         1 |         1 | 230.312213100192
         2 |         7 | 425.235726367617
         7 |        20 | 127.14619512253
        13 |        44 | 455.339748958951
        28 |        61 | 673.880111548781
        47 |        71 | 145.556586554679
        49 |        76 | 288.486244051795
        54 |        93 | 275.902539273956
        61 |       108 | 104.461639856917
        66 |        -1 |              0
(10 filas)

routing=# 

```

Figura 5.10: Resultado de la consulta usando la función `shortest_path`

La herramienta *MapServer* es, como su nombre lo indica, un servidor de mapas. En el momento que un usuario entra al sistema NOTU, realiza automáticamente una petición para visualizar los mapas disponibles. Entonces el navegador Web se conecta con *MapServer*, mediante *OpenLayers*. *MapServer* tiene definido el origen de los datos geográficos, en este caso es una base de datos espacial. Posteriormente, el sistema NOTU realiza una consulta a la base de datos espacial para obtener toda la información necesaria para generar los mapas.

Una vez que *MapServer* genera los mapas, con base en los parámetros preestablecidos, regresa esta información como una imagen de formato PNG, así el usuario puede ver el mapa solicitado en su navegador.

En este punto, el usuario puede elegir la operación que desea realizar en los mapas; básicamente puede efectuar tres operaciones:

1. **publicar un nuevo recurso:** esta tarea está a cargo del usuario que maneja o controla los recursos; se debe especificar la ubicación del nuevo recurso así como las características del mismo.
2. **buscar un recurso disponible:** el usuario realiza una petición o una búsqueda de algún recurso disponible en el sistema NOTU; como resultado, se muestra la ruta más corta al recurso solicitado si existe y

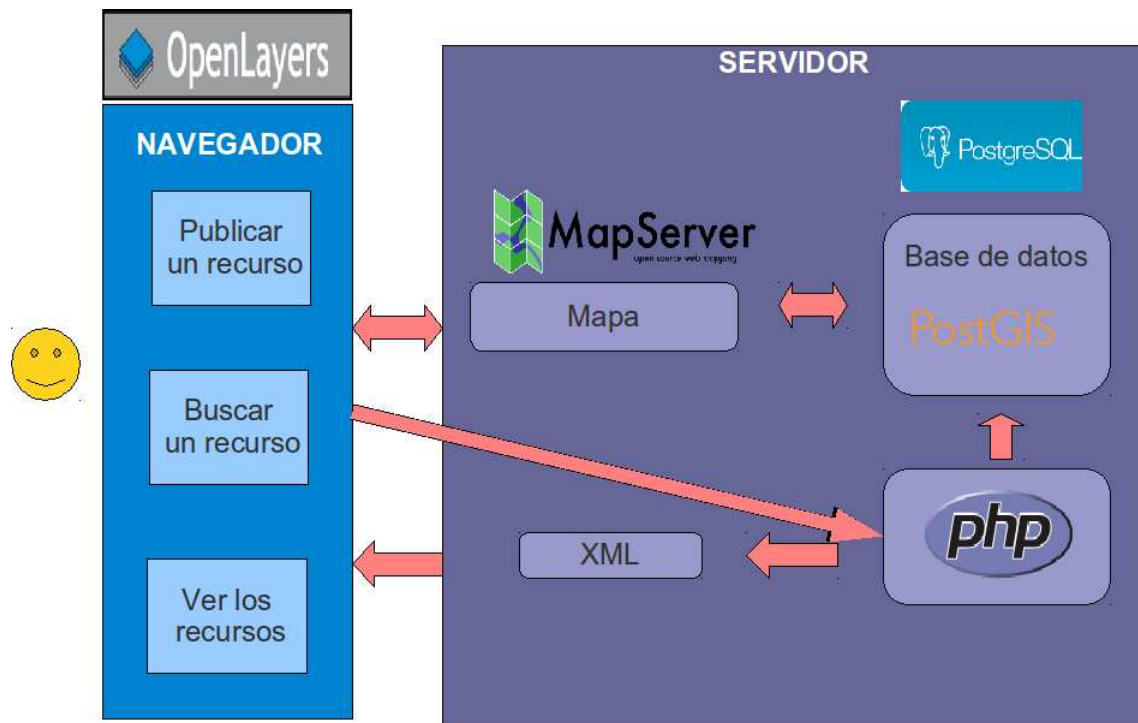


Figura 5.11: Interacción de las herramientas

3. **consultar la información de los recursos disponibles:** el sistema NOTU despliega los recursos en el mapa para que el usuario pueda consultarlos.

Para realizar todas estas opciones se lleva a cabo una interacción directa del usuario con los mapas vectoriales. Esta interacción con el mapa es implementada mediante la herramienta *OpenLayers* que tiene las siguientes ventajas:

- es un ambiente meramente gráfico,
- se ejecuta del lado del cliente, lo cual aligera la carga de trabajo del servidor, y
- permite manejar una lógica de negocios por cada operación.

Cada vez que el usuario realiza una petición, el servidor usa el lenguaje PHP para consultar o modificar los datos almacenados en la base de datos. Posteriormente regresa al usuario un documento XML, el cual contiene la información correspondiente al resultado de dicha petición.

Al publicar o modificar un recurso, el cliente realiza una petición al servidor especificando la ubicación y la información correspondiente al recurso. Como resultado el servidor

procesa esta información y la almacena usando la extensión *PostGIS* para manejar los datos espaciales.

Cuando un usuario realiza la búsqueda de un recurso, el servidor recibe una petición en donde se especifica la ubicación del usuario y el tipo de recurso que este necesita. Entonces el sistema NOTU realiza una consulta a la base de datos para obtener los recursos que satisfacen estas necesidades. Una vez que se obtiene la información de dichos recursos se puede calcular la ruta más corta usando el punto origen (ubicación del usuario) y el punto destino (ubicación de los diferentes recursos que satisfacen la petición).

Para calcular la ruta más corta el servidor usa el lenguaje PHP para comunicarse con la herramienta *PgRouting* y enviarle los puntos origen y destino. *PgRouting* calcula la distancia mínima usando el algoritmo de Dijkstra sobre la base de datos con encaminamiento.

Como resultado de una petición, el servidor regresa al cliente un documento XML. Dicho documento es interpretado por el navegador quien usa *OpenLayers* para interpretar la información del documento y así generar el nuevo mapa. Por ejemplo, cuando un usuario busca algún recurso, el servidor regresa un documento XML que contiene la ruta que *OpenLayers* genera sobre el mapa.

## 5.4 Diagrama entidad-relación

La figura 5.12 muestra el diagrama entidad-relación de la base de datos denominada *routing2* del sistema NOTU. Este diagrama es útil para mostrar la interacción de las nueve tablas que la conforman. A continuación se describen las tablas, sus atributos y relaciones.

### 5.4.1 Entidad *Geometry\_Columns*

Esta entidad se genera automáticamente cuando se crea una base de datos espacial donde se almacena el nombre de la tabla que contiene información geográfica. Por tal motivo, por cada entidad con datos geográficos que se genere, se almacena el nombre de esa entidad dentro de la columna `f_table_name`. Los atributos de la entidad *Geometry\_Columns* son:

- `srid`: es un identificador único de una tabla geográfica,
- `f_table_schema`: contiene un atributo que almacena el esquema de la tabla,
- `f_table_name`: almacena el nombre de la tabla con información geográfica,



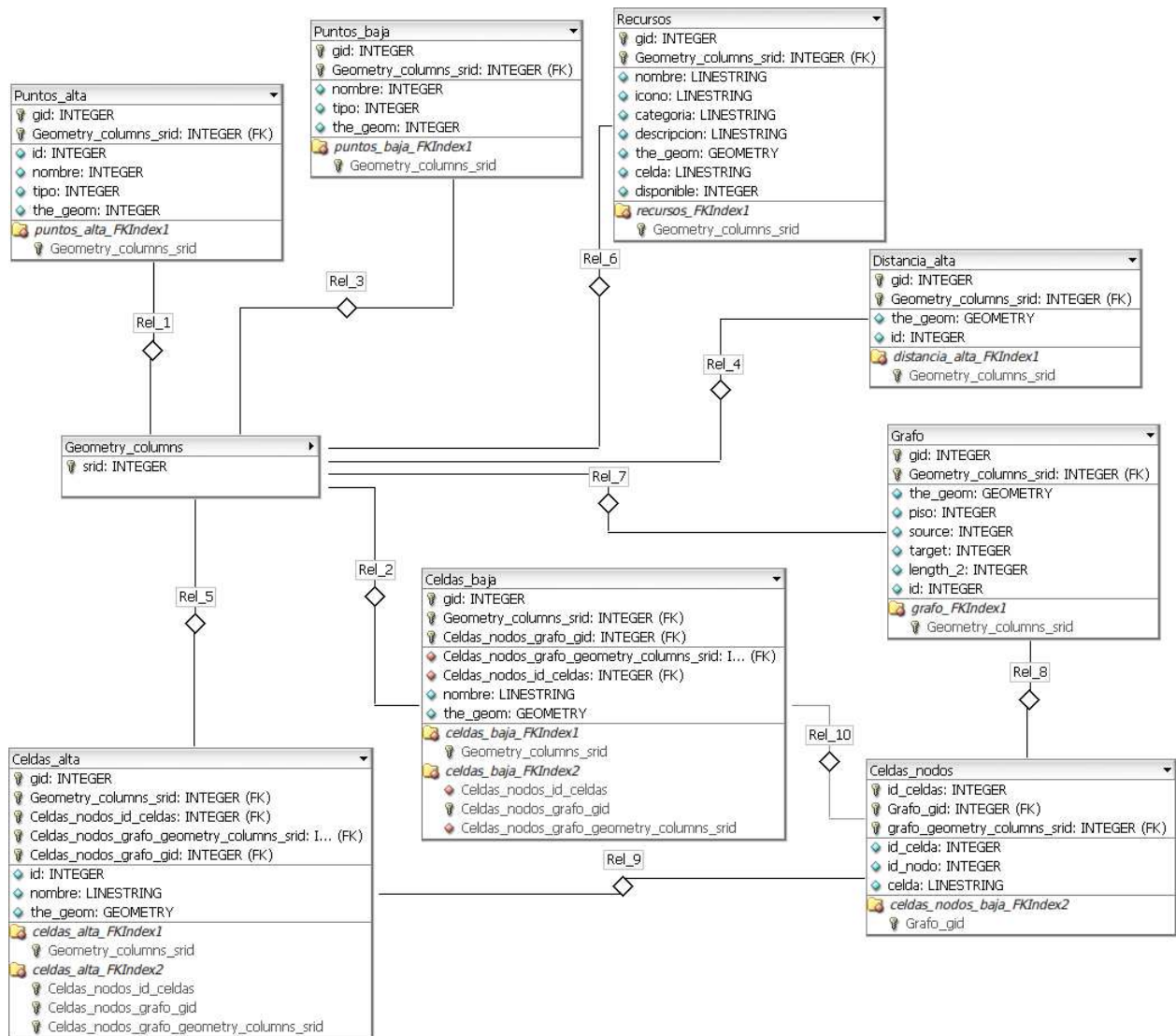


Figura 5.12: Diagrama entidad-relación de la base de datos routing2

- `f_geometry_column`: guarda el nombre de la columna con información geográfica de la tabla,
- `coord_dimension`: describe el número de dimensiones de la tabla con información geográfica,
- `Oid`: representa el identificador del registro,
- `Type_2`: define el tipo de geometría (*point*, *multiline* o *polygon*).



### 5.4.2 Entidad Puntos\_Alta

Esta entidad almacena la información geográfica de los puntos centrales y de transición de la planta alta del edificio del Departamento de Computacion. Cada punto tiene un nombre en específico: si el punto es central, su nombre será igual que al de la celda a la que pertenece, de lo contrario al final de su nombre se coloca una letra *t* para indicar que es un punto de transición. Los atributos de la entidad **Puntos\_Alta** son:

- **gid**: guarda un identificador único para un punto,
- **id**: contiene un identificador único para un punto contenido en el mapa,
- **nombre**: almacena el nombre de un punto en el mapa,
- **tipo**: indica si es un punto central o de transición,
- **the\_geom**: maneja la información geográfica del punto.

### 5.4.3 Entidad Puntos\_Baja

Esta entidad es muy similar a la entidad **Puntos\_Alta**, la única diferencia es que almacena la información de los puntos centrales y de transición de la planta baja del mencionado edificio del Departamento de Computación. Sus atributos son:

- **gid**: contiene un identificador único para un punto,
- **nombre**: es el nombre del punto,
- **tipo**: indica si es un punto central o de transición,
- **the\_geom**: representa la información geográfica del punto.

### 5.4.4 Entidad Celdas\_Alta

Esta entidad contiene la información de las celdas de la planta alta de dicho edificio. Cada celda tiene un nombre específico que fue asignado conforme al nombre que se maneja en el plano. Además esta entidad tiene una relación de 1:1 con la entidad **Celdas\_Nodos\_baja** (ver figura 5.12 rel\_9). Esta relación ayuda a identificar qué nodo corresponde a qué celda. Sus atributos son:

- **gid**: maneja un identificador único para cada polígono (celda),
- **id**: almacena un identificador único para cada celda,
- **nombre**: contiene el nombre de la celda,
- **the\_geom**: guarda la información geográfica de la celda.

### 5.4.5 Entidad Celdas\_Baja

Al igual que la entidad `Celdas_Alta`, contiene la información de las celdas pero de la planta baja de dicho edificio y también tiene la misma relación de 1:1 con la entidad `Celdas_Nodos_baja` (ver figura 5.12 rel\_10). Esta relación ayuda a identificar qué nodo corresponde a qué celda. Sus atributos son:

- `gid`: representa un identificador único para cada polígono (celda),
- `nombre`: es el nombre de la celda,
- `the_geom`: guarda la información geográfica de la celda.

### 5.4.6 Entidad Distancias\_Alta

Esta entidad muestra la información de la distancia o del camino entre las celdas de la planta alta del edificio y sólo contiene información geográfica. Sus atributos son:

- `gid`: es un identificador único para una línea (camino),
- `id`: contiene un identificador único para cada línea,
- `the_geom`: indica la información geográfica del camino.

### 5.4.7 Entidad Recursos

Esta entidad almacena la información de los recursos, así como su ubicación geográfica. Sus atributos son:

- `gid`: maneja un identificador único para un punto (recurso),
- `nombre`: es el nombre que el usuario asignó al recurso,
- `icono`: indica la ruta en donde está almacenada la imagen del recurso,
- `categoría`: guarda la categoría que el usuario asignó al recurso,
- `descripcion`: es una breve explicación de las características del recurso,
- `the_geom`: contiene la información geográfica del recurso (punto),
- `celda`: representa la celda en la que fue dado de alta el recurso,
- `disponible`: registra el valor de 1 si el recurso está disponible, de lo contrario almacena el valor de 0.

### 5.4.8 Entidad Grafo

Esta entidad contiene toda la información para calcular la ruta más corta. Básicamente todos los caminos del mapa se juntaron en esta entidad para manejar un sólo grafo para todo el edificio. Sus atributos son:

- **gid**: es un identificador único para una línea (arista)
- **the\_geom**: contiene la información geográfica de la línea (arista)
- **piso**: identifica en que piso se encuentra la arista.
- **source**: representa el nodo origen de la arista (de donde viene)
- **target**: registra el nodo destino de la arista (a donde va)
- **length\_2**: indica la longitud (o costo) del elemento
- **id**: almacena un identificador de la arista.

### 5.4.9 Entidad Celdas\_Nodos\_baja

Esta entidad sirve para asociar una celda con su punto central. Tiene una relación 1 a 1 con la entidad **Grafo**, por cada nodo que se crea, se almacena su identificador en la entidad **Celdas\_Nodos\_baja** en el atributo **id\_nodo** el cual se asocia con la celda a la que corresponde. Esta tabla se generó manualmente. Sus atributos son:

- **id\_celda**: es un identificador único de la celda
- **id\_nodo**: representa un identificador único del nodo
- **celda**: indica nombre de la celda

## 5.5 Descripción del ambiente gráfico

En esta sección se describen los elementos que componen el ambiente gráfico del sistema NOTU. En la subsección 5.6.1 se describe la interfaz del usuario la cual facilita la interacción del usuario con el sistema; en la sección 5.6.2 se describen los elementos para la interacción con el mapa; en la subsección 5.6.3 se describen los pasos para publicar un recurso usando la interfaz del usuario; en la subsección 5.6.4 se describe el proceso para modificar la información (o ubicación) de un recurso. En la subsección 5.6.5 se describe el procedimiento para realizar una búsqueda del recurso más cercano disponible

### 5.5.1 Interfaz de usuario

Para poder usar el sistema NOTU se debe contar con una PC o un dispositivo móvil que pueda conectarse a Internet. También es necesario que se tenga instalado algún navegador Web, e.g., Internet Explorer, Mozilla, Chrome o Safari.

El primer paso es ingresar a la página : <http://azay.cs.cinvestav.mx/~tanibet/mapas> en donde se verá la pantalla de la figura 5.13

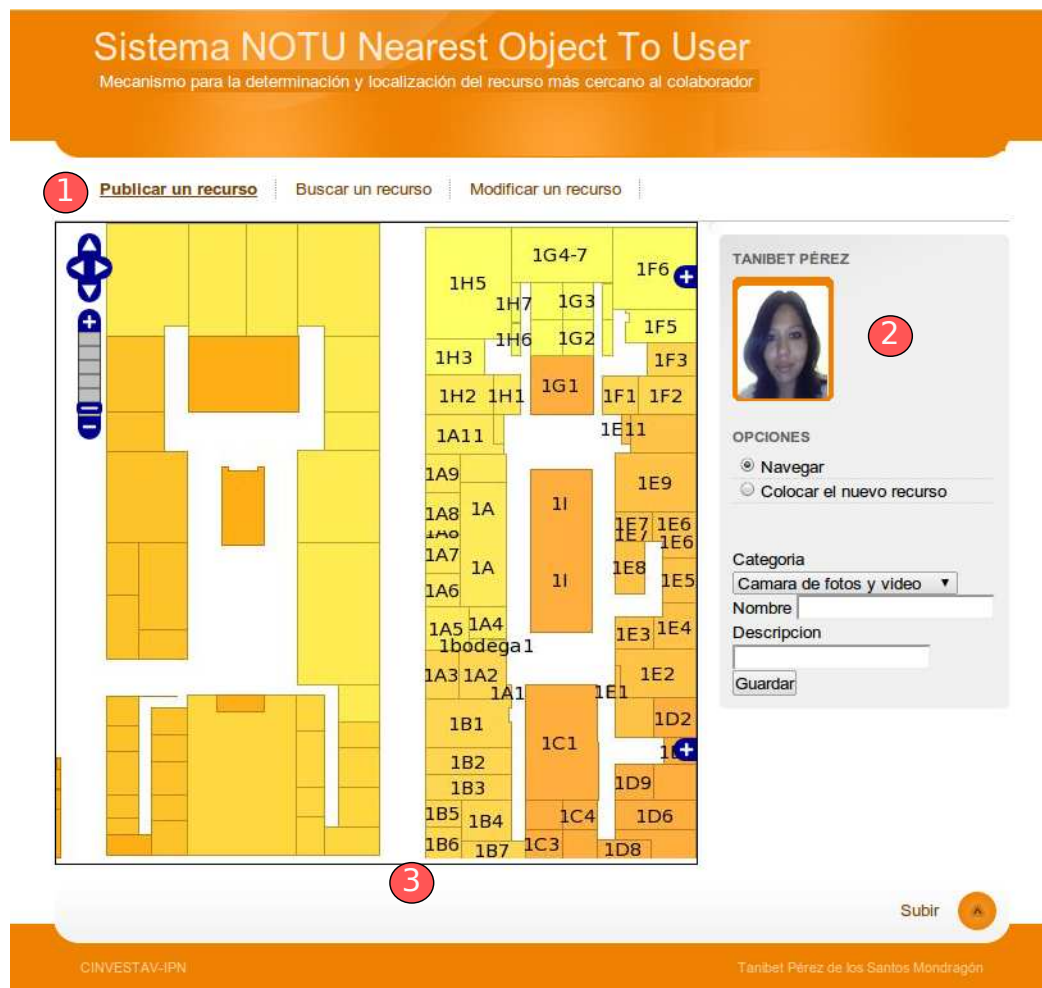


Figura 5.13: Vista general de la interfaz gráfica del sistema NOTU

A continuación se describe el ambiente gráfico del sistema NOTU:

1. **Menú general:** contiene las opciones que un usuario puede elegir en el sistema. Básicamente puede publicar un nuevo recurso, buscar un recurso existente y modificar

los datos de un recurso.

2. **Opciones de usuario:** en este panel se puede ver la imagen del usuario que está conectado al sistema, así como varios controles que permiten mandar información al sistema.
3. **Mapa:** es la representación gráfica del edificio de computación y los recursos que están localizados en las ambas plantas. Según el modelo topológico diseñado el departamento de Computación cuenta con dos plantas (ver figura 5.13) la plana baja está representada a la izquierda mientras que la planta alta se puede ver en el lado derecho.

### 5.5.2 Visualización de mapas

Los mapas que se visualizan en la imagen 5.13 constituyen la parte más importante del sistema NOTU. Estos mapas son interactivos y ofrecen varias funcionalidades, las cuales se describen a continuación:

- a) **Zoom:** esta funcionalidad se utiliza para agrandar una sección del mapa. Hay dos maneras de usarlo:
  1. **técnica *zooming*:** implica dar doble clic en cualquier punto del mapa, lo cual hará que el mapa se acerque a esa región, dando una sensación de movimiento.
  2. **panel de *zoom*:** este panel tiene seis niveles. Al hacer clic en el signo + el mapa se acercará tomando como referencia el centro de la vista del mapa; por el contrario al hacer clic en el signo - el mapa se alejará. También se puede hacer clic en los rectángulos que están entre los signos + y - . Estos rectángulos representan cada nivel de *zoom*, así el usuario puede ajustar el nivel de *zoom* que más se adapte a sus necesidades.
- b) **Navegación:** esta funcionalidad es usada cuando el usuario quiere mover el mapa. Hay dos formas de hacerlo:
  1. **técnica *panning*:** requiere hacer clic y arrastrar el ratón. Al mover el ratón el mapa también se moverá en la misma dirección que el ratón.
  2. **panel de navegación:** este panel tiene cuatro flechas (abajo, arriba, izquierda y derecha). Al hacer clic en alguna de estas flechas, el mapa se moverá en el mismo sentido.
- c) **Capas:** el mapa está compuesto de varias capas, las cuales organizan la información del mapa, e.g., la capa denominada **Planta baja** muestra u oculta la planta baja





Figura 5.14: Control que muestra el estado de las capas

del edificio. Al hacer clic en el primer signo + del lado derecho del mapa, se despliega un menú con todas las capas disponibles. Al hacer clic en el recuadro, la capa se vuelve visible, en tanto que al desactivarlo la capa se oculta.

El mapa cuenta con las siguientes capas (ver figura 5.14):

1. **Punto inicial:** contiene el punto donde iniciará la búsqueda (solo es visible en el módulo de búsqueda),
2. **Recursos:** indica los recursos que cumplen con los requerimientos del usuario,
3. **Etiquetas alta:** señala los nombres de las celdas de la planta alta del edificio,
4. **Puntos alta:** expone los puntos centrales y de transición de las celdas de la planta alta del edificio,
5. **Puntos baja:** contiene los puntos centrales y de transición de las celdas de la planta baja del edificio,
6. **Caminos alta y caminos baja:** representa las capas que muestran los caminos de sus respectivas plantas,
7. **Planta alta y baja:** muestra u oculta las plantas alta o baja respectivamente,
8. **Ver celdas:** contiene todas las celdas.

- d) **Minimapa:** el sistema cuenta con un minimapa (ver figura 5.15) para ayudar al usuario a saber en todo momento en dónde se encuentra. El minimapa muestra un mapa en miniatura que no cambia cuando el usuario hace *zoom*; también tiene un recuadro que muestra la ubicación de la vista del mapa. Si el usuario mueve el mapa, el recuadro del minimapa también se moverá. Para visualizar el minimapa se debe hacer clic en el signo + que se encuentra en la esquina inferior derecha del mapa.

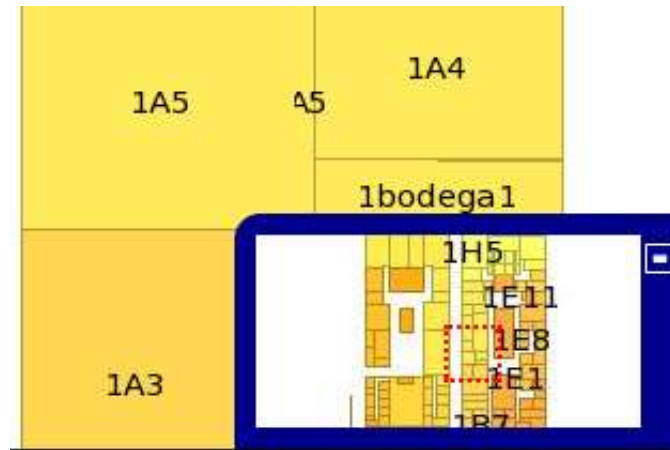


Figura 5.15: Control minimapa

En la figura 5.16 se muestra el mapa con todas sus opciones y todas las capas activadas.

### 5.5.3 Publicación de un recurso

Un usuario puede publicar un recurso para que sus colegas puedan verlo. Para tener una mejor organización en la administración de dichos recursos se definieron veintidós categorías que clasifican el funcionamiento de los recursos informáticos más importantes que se manejan en el Departamento de Computación. Dichas categorías se definen en la tabla 5.1 donde la columna de la izquierda representa la imagen que simboliza la categoría y la columna de la derecha especifica una descripción.

Estas categorías engloban los recursos informáticos que maneja el Departamento de Computación. Además de elegir la categoría a la que el recurso pertenece, el usuario debe asignar un nombre y una descripción. El nombre es un identificador para distinguirlo de los otros recursos en la misma categoría, e.g., el nombre de una impresora podría ser: Impresora 1, Impresora canon o Impresora a color. La descripción del recurso es la explicación de forma detallada de las características del recurso. Retomando el ejemplo anterior, la impresora puede tener como descripción la marca, el modelo, las páginas impresas por

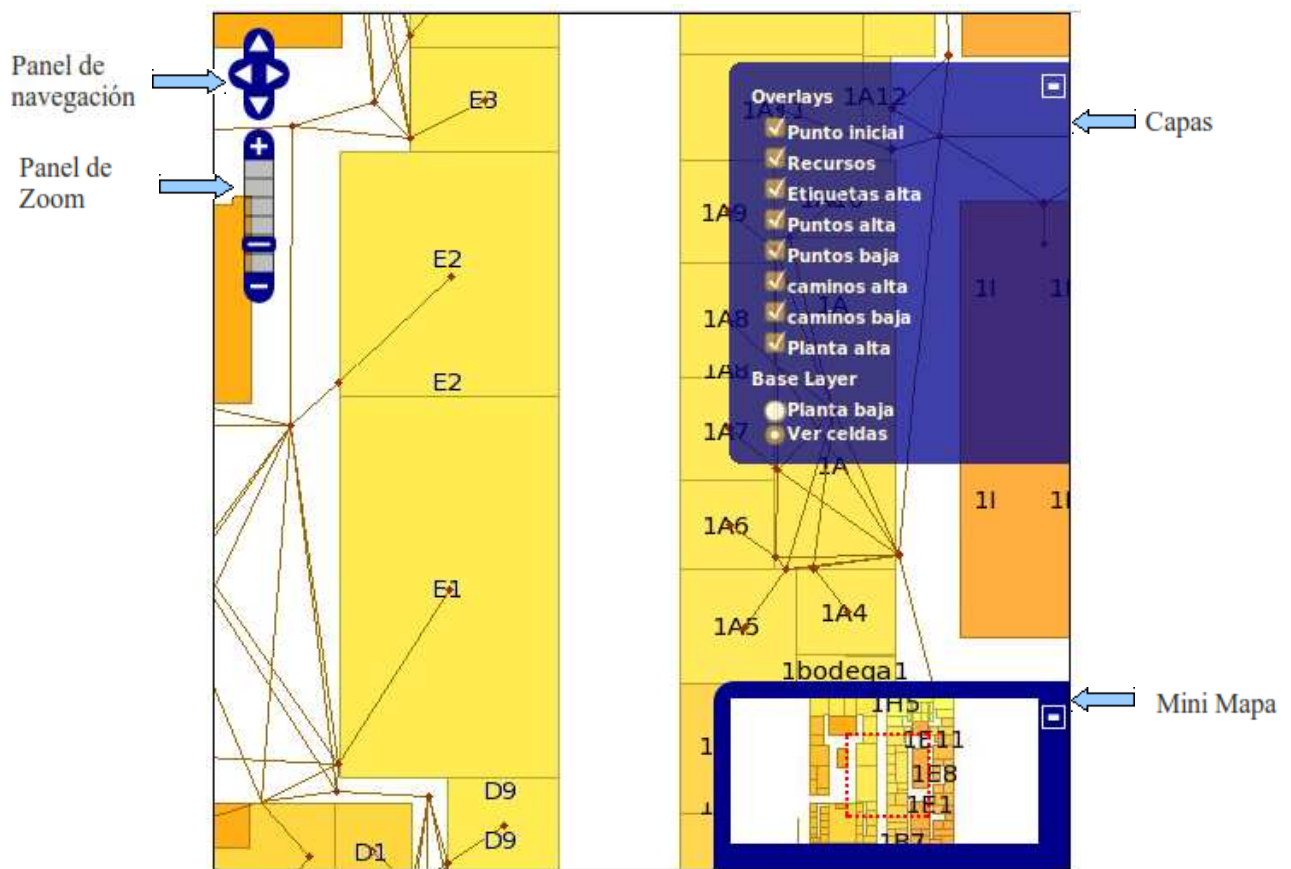


Figura 5.16: Controles que facilitan la interacción con el mapa

minuto, etc.

Para acceder a este módulo, se da clic en el menú de publicación de un recurso y como resultado aparece la pantalla de la figura 5.17. En el panel de la derecha se puede ver las opciones que tiene un usuario en este módulo. El usuario puede navegar en el mapa o puede colocar un nuevo recurso en su ubicación correspondiente. La opción **Navegar** permite al usuario interactuar con el mapa, usando tanto el ratón como los paneles. Para publicar un recurso se debe seleccionar la opción **colocar nuevo recurso**, la cual permite al usuario definir la ubicación del recurso mediante un clic en el mapa.

### Pasos para publicar un recurso

A continuación se enlistan los pasos para publicar un recurso:

1. seleccionar la opción **colocar nuevo recurso**,



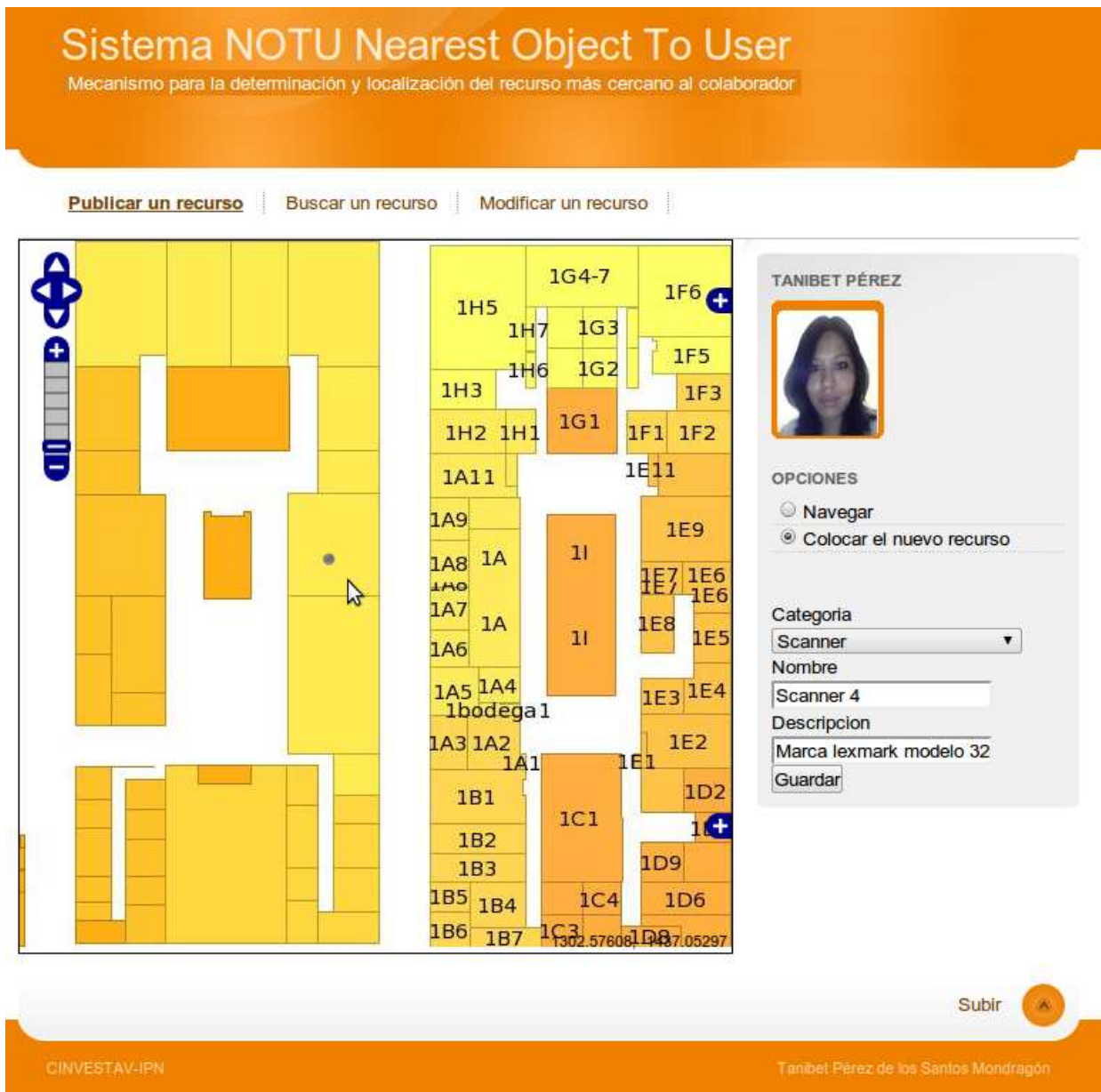


Figura 5.17: Pantalla de publicación de un recurso

2. dar clic en el lugar donde está físicamente colocado el recurso,
3. en respuesta, el sistema NOTU detecta el punto y coloca un punto gris,
4. elegir la categoría del recurso mediante un clic,

5. escribir el nombre del recurso,
6. escribir una breve descripción del recurso, y
7. dar clic en el botón **Guardar**.

#### 5.5.4 Modificación de un recurso

Este módulo fue diseñado para correcciones o modificaciones de la información referente a un recurso. Hay dos casos en los que un usuario modifica la información de un recurso. El primero corresponde a un cambio físico en la ubicación del recurso, e.g., cuando un recurso es cambiado de oficina. El segundo caso de modificación implica un cambio (o actualización) de la información correspondiente a un recurso, e.g., si un recurso ya no es disponible o si es reemplazado por otro. Para modificar un recurso se debe ingresar al módulo de modificación de recurso y dar clic en la opción **Modificar un recurso** en el menú del sistema NOTU, como resultado, se desplegará la pantalla que se muestra en la figura 5.18

##### Pasos para mostrar la información del recurso

En la pantalla de la figura 5.18 el usuario tiene tres opciones que se muestran del lado derecho. La opción **Navegar** permite la interacción del usuario con el mapa. La opción **seleccionar un recurso** permite que el usuario pueda obtener más información del recurso. Para desplegar la información de un recurso es necesario realizar los siguientes pasos:

1. dar clic en la opción **seleccionar un recurso**,
2. dar clic en el recurso que se desea modificar, y
3. en respuesta, el sistema NOTU despliega un cuadro de texto que muestra el nombre de recurso y su descripción.

En la figura 5.19 se muestra el cuadro de texto que indica, con una flecha, el recurso que se está consultando y, además, dentro del cuadro de texto hay una imagen con forma de tache que sirve para cerrar el cuadro de texto.

Se puede hacer clic en todos los recursos disponibles para conocer su información. Cuando los recursos están muy juntos o incluso encimados, basta con hacer *zoom* en el mapa para que los recursos se alejen, permitiendo al usuario poder seleccionarlos individualmente.



Figura 5.18: Pantalla de modificación de un recurso

### Pasos para modificar un recurso

Para modificar un recurso, el usuario debe localizarlo en el mapa y realizar los siguientes pasos:



Figura 5.19: Cuadro de texto con información de un recurso

1. dar clic en la opción **Modificar un recurso**, en las opciones del usuario
2. seleccionar el recurso a modificar en el mapa
3. mover el ratón con el botón presionado para arrastrar el recurso hasta la nueva ubicación
4. soltar el botón cuando el recurso está colocado en la nueva ubicación
5. en ese momento aparece otro cuadro de texto que permite modificar la información del nombre, la descripción y la clasificación del recurso
6. dar clic en el botón **Guardar**, una vez modificada la información (ver figura 5.20)

Si la ubicación del recurso no es la deseada se puede repetir del paso 3 al 6.

### 5.5.5 Búsqueda de un recurso

Este módulo está diseñado para auxiliar a un usuario que requiere conocer la ubicación de algún recurso y cómo llegar a este. El sistema NOTU indica el recurso más cercano al colaborador mediante el trazo de líneas que ayudarán al usuario a llegar a él.

Para realizar la búsqueda de un recurso, el usuario debe ingresar módulo de búsqueda de un recurso, dar clic en la opción **Buscar un recurso** del menú del sistema NOTU, el cual mostrará la pantalla de la figura 5.21. Esta pantalla ofrece dos opciones en el panel del lado derecho:

1. **Navegar:** mediante esta opción el usuario puede interactuar libremente con el mapa,

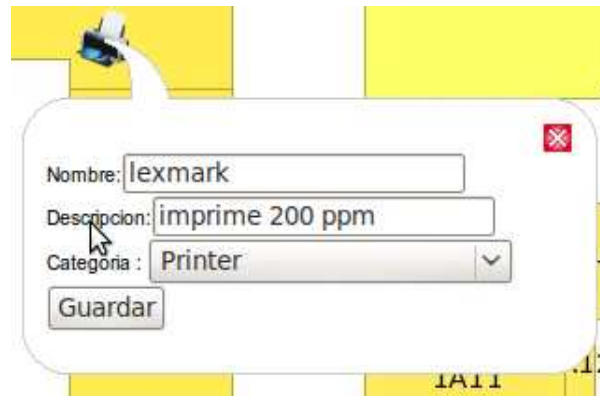


Figura 5.20: Cuadro de texto para modificar un recurso

2. **Colocar el punto de inicio:** esta opción sirve para colocar la ubicación actual del usuario; esta ubicación es necesaria para saber cual es el recurso que se encuentra más cerca del usuario.

En el panel de opciones también se encuentra una serie de imágenes, las cuales representan las categorías de los recursos. Para saber el nombre de cada categoría, el usuario puede poner el cursor sobre alguna imagen y como resultado se desplegará un texto de ayuda con el nombre de la categoría.

### Pasos para buscar un recurso

Para buscar un recurso se deben realizar los siguientes pasos:

1. dar clic en la opción **Colocar el punto de inicio**,
2. dar clic en el mapa para indicar la ubicación actual del usuario. El sistema colocará un círculo verde que se puede mover a cualquier otra ubicación,
3. seleccionar la imagen que representa el recurso a localizar (e.g., si el usuario desea encontrar una impresora debe dar clic en la imagen de la impresora)

En el mapa se visualiza el objeto más cercano al colaborador y se indica, con color rojo el trayecto para llegar a él. Si en el sistema se encuentran más recursos, estos se desplegarán también pero sólo un recurso es el más cercano al usuario y sólo se trazará el trayecto para ese recurso.

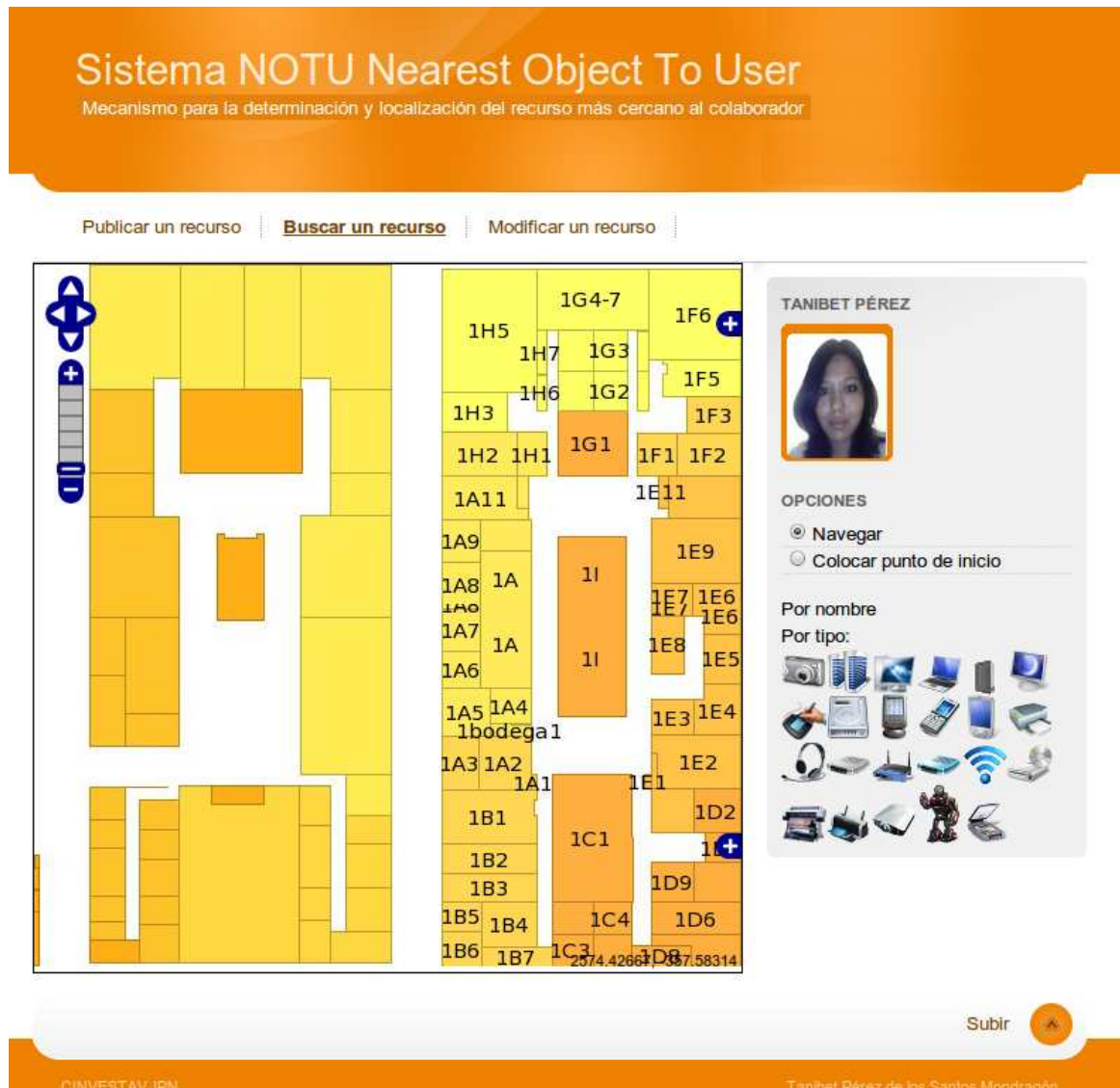


Figura 5.21: Pantalla de búsqueda de un recurso

Un aspecto muy importante que se debe mencionar es que el sistema NOTU distingue a los recursos que están en el primer piso de los que están en el segundo, mostrando únicamente la ruta más corta. Esto se logra mediante el modelado de la topología de distancias de recorrido (ver Capítulo 3 sección 3). Por cada piso se crea una topología diferente lo cual hace posible la distinción de recursos en diferentes pisos.



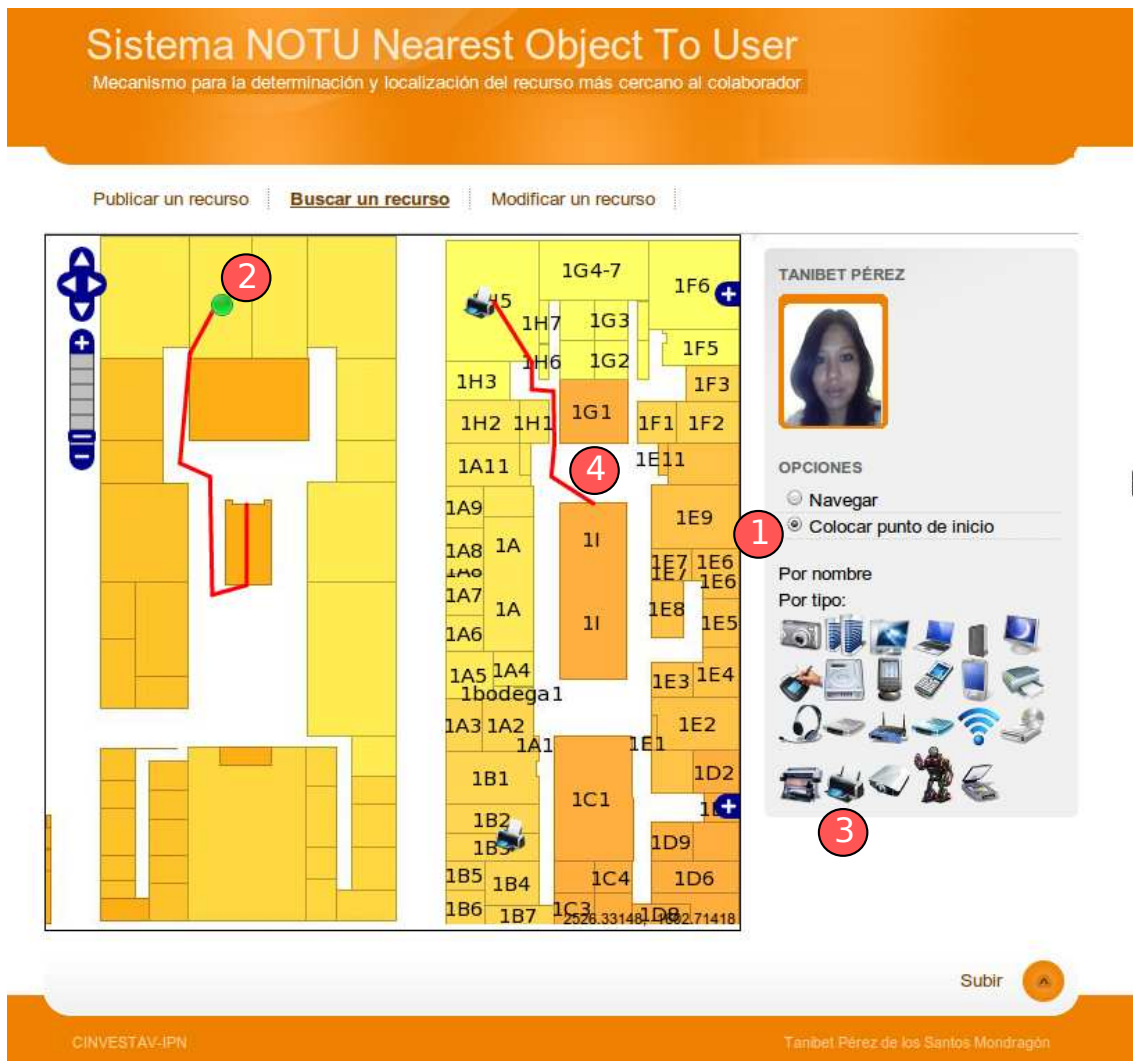


Figura 5.22: Resultado de la búsqueda del recurso más cercano

## 5.6 Pruebas

En esta sección se realizarán las pruebas del sistema con base en el caso de estudio que se describe a continuación. Para corroborar que el sistema funciona adecuadamente se plantea un escenario en el que se pone a prueba la distinción de recursos en varios pisos y el hecho de que los usuarios no atraviesen muros.

### 5.6.1 Caso de estudio

Un edificio cuenta con varios recursos que están distribuidos en todas las oficinas por lo que los usuarios necesitan compartir los recursos. El primer problema surge cuando el usuario desconoce la existencia de algún recurso. Para que un usuario utilice un recurso debe conocer la ubicación del recurso más cercano y la disponibilidad del mismo. Cuando el edificio es de varios pisos tal vez también se requiera una guía para conocer el camino más corto para llegar a él.

El usuario Tanibet Pérez se encuentra en el salón H2 dentro del edificio del Departamento de Computación en la planta alta, necesita imprimir una lámina para su exposición acerca de las nuevas tecnologías TABLET, el usuario Tanibet Pérez desconoce la existencia de un PLOTTER por lo que es necesario consultar con sus colegas para investigar si existe algún PLOTTER en el edificio, después de unos minutos se entera que hay un PLOTTER en la planta baja. Se dirige al sitio para corroborar que realmente se encuentre ahí.

Posteriormente llega al salón D5 y efectivamente existe un PLOTTER pero no está disponible ya que hay varios trabajos de impresión, pregunta al encargado si hay otro PLOTTER disponible la respuesta es positiva y le indican que existe otro PLOTTER en el salón h4 en la planta baja. Por lo que Tanibet debe volver a la planta baja a imprimir su trabajo.

### 5.6.2 Prueba del sistema NOTU

El usuario Tanibet Pérez utiliza el sistema NOTU para realizar la búsqueda de un PLOTTER como respuesta el sistema muestra la pantalla que se visualiza en la figura 5.23. Cuando el sistema NOTU localiza uno o varios recursos disponibles los compara para localizar el más cercano y genera la ruta más corta a él, de lo contrario el sistema NOTU muestra una notificación de que no hay recursos disponibles.

En la figura 5.23 se puede ver una notificación de la distancia en metros que un usuario debe recorrer para llegar al recurso, en este caso el usuario debe recorrer 61.6 metros.

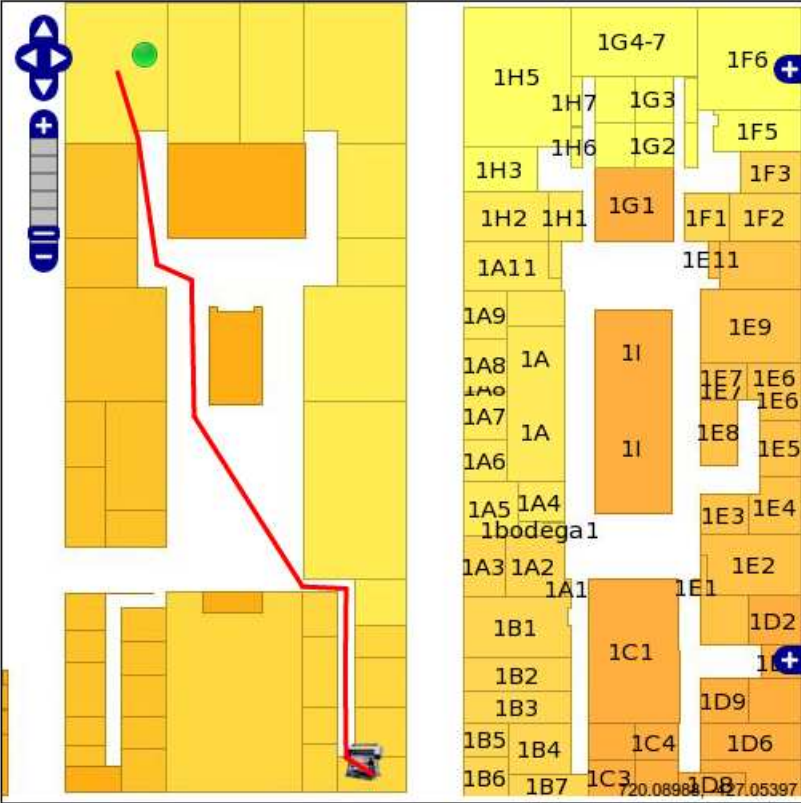
En el sistema NOTU se ha publicado un nuevo PLOTTER ahora en la planta alta. Si el usuario realiza una nueva búsqueda, obtiene la pantalla de la figura 5.24. En la notificación que aparece en dicha figura se puede observar que la distancia para llegar al PLOTTER es menor que la anterior (52.8 metros), por lo que el sistema dibuja la ruta al recurso más cercano aún si el recurso se encuentra en otra planta.




## Sistema NOTU Nearest Object To User

Mecanismo para la determinación y localización del recurso más cercano al colaborador

Publicar un recurso
Buscar un recurso
Modificar un recurso



**TANIBET PÉREZ**




**OPCIONES**


- Navegar
- Colocar punto de inicio

Por nombre

Por tipo:



**Distancia de recorrido:**  
**61.65357 m**

Subir 

CINVESTAV-IPN
Tanibet Pérez de los Santos Mondragón

Figura 5.23: Resultado 1 de la búsqueda del recurso más cercano

Por último, como se ve en las figuras 5.23 y 5.24, el sistema genera la ruta más corta con base en el grafo definido anteriormente, por lo que un usuario puede seguir la ruta sin temor a toparse con un muro.

## Sistema NOTU Nearest Object To User

Mecanismo para la determinación y localización del recurso más cercano al colaborador

Publicar un recurso
Buscar un recurso
Modificar un recurso

1H5	1G4-7	1F6
1H7	1G3	1F5
1H6	1G2	1F3
1H3	1G1	1F1
1H2	1H1	1F2
1A11		1E11
1A9		
1A8	1A	1I
1A7	1A	1E7
1A6		1E8
1A5	1A4	1E3
1A4	1bodega1	1E4
1A3	1A2	1E2
1A1		1E1
1B1	1C1	1D2
1B2		1D1
1B3		
1B5	1B4	1C4
1B6	1B7	1C3
		1D8

**TANIBET PÉREZ**

**OPCIONES**

- Navegar
- Colocar punto de inicio

Por nombre

Por tipo:

**Distancia de recorrido:**  
52.82902 m

Subir

CINVESTAV-IPN
Tanibet Pérez de los Santos Mondragón

Figura 5.24: Resultado 2 de la búsqueda del recurso más cercano























Imagen	Descripción
	<b>Cámara:</b> dispositivo de cámara fotográfica y de vídeo
	<b>Cluster:</b> conjunto de computadoras que actúan como una sola
	<b>PC:</b> computadora de escritorio
	<b>Laptop:</b> computadora portátil
	<b>Servidor:</b> computadora que provee servicios a clientes
	<b>Firma electrónica:</b> dispositivo para capturar la firma de un usuario
	<b>Almacenamiento externo:</b> dispositivo para almacenar información
	<b>PDA:</b> computadora portátil con reconocimiento de escritura
	<b>Celular inteligente:</b> celular de última generación con WiFi
	<b>Tablet PC:</b> computadora portátil con pantalla táctil
	<b>Multifuncional:</b> dispositivo que permite imprimir, copiar y escanear
	<b>Hardware multimedia:</b> reproductor de contenido multimedia
	<b>Hub:</b> dispositivo para compartir una red de datos
	<b>Router:</b> dispositivo que permite el enrutamiento de paquetes
	<b>Switch:</b> dispositivo de interconexión de segmentos de redes
	<b>Base WiFi:</b> punto de acceso a una red WiFi
	<b>Dispositivo CD:</b> dispositivo lector-escritor de CD o DVD
	<b>Plotter:</b> dispositivo para impresiones de grandes dimensiones
	<b>Impresora:</b> dispositivo para imprimir documentos
	<b>Cañón:</b> dispositivo óptico para proyectar imágenes o texto
	<b>Robot:</b> entidad virtual, mecánica o artificial
	<b>Escáner:</b> dispositivo óptico para introducir imágenes o textos

Tabla 5.1: Categorías de recursos



# Capítulo 6

## Conclusiones y trabajo a futuro

En este capítulo se resumen todas las actividades y los resultados obtenidos en el proceso de implementación del sistema NOTU. En la sección 6.1 se describen brevemente todas las actividades que se realizaron con las respectivas problemáticas o dificultades que se fueron presentando. En la sección 6.2 se describe el alcance del sistema y las actividades complementarias a él.

### 6.1 Conclusiones

Es importante mencionar que la importancia de este trabajo de tesis radica en el mecanismo para la determinación y la localización del recurso más cercano a la posición actual del usuario. Para mostrar este mecanismo fue necesaria la implementación de un Sistema de Información Geográfica para poder visualizar cómo funciona de dicho mecanismo.

Un Sistema de Información Geográfica facilita la visualización y la representación de datos espaciales por lo que se utilizó para la representación de las oficinas, salones, caminos y recursos del edificio del Departamento de Computación.

El proceso de elaborar un Sistema de Información Geográfica es complicado, además de que se debe conocer las herramientas adecuadas para poder implementar dicho sistema.

El primer paso de la creación del sistema fue la obtención de los planos del edificio del Departamento de Computación ya que con base en ellos se realizó un proceso de digitalización. La digitalización es la transformación de la información desde un formato analógico, como un mapa en papel, a un formato digital, de forma que pueda ser almacenada y visualizada en la computadora.

El procedimiento de digitalización fue realizado de forma “manual” lo cual implica traducir todas las entidades del mapa a puntos, líneas o polígonos usando el ratón. La dificultad

de este proceso radica en que al dibujar las entidades del mapa se debe manejar un nivel de *zoom* adecuado ya que las líneas o los polígonos no se veían totalmente alineados, e.g., en las primeras fases de la digitalización al aplicar *zoom* en el mapa se observaba que las líneas que componen los caminos se separan, dando como resultado caminos independientes.

Una vez realizado este proceso se llevó a cabo un estudio de las tecnologías que ofrecen soporte para la creación de Sistemas de Información Geográfica, se buscó una tecnología que permitiera implementar un sistema distribuido así como modificar y actualizar la información espacial en tiempo de ejecución. Por lo que se instalaron y probaron varias tecnologías para elegir la mejor.

Un requerimiento muy importante del sistema NOTU es que debe ser distribuido, por tal motivo se implementó un sistema Web que administrara los mapas de una forma distribuida; este sistema también es conocido como *Web Map Server (WMS)*

Para administrar la información de la ubicación de los recursos, cuartos, oficinas y de todas las entidades que componen el edificio del Departamento de Computación fue necesario implementar una base de datos con soporte espacial. Por tal motivo, se instalaron algunos sistemas gestores de base de datos para determinar cual era el que ofrecía una mejor comunicación con el WMS.

Otro requerimiento del sistema NOTU es que se pueda visualizar en dispositivos móviles. Se realizó una investigación de tecnologías que proporcionarían un formato de salida compatible con dichos dispositivos y se descubrió que el servidor de mapas debe actuar como una interfaz. Dicha interfaz obtiene como entrada los datos almacenados en la base de datos espacial y como salida muestra los datos en forma de mapas con un formato de imagen PNG.

Todas estas herramientas se instalaron y configuraron en el sistema operativo LINUX. El proceso de configuración e instalación fue complicado ya que cada herramienta tiene su propio proceso de instalación y configuración.

A continuación se describen los mecanismos, técnicas y herramientas para la implementación del mecanismo para la determinación y la localización del recurso más cercano.

Para representar los caminos y los recursos que se encuentran dentro del edificio del Departamento de Computación fue necesario diseñar e implementar un grafo dirigido el cual abstrae la información del mundo real para poder manejarla en una computadora. Dicho grafo se compone de aristas y nodos; los nodos representan tanto los puntos centrales de un salón u oficina como los puntos de transición; las aristas simbolizan los caminos que un usuario debe seguir para trasladarse de un lugar a otro.

El sistema NOTU es capaz de resolver el problema de la ruta más corta usando el algoritmo de Dijkstra en el grafo mencionado anteriormente. La serie de pasos se muestra a continuación:

- el usuario especifica su ubicación en el mapa,
- el usuario elige el tipo de recurso que desea encontrar,
- el sistema realiza una búsqueda de los recursos que satisfacen las necesidades del usuario,
- si el sistema encuentra uno o varios recursos calcula con el algoritmo de Dijkstra la ruta más corta para elegir el más cercano,
- el sistema muestra en pantalla al recurso más cercano y la ruta para llegar a él.

El usuario puede especificar su ubicación usando la interfaz. Para elegir el tipo de recurso se estableció una clasificación de los recursos informáticos que más se usan en el Departamento de Computación. Se crearon 22 clasificaciones de los recursos dependiendo de su funcionalidad, lo cual facilita al usuario la creación o búsqueda de un recurso.

Una vez que el usuario especifica el recurso que desea buscar, el sistema NOTU realiza una consulta a la base de datos para obtener la información del recurso o los recursos que satisfacen la petición del usuario. Hay tres resultados a esta consulta: 1) no encuentra recursos, 2) encuentra un recurso o 3) encuentra dos o más recursos.

Cuando la consulta no regresa información significa que no hay ningún recurso que satisfice las necesidades del usuario o que el recurso no está disponible. Si la consulta regresa un recurso entonces el sistema NOTU busca la ruta más corta y despliega el recurso y la ruta para llegar a él, cuando la consulta regresa dos o más recursos se debe hacer una comparación de las distancias mínimas de la ubicación del usuario a cada recurso para determinar cuál es el más cercano.

Posteriormente, el sistema NOTU construye un archivo XML como respuesta a la petición; dicho archivo contiene la información de la ubicación del recurso más cercano y de la ruta para llegar a él. El navegador recibe el archivo y usando la herramienta *Openlayers* se puede traducir en mapas que el usuario puede interpretar más fácilmente.

El proceso de calcular la ruta más corta se puede llevar a cabo cuando se tiene un punto origen y uno o más puntos destinos, en este caso el punto origen es la ubicación del usuario y el punto destino es la ubicación de los recursos que satisfacen la necesidad del usuario. El algoritmo de Dijkstra fue implementado usando una herramienta para crear

una topología en la base de datos espacial. Para crear una topología se debe agregar a la tabla del grafo los atributos origen y destino para cada arista. El parámetro para determinar la ruta más corta es la distancia de cada arista con base en la cual se puede construir el camino con la distancia mínima.

Se diseñó un modelo topológico del edificio del Departamento de Computación porque facilita la representación de los datos reales, pero también es necesario para distinguir los recursos cuando se encuentran en diferentes pisos. El modelo topológico está constituido por dos grafos los cuales representan a la planta alta y la planta baja respectivamente. Estos grafos están unidos por una arista que representa la escalera, así el sistema puede distinguir a los recursos que se encuentran en diferentes pisos pero a la misma distancia euclidiana del usuario.

El sistema NOTU permite publicar nuevos recursos. Para que un usuario publique un recurso debe introducir la información referente al recurso (nombre, clasificación y marca) y también debe especificar la ubicación del recurso dentro del mapa. Un usuario puede modificar la información o la ubicación de un recurso mediante el módulo de modificación de recursos.

El sistema NOTU fue integrado al sistema RAMS usando una interfaz debido a que el sistema RAMS está programado en JAVA y es de escritorio, en tanto que el sistema NOTU es una aplicación Web programada en PHP. Para la intercomunicación entre el sistema NOTU y el sistema RAMS se implementó una interfaz XML. El sistema RAMS consulta su ontología para consultar los datos referentes a los recursos y envía al sistema NOTU una respuesta mediante un archivo XML usando JSP para publicar los datos en la Web.

## 6.2 Trabajo a futuro

Actualmente el sistema NOTU es una parte del sistema RAMS el cual detecta a los usuarios usando reconocimiento de caras, sin embargo en el sistema NOTU el usuario debe indicar su ubicación de manera manual. El sistema NOTU puede mejorar su funcionamiento si el usuario es detectado de forma automática para lo cual es necesario implementar un servidor de ubicaciones que pueda detectar a un usuario y así determinar su ubicación.

El sistema NOTU es una aplicación para mostrar el funcionamiento del mecanismo para la determinación y la localización del recurso más cercano al usuario. Sin embargo el proceso para aplicar este mecanismo para en otro edificio es complicado ya que se debe diseñar un nuevo modelo topológico del edificio en donde se desea aplicar. Para poder



facilitar esta situación se puede implementar una herramienta que facilite el diseño de dicho modelo y así hacer más general el proceso de aplicación del mecanismo.

Al aplicar este mecanismo en diferentes áreas o lugares se pueden optimizar los diferentes procesos, e.g., rutas de distribución, aeropuertos, etc. En el caso de los aeropuertos no existe algún mecanismo para poder calcular la ruta más corta del equipaje de los usuarios. También se puede aplicar este mecanismo para optimizar las rutas de distribución de algún producto.

También es aplicable en la visita de algunos lugares de interés como museos o centros comerciales. Al aplicar el mecanismo de determinación y localización del recurso más cercano un usuario puede saber cómo llegar a alguna sala de un museo o a algún puesto dentro de un centro comercial.

Este mecanismo puede aplicarse también dentro de hospitales donde el tiempo es un factor de vital importancia así como conocer la disponibilidad de algún recurso como por ejemplo instrumentos quirúrgicos o quirófanos puede reducir mucho los tiempos para los doctores.

En el mecanismo para la determinación y localización del recurso más cercano puede aplicarse en cualquier ambiente donde sea necesario optimizar rutas donde se involucren varios recursos distribuidos.



# Bibliografía

- [map, 2009] (2009). *Sistemas de localización e Información Geográfica*. Observatorio Regional de la Sociedad de la Información (ORSI), Junta de Castilla y León.
- [Azizyan et al., 2009] Azizyan, M., IonutConst, anche, and Choudhury, R. R. (2009). Surroundsense: Mobile phone localization via ambience fingerprinting. *Communications of the ACM*.
- [Balakrishnan, 1997] Balakrishnan, V. (1997). *Shaum's outline of Theory and problems of Graph theory*. McGraw-Hill.
- [Borriello et al., 2005] Borriello, G., Chalmers, M., LaMarca, A., and Nixon, P. (2005). Delivering real-world ubiquitous location system. *Communications of the ACM*.
- [Cherkassky et al., 1993] Cherkassky, B. V., Goldber, A. V., and Radzik, T. (1993). Shortest paths algorithms:theory and experimental evaluation.
- [Cormen et al., 2002] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2002). *Introduction to algorithms*. McGraw-Hill.
- [Decouchant et al., 2010] Decouchant, D., Mendoza, S., and Rodríguez, J. (2010). *Emergent Web Intelligence Book: Advanced Semantic Technologies*, chapter PIÑAS - A Realistic Way to Support Distributed Web Cooperative Work, pages x-x (50 pages). Advanced Information and Knowledge Processing Series. Springer Verlag.
- [Deuker and Kjerne, 1989] Deuker, K. and Kjerne, D. (1989). Multipurpose cadastre terms and definitions. In *Proceedings of the American Society for Photography and Remote Sensing and Amercian Confress of Surveying an Mapping*.
- [Diestel, 2000] Diestel, R. (1997-2000). *Graph Theory*. Springer-Verlag.
- [Egenhofer et al., 1991] Egenhofer, M. J., Herring, J. R., Smith, T., and Park, K. K. (1991). A framework for the definition of topological relationships and algebraic approach to spatial reasoning within this framework. Technical Report 91-7, National Center for Geographic Information and Analysis.

- [Ellis et al., 1991] Ellis, C., Gibbs, S., and Rein, G. (1991). Groupware: Some issues and experiences. *Communications of the ACM*.
- [Ender, 2009] Ender, G. (2009). The revolutionary global dimension of collaborative excellence and sustainable development in economy. *Society, Politics, Education and Research, Open space Online eBook*.
- [García, 2009] García, E. K. G. (2009). *Disponibilidad de recursos compartidos en un ambiente colaborativo con carácter ubicuo*. PhD thesis, Centro de Investigación de Estudios Avanzados del Instituto Politécnico Nacional.
- [Greif, 1988] Greif, I. (1988). *Computer-Supported Cooperative Work: A Book of Readings*. Morgan Kaufmann.
- [Gross et al., 2004] Gross, I., Jonathan, L., and Yellen, J. (2004). *Handbook of graph theory*. CRC Press.
- [Gurevich, 2006] Gurevich, M. (2006). Jamspace: Designing a collaborative networked music space for novices. In *Proc. New Interfaces for Musical Expression (NIME 06) Conf.*, pages 118–123, Paris(France).
- [Harter et al., 1999] Harter, A., Hopper, A., P.Steggles, Ward, A., and Webster, P. (1999). The anatomy of a context-aware application. In *Proc. 5th ACM MOBICOM Conf.*, Seattle, WA.
- [Hazas et al., 2004] Hazas, M., Scott, J., and Krumm, J. (2004). Location-aware computing comes of age. *IEEE Computer*.
- [Hopper et al., 1992] Hopper, R. W. A., Falcao, V., and Gibbson, J. (1992). The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102.
- [Kay, 2007] Kay, T. (2007). Alphamail is scalable and accessible web mail. *Linux Journal, Specialized Systems Consultants, Inc.*, (160):11.
- [Kjaergaard et al., 2008] Kjaergaard, M. B., Treu, G., Ruppel, P., and Küpper, A. (2008). Efficient indoor proximity and separation detection for location fingerprinting. *Communications of the ACM*.
- [Krumm and Hinkley, 2004] Krumm, J. and Hinkley, K. (2004). The nearme wireless proximity server. In *Proc. of the Intl Conf. on Ubiquitous Computing*.
- [Maqsood and Javed, 2007] Maqsood, M. and Javed, T. (2007). Practicum in software project management: An endeavor to effective and pragmatic software project management education. In *Proc. 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering*, pages 471–480, Dubrovnik (Croatia). ACM Press.

- [Pashtan, 2005] Pashtan, A. (2005). *Mobile Web Services*, chapter User mobility and location management. Cambridge University Press.
- [Priyantha et al., 2001] Priyantha, N. B., Miu, A. K. L., Balakrishnan, H., and Teller, S. (2001). The cricket compass for context-aware mobile applications. In *Proc. of the 6th ACM MOBICOM Conf*, Rome, Italy.
- [Saha and Mukherje, 2003] Saha, D. and Mukherje, A. (2003). Pervasive computing: A paradigm for the 21st century. *IEEE Computer Society*.
- [Snekkens, 2001] Snekkens, E. (2001). Concepts for personal location privacy policies. *ACM Conference on Electronic Commerce*, pages 14–17. Tampa, Florida, USA.
- [Vallidis, 2002] Vallidis, N. M. (2002). *Whisper: A Spread Spectrum Approach to Occlusion in Acoustic Tracking*. PhD thesis, University of North Carolina at Chapel Hill.
- [Wallis, 2007] Wallis, W. (2007). *A Beginner's Guide to Graph Theory*. Birkhäuser Boston.
- [Weiser, 1993] Weiser, M. (1993). Some computer science problems in ubiquitous computing. *Communications of the ACM*.
- [Welch and Bishop, 1997] Welch, G. and Bishop, G. (1997). Scaat: Incremental tracking with incomplete information. *Computer Graphics 31, Annual Conference Series*.