

Desarrollo de aplicación utilizando el índice espacio-temporal I+3 R-Tree

Marcela Aballay¹, Anabella De Battista², Olinda Galiardi³

¹*Maestría en Ing. en Sist. de Información, Univ. Tecnológica Nacional, FRC, Córdoba, Argentina
marcelaaballay@yahoo.com.ar*

²*Dpto. de Sist. de Información, Univ. Tecnológica Nacional, FRCU, Entre Ríos, Argentina
debattistaa@frcu.utn.edu.ar*

³*Dpto. de Informática, Universidad Nacional de San Luis, Argentina
oli@unsl.edu.ar*

Abstract

Este artículo describe el desarrollo de un prototipo para la atención de eventos de emergencia y la administración de los móviles correspondientes del Municipio de la localidad de Córdoba. Parte de esta solución requería la implementación de una estructura de indexación para objetos que modifican su posición en el tiempo, por lo que se decidió utilizar el índice espacio-temporal I+3 R-Tree. Con la implementación de este índice se resolvió el tema de la gestión de objetos móviles así como las consultas relacionadas a la necesidad de reproducir la trayectoria que realizó un móvil en la atención de un evento.

1. Introducción

La selección e implementación de un sistema de gestión de base de datos (SGBD) es parte fundamental en el desarrollo de software. Actualmente la mayoría de los SGBD trabajan con objetos que se pueden representar a través del modelo relacional compuesto básicamente de tablas, tuplas, campos y las relaciones establecidas entre ellos.

Si bien el modelo relacional es el más utilizado, no es el apropiado para otros tipos de sistemas donde la función principal es poder representar la forma y/o posición de objetos en el espacio y su evolución en relación al tiempo. Es para estos tipos de datos que se desarrollaron nuevos modelos como las bases de datos espaciales y espacio-temporales.

También existen sistemas donde se requiere mantener distintos estados de la base a través del tiempo. En este caso el modelo relacional no se puede aplicar ya que solo registra el estado actual de las entidades. Las bases de datos temporales permiten administrar los datos resguardando su estado pasado, actual e incluso futuro.

Si se pretende gestionar información espacio-temporal es imprescindible contar con índices específicos, en este caso se habla de Métodos de Acceso Espacio-Temporales. Estos índices evitan examinar todos los objetos al momento de responder consultas y resultan eficientes ante consultas que se realizan con alta frecuencia. La indexación se utiliza para optimizar la recuperación de datos almacenados en memoria secundaria y el tiempo de procesamiento de las consultas.

En este artículo se presenta un desarrollo cuyo objetivo es la implementación de una estructura del tipo espacio-temporal para objetos en movimiento. En este caso se eligió implementar el método de acceso espacio-temporal I+3 R-Tree que se adecúa a las necesidades de almacenamiento y consultas propuestas en la solución.

2. Marco teórico

2.1. Bases de datos espaciales

Las Bases de Datos Espaciales permiten procesar objetos con alguna referencia espacial, y que poseen normalmente una estructura compleja. Un dato espacial puede ser un punto, una polilínea o un polígono [1]. Con un punto podrían representarse automóviles, edificios, estaciones de trenes, etc.; las líneas se utilizan para modelar ríos, rutas, cableados, etc., y los polígonos se pueden utilizar para representar países o regiones. El principal ámbito de aplicación de las base de datos espaciales es el de los Sistemas de Información Geográfica (GIS).

La recuperación y actualización de los tipos de datos espaciales se basan no sólo en el valor de ciertos atributos, sino también en la ubicación espacial del objeto. Por ejemplo, nos podría interesar obtener los terrenos geográficamente adyacentes a uno dado, o encontrar todas las estaciones de servicio por las que pasa una ruta.

Este tipo de bases de datos proporciona la estructura capaz de soportar datos espaciales, además de proveer un lenguaje de consulta y métodos de acceso eficientes [2].

Consultas espaciales [3]

- *Exact match query*: dado un objeto encuentra todos los iguales en forma, posición y tamaño.
- *Point query*: encuentra los objetos que se superponen o cubren al punto consultado.
- *Window query o range query*: dado un rectángulo de dimensión d , encuentra los objetos que tienen al menos un punto en común con el rectángulo de consulta.
- *Intersection query, region query u overlap query*: dado un objeto devuelve los objetos de la base de datos que tienen al menos un punto en común con el objeto de consulta.
- *Enclosure query*: devuelve los objetos de la base de datos que encierran o contienen completamente al objeto de consulta.
- *Containment query*: devuelve los objetos de la base de datos que están encerrados por el objeto de consulta.
- *Adjacency query*: encuentra los objetos adyacentes al objeto de consulta.
- *Nearest-neighbor query*: devuelve los objetos de la base de datos que tienen la distancia mínima al objeto de consulta. En este caso la distancia se define como la distancia entre sus puntos más cercanos (Euclidiana o de Manhattan).
- *Spatial join*: dadas dos colecciones de objetos espaciales R y S y un predicado espacial P , encuentra todos los pares de objetos de $R \times S$ donde $P(R, S)$ es verdadero.

2.2. Bases de Datos Temporales

Incorporan al tiempo como una dimensión, por lo que permiten asociar tiempos a los datos almacenados [4]. Existen tres clases de bases de datos temporales, en función de la forma en que manejan el tiempo [5]:

- *de tiempo transaccional* donde el tiempo se registra de acuerdo al orden en que se procesan las transacciones,
- *de tiempo vigente* que almacenan el tiempo en que el hecho ocurrió en la realidad, que puede no coincidir con el momento de su registro y
- *bitemporales* que integran la dimensión transaccional y la dimensión vigente a través del versionado de los estados, es decir, cada estado se puede modificar para actualizar el conocimiento de la realidad pasada, presente o futura, pero esas modificaciones se realizan generando nuevas versiones de los mismos estados.

Es decir que el tiempo válido identifica el momento de tiempo durante el cual la información es cierta en el

mundo modelado y el tiempo transaccional indica el momento que los datos fueron incorporados a la base de datos [6].

Si se realiza una consulta teniendo en cuenta el tiempo vigente esta sería: encontrar todos los móviles activos para un tiempo determinado; en cambio si tomamos como parámetro el tiempo transaccional se consultaría: todos los móviles activos registrados en un tiempo determinado.

2.3. Base de datos espacio-temporales

Las Bases de Datos Espacio-Temporales están compuestas por objetos espaciales que cambian su posición o forma a lo largo del tiempo [7]. Este modelo permite administrar ambos componentes de un objeto al mismo tiempo, el espacial y el temporal.

Estas bases de datos poseen ciertas características [8]:

- Tienen una estructura compleja que no puede ser representada por una tupla. Un dato espacial, entre otros, puede ser un punto, una línea o un polígono.
- Tienen a ser muy grandes. Por ejemplo un mapa puede ocupar varios gigabytes de almacenamiento.
- No existe un álgebra espacial estándar, por lo tanto no hay un conjunto de operadores estandarizados.
- Se emplean operadores espaciales altamente dependientes de la aplicación en particular.
- Debido a la complejidad de los datos las operaciones espaciales implican un costo de procesamiento más alto que las operaciones relacionales.
- Por su dinamismo, requieren que la estructura de datos soporte frecuentes inserciones, eliminaciones y actualizaciones.
- Los operadores espaciales no cumplen con la propiedad de clausura. Por ejemplo, la intersección de polígonos no necesariamente da como resultado otro polígono.

Las principales consultas que se pueden realizar sobre una base espacio-temporal son [9]:

- *TimeSlice*: devuelve los objetos que se encuentran en una determinada área en un instante de tiempo dado.
- *Intervalo*: obtiene los objetos que se encuentran en una determinada área en un intervalo de tiempo.
- *Eventos*: se obtienen todos los eventos que sucedieron en una región en un instante dado.
- *Trayectoria*: recupera las posiciones espaciales en las que un objeto ha permanecido en un intervalo de tiempo dado.
- *Consultas de los k vecinos más cercanos*: dado un objeto, se devuelven los k objetos más cercanos al

objeto de entrada para un instante o intervalo de tiempo.

2.4. Índices Espacio-Temporales

El procesamiento eficiente de las consultas es fundamental para el desarrollo de aplicaciones, por lo que se debe contar también con métodos adecuados para la recuperación de datos que eviten examinar todos los objetos de la base en el momento de la consulta. Como respuesta a esto se implementan índices espacio-temporales que permiten seleccionar eficientemente solo aquellos objetos que son candidatos a formar parte de la respuesta.

Existe un método base de indexación espacial que implementan la mayoría de los STDBMS, como Oracle, SQL y Postgres que es el R-Tree, propuesto por Guttman en 1984 [10].

Dentro de las variantes del R-Tree se encuentran métodos que consideran el tiempo como otra dimensión [11], por ejemplo 3D R-Tree, I+3 R-Tree y 2+3 R-Tree. Por el contrario, en otros el tiempo se incorpora como información dentro de la estructura de los nodos, como el RT-Tree.

En este caso se propone la utilización del índice I+3 R-Tree [12] que nos permite manipular los puntos en movimiento, administra el tiempo como otra dimensión y responde a las consultas timeslice, intervalo, evento y trayectoria. Es una variante del 2+3 R-Tree, que se compone de dos R-Tree, uno de dos dimensiones para la información actual (objetos que no tienen definido su tiempo final de estadía en una posición determinada) y otro de tres dimensiones, dos para el tiempo y una espacial, para la información pasada (objetos que tienen tiempo inicial y final en una posición).

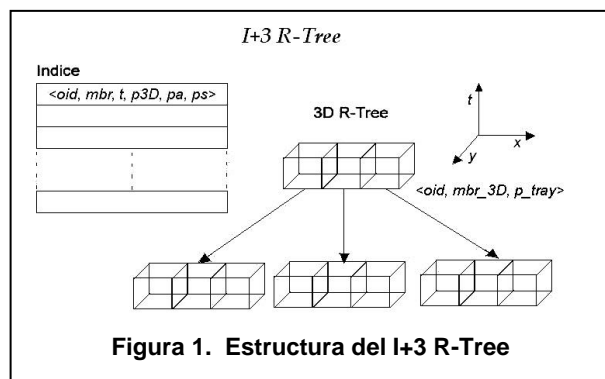


Figura 1. Estructura del I+3 R-Tree

La estructura del I+3 R-Tree (Figura 1) está formada por un R-Tree de tres dimensiones (3D R-Tree) más una estructura denominada I que guarda los datos de las posiciones actuales de los objetos.

3. Trabajo Realizado

En este trabajo se presenta el desarrollo de un prototipo de aplicación para la atención de eventos de emergencia y la administración de los móviles correspondientes del Municipio de la localidad de Córdoba.

El proceso consiste básicamente en recibir una llamada a través de los operadores telefónicos, quienes la registran y derivan al despacho (Figura 2).

Luego de registrada la llamada, ésta se puede derivar al despacho si requiere la intervención de algún móvil o finalizarla. En el despacho se cuenta con una vista de los eventos pendientes y en curso con su correspondiente localización en el mapa (Figura 3).

Para la asignación de un móvil a un evento pendiente se obtiene de la tabla de Móviles de la base de datos relacional aquellos que están en estado disponibles y se los ubica sobre el mapa, formando con ellos un diagrama de Voronoi (Figura 4).

De esta forma los despachantes pueden ubicar rápidamente cual es el móvil más cercano al evento que están atendiendo.

Luego de asignado el móvil, este se incorpora a la vista del despacho y se actualizan los eventos pendientes y en curso (Figura 5).

Una vez finalizada la atención se cierra el evento y se libera el móvil, pasando a estar disponible para cubrir un nuevo evento.

De los móviles se registra periódicamente su posición, lo que permite después reconstruir su trayectoria en la atención de un determinado evento.

4. 4. Elementos de Trabajo y Metodología

Para este desarrollo se tomaron como base los requerimientos del *call center* de emergencias de la Municipalidad de Córdoba. Dichos requerimientos fueron documentados con sus Actores y Casos de Uso (Figura 6).

El prototipo está programado en Java, utiliza una base de datos relacional de PostgreSQL e implementa el método de acceso espacio temporal I+3 R-Tree a través de sus librerías en Java.

Los principales objetos que se modelan a través de la base relacional son Llamadas, Eventos y Móviles (Figura 7).

El I+3 R-Tree está conformado por dos estructuras un 3D R-Tree y un Índice

Las nuplas en el 3D R-Tree están compuestas por <oid, mbr_3D, p_tray>

- oid: código identificador del objeto

- mbr_3D: región tridimensional cuya altura representa el intervalo temporal durante el cual el objeto se mantuvo en la posición espacial definida por su base
- p_tray: puntero al cubo anterior correspondiente al mismo oid, utilizado para mantener un historial de trayectoria.

Las tuplas en el índice son de la forma $\langle oid, mbr, t, p3D, pa, ps \rangle$

- mbr: región aproximada que ocupa actualmente el objeto
- t: tiempo de llegada del objeto a su ubicación actual
- p3D: puntero al cubo anterior correspondiente al mismo oid, utilizado para mantener un historial de trayectoria;
- pa: puntero al objeto insertado en el instante de tiempo inmediatamente anterior
- ps: puntero al objeto insertado en el instante de tiempo siguiente.

Esquemáticamente el I+3 R-Tree tiene el siguiente funcionamiento:

Para comenzar solamente se registran en el Índice los móviles con su posición inicial (el 3D R-Tree vacío) (Tabla 1).

Tabla 1. Índice

Oid	Mbr	t	p3D	Pa	Ps
om1	om1_posicion0	om1_tiempo0	null	Null	puntero a om2
om2	om2_posicion0	om2_tiempo0	null	puntero a om1	puntero a om3
om3

Al recibir información de una nueva posición del móvil *om1* se actualiza el Índice y el 3D R-Tree (Tablas 2 y 3).

Tabla 2. Índice

Oid	Mbr	t	p3D	Pa	Ps
om1	om1_posicion1	om1_tiempo1	puntero al primer cubo del R-Tree	Null	puntero a om2
om2	om2_posicion0 (la de la base)	om2_tiempo0	null	puntero a om1	puntero a om3
om3

Tabla 3. 3D R-Tree

Oid	mbr_3D	p_tray
om1	posicion0 - Desde tiempo0 Hasta tiempo1	null

Y así sucesivamente con cada recepción de información (Tablas 4 y 5).

Tabla 4. Índice

Oid	mbr	T	p3D	Pa	ps
om1	om1_posicion2	om1_tiempo2	puntero al segundo cubo del R-Tree	Null	puntero a om2
om2	om2_posicion0 (la de la base)	om2_tiempo0	null	puntero a om1	puntero a om3
om3

Tabla 5. 3D R-Tree

Oid	mbr_3D	p_tray
om1	posicion0 - Desde tiempo0 Hasta tiempo1	null
om1	Posicion1 - Desde tiempo1 Hasta tiempo2	puntero al cubo anterior

Para la visualización y georreferenciación se utilizaron las APIs de Google Map. De esta manera se puede localizar en el mapa de la ciudad los puntos donde se encuentran los eventos en curso y pendientes de atención.

En el caso del despacho, en la pantalla correspondiente a la asignación de un móvil para un evento, se implementó un diagrama de Voronoi. Esta estructura geométrica subdivide un área en regiones tomando como base una serie de puntos. Dichas regiones quedan formadas por los puntos más próximos a cada uno de los puntos de base [13]. Para este caso el área a dividir es el mapa de la ciudad de Córdoba y los puntos bases son los móviles disponibles. De esta forma se pueden identificar las distintas áreas de cobertura de cada móvil.

Como en el prototipo no se dispone de una conexión real con los móviles para que transmitan cada determinado período su posición, se realizó una simulación, registrando en el I+3 R-Tree distintos tiempos y posiciones (latitud, longitud) para algunos móviles.

Luego de finalizada la atención, en base a los datos registrados en la estructura espacio-temporal se puede resolver la consulta de trayectoria de un móvil, que es uno de los principales requerimientos de información para este sistema.

SISTEMA DE EMERGENCIAS 109 - Registro de Llamada Menu

Marcela Aballay -06/10/2016 11:35

Datos del Denunciante

Nombre:

Apellido:

Telefono:

Datos del Evento

Calle:

Nro:

Barrio:

Evento:

Detalle:

EMERGENCIA MEDICA VIA PUBLICA
 EMERGENCIA MEDICA DOMICILIO
 EMERGENCIA DEFENSA CIVIL

Diagnostico:

Figura 2. Pantalla para el Registro de Llamada

SISTEMA DE EMERGENCIAS 109 - Despacho de Eventos Menu

EVENTOS PENDIENTES

14	EMERGENCIA MEDICA VP	RAYMUNDO MONTENEGRO	1300	SAN MARTIN	<input type="button" value="Movil"/>
15	EMERGENCIA MEDICA VP	NICOLAS AVELLANEDA	800	CENTRO	<input type="button" value="Movil"/>

EVENTOS EN CURSO

Córdoba

Google
 Datos del mapa ©2015 Google
 Condiciones del servicio
 Informar un error en el mapa

Figura 3. Vista del despacho con los eventos pendientes



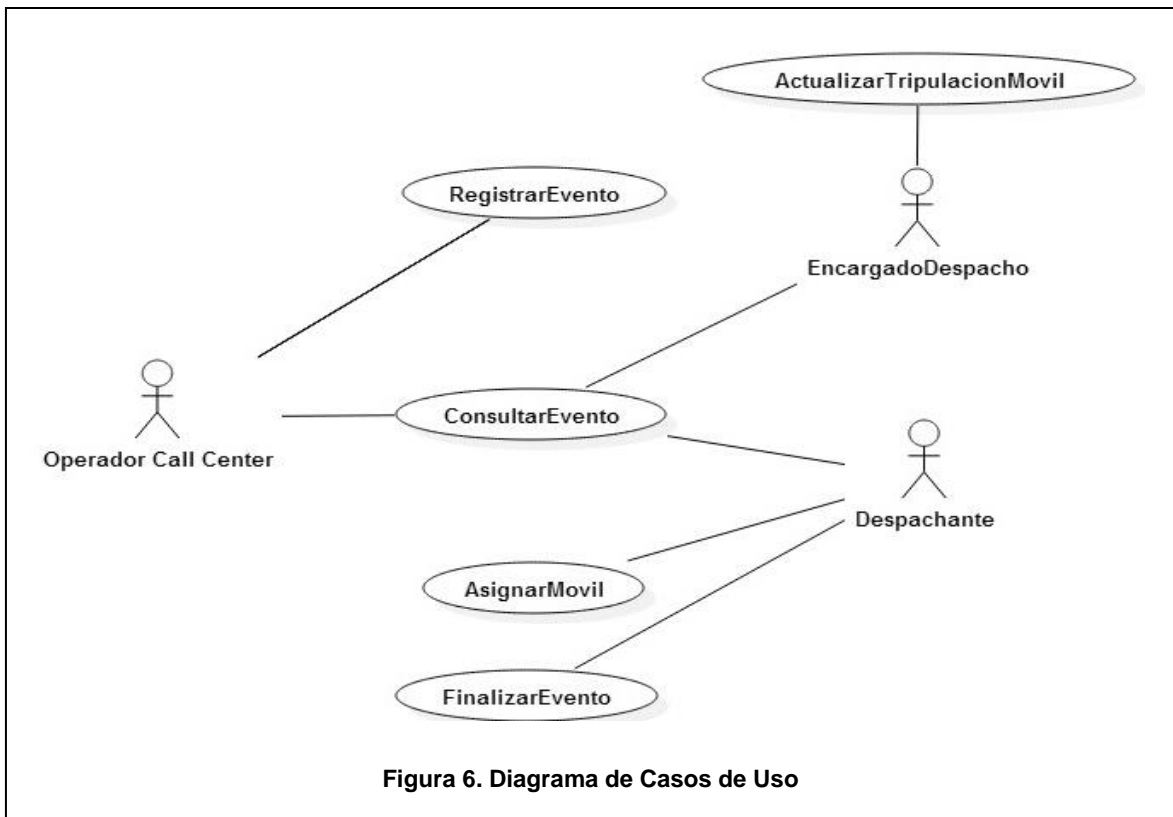


Figura 6. Diagrama de Casos de Uso

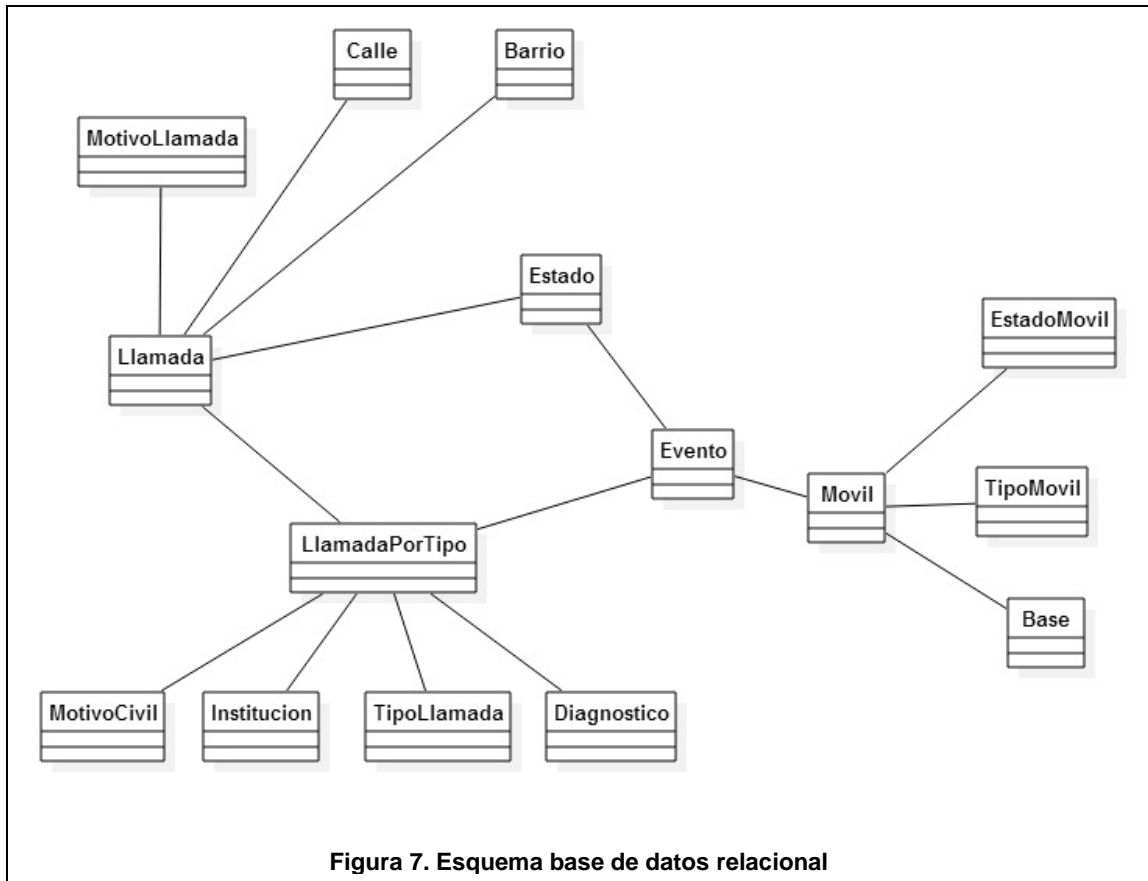


Figura 7. Esquema base de datos relacional

5. Discusión

Tratándose de un desarrollo con objetos en movimiento donde se deben considerar la componente espacial y temporal no se encontró en el mercado un sistema de gestión de base de datos que proporcione las facilidades de almacenamiento y consulta para tipos espacio-temporales.

Teniendo en cuenta las características de los móviles de este sistema se consideró la estructura del I+3 R-Tree como la más conveniente de implementar, teniendo en cuenta las siguientes consideraciones:

- Los objetos se mueven a gran velocidad.
- El tamaño y la forma de los objetos no son importantes, el interés se centra en su posición en el tiempo.
 - El espacio no cambia y la cantidad de objetos no tiene mucha variación.
 - Los objetos se mueven en áreas previamente establecidas.
 - Se necesita consultar la trayectoria realizada por los objetos.

Otro aspecto importante que se resolvió a través de la utilización de este índice es el de las consultas. En este caso se necesita reproducir la trayectoria que realizó un móvil en la atención de un evento, consulta que no está disponible en todos los índices de acceso espacio-temporal.

Además en el I+3 R-Tree la lista secuencial de los objetos se encuentra ordenada por su identificador lo que hace más eficiente la búsqueda de un determinado móvil.

En este trabajo también es importante la integración de ambos modelos, el relacional y el espacio-temporal. La utilización de PostgreSQL permite la administración y mantenimiento de la base relacional de forma accesible. Con utilización de otros productos como SQL Server con manejo de datos espaciales se necesitaría un entrenamiento especial además del costo del producto correspondiente. .

6. Conclusiones

El desarrollo de este prototipo ha resultado un desafío muy interesante para las partes intervinientes, tanto por los retos asumidos desde lo técnico como por la relevancia que adquiere el mismo al responder a las necesidades de gestión de móviles ante incidentes en la ciudad de. La combinación del motor de Bases de Datos Relacional con el índice métrico i+3 R-tree para tratar objetos en movimiento, ha resultado muy satisfactorio, y los resultados de las primeras pruebas son alentadores, dado que el prototipo ha respondido de manera satisfactoria a las consultas efectuadas, devolviendo

adecuadamente los móviles disponibles en los radios más cercanos al lugar del evento.

Además este trabajo significó una posibilidad de estudiar de forma más amplia la temática de los distintos tipos de base de datos y la problemática asociada a los tipos de datos que no se adaptan a las estructuras estándares. Resultó además muy interesante combinar la teoría de los temas involucrados con el desarrollo de una solución basada en requerimientos reales.

7. Referencias

- [1] Ralf Hartmut Güting, Markus Schneider, Moving Objects Databases. Morgan Kaufmann Publishers. 2005.
- [2] Ralf Hartmut Güting. An Introduction to Spatial Database Systems. 1994, pages 1-7.
- [3] Volker Gaede and Oliver Günther. 1998. Multidimensional access methods. *ACM Comput. Surv.* 30, 2 (June 1998), 170-231.
- [4] Bela Stantic, John Thornton, Abdul Sattar. Approach to Model NOW in Temporal Databases. School of Information Technology Griffith University Gold Coast, Australia A Novel.
- [5] Christian S. Jensen, James Clifford, Ramez Elmasri, Shashi K. Gadia, Pat Hayes, Sushil Jajodia (editors) Curtis Dyreson, Fabio Grandi, Wolfgang Kafer, Nick Kline, Nikos Lorentzos, Yamzis Mitsopoulos, Angelo Montanari, Daniel Nonen, Elisa Peressi, Barbara Pernici, John F. Roddick, Nandlal L. Sarda, Maria Rita Scalas, Arie Segev, Richard T. Snodgrass, Mike D. Soo, Abdullah Tansel, Paolo Tiberio, Gio Wiederhold. A Consensus Glossary of Temporal Database Concepts
- [6] Anabella C. De Battista. Búsquedas por similitud sobre objetos dinámicos. Tesis para optar al grado de Magister en Ciencias de la Computación con orientación Bases de Datos. 2008.
- [7] Mohamed F. Mokbel, Thanaa M. Ghanem, Walid G. Aref. Spatio-temporal Access Methods. Department of Computer Sciences, Purdue University
- [8] Hee Kap Ahn, Nikos Mamoulis, Ho Min Wong. A Survey on Multidimensional Access Methods. Institute of Information and Computing Sciences, Utrecht University, The Netherlands, 2001
- [9] Gilberto Gutiérrez. Métodos de Acceso y Procesamiento de Consultas Espacio-Temporales. Tesis para optar al grado de Doctor en Ciencias mención Ciencias de la Computación.

[10] Antonin Guttman. 1984. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data (SIGMOD '84)*. ACM, New York, NY, USA, 47-57.

[11] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N. Papadopoulos and Yannis Theodoridis R-Trees: Theory and Applications. Springer-Verlag London Limited 2006, pages 99 - 104.

[12] Edilma O. Gagliari, Fernando D. Carrasco, Juan C. García Sosa, Gilberto Gutierrez. I+3 R-Tree: un método de acceso espacio-temporal. CACIC 2009.

[13] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. Computational Geometry: Algorithms and Applications. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008, pages 147-149