# Bases de Datos Métrico-Temporales

**Anabella De Battista , Andrés Pascal**
Departamento de Sistemas de Información
Universidad Tecnológica Nacional
Fac. Reg. Concepción del Uruguay
Entre Ríos, Argentina
{debattistaa, pascalj}@frcu.utn.edu.ar

**Norma Edith Herrera**
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
nherrera@unsl.edu.ar

**Gilberto Gutierrez**
Facultad de Ciencias Empresariales
Universidad del Bio-Bio
Chillán, Chile
ggutierr@ubiobio.cl

***Contexto***

*El presente trabajo se desarrolla en el ámbito del Grupo de Investigación en Bases de Datos (Proy. Nro 25-D040) perteneciente al Departamento de Sistemas de la Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay, cuyo objetivo principal es el estudio de métodos de acceso, procesamiento de consultas y aplicaciones de bases de datos no tradicionales.*

***Resumen***

*Las bases de datos métrico-temporales constituyen un nuevo modelo de bases de datos orientado al procesamiento de consultas por similitud en un intervalo o instante de tiempo. Este modelo está basado en la combinación de espacios métricos con bases de datos temporales. Para resolver eficientemente consultas métrico-temporales, se han propuesto varios índices cuyas evaluaciones empíricas demuestran que son competitivos. En este trabajo estamos interesado en el diseño de índices eficientes para el procesamiento de consultas métricos temporales.*

***Palabras claves:*** *Espacios Métricos, Bases de Datos Temporales, Bases de Datos Métrico-Temporales, Índices*

## 1. INTRODUCCIÓN

Las operaciones de búsquedas en una base de datos requieren de algún soporte y organización especial a nivel físico. En el caso de las bases de datos clásicas, la organización de la información se basa en el concepto de búsqueda exacta sobre datos estructurados. Esto significa que la información se organiza en registros con campos completamente comparables. Una búsqueda en la base retorna todos aquellos registros cuyos campos coinciden con los aportados en la consulta (búsqueda exacta). Otra característica importante de las bases de datos clásicas es que capturan sólo un estado de la realidad modelizada, usualmente el más reciente. Por medio de las transacciones, la base de datos evoluciona de un estado al siguiente descartando el estado previo.

Actualmente las bases de datos han incluido la capacidad de almacenar otros tipos de datos tales como imágenes, sonido, texto, video, datos geométricos, etc. La problemática de almacenamiento y búsqueda en estos tipos de base de datos difiere notablemente de las bases de datos clásicas en tres aspectos: primero los datos no son estructurados, esto significa que es imposible organizarlos en registros y campos, segundo la búsqueda

exacta carece de interés y tercero resulta de interés mantener todos los estados de la base de datos y no sólo el más reciente a fin de poder consultar el instante o intervalo de tiempo de vigencia de dichos objetos. Como solución a esta problemática surgen modelos que permiten procesar esta clase de datos. Entre estos nuevos modelos encontramos los siguientes:

**Espacios métricos** [1, 2, 6, 8, 9, 10, 5, 17, 12, 13], que permiten almacenar objetos no estructurados y realizar búsquedas por similitud sobre los mismos. Un espacio métrico es un par $(U, d)$ donde $U$ es un universo de objetos y $d : U \times U \rightarrow R^+$ es una función de distancia definida entre los elementos de $U$ que mide la similitud entre ellos. Una de las consultas típicas en este nuevo modelo de bases de datos es la búsqueda por rango, denotado por $(q, r)_d$, que consiste en recuperar los objetos de la base de datos que se encuentren como máximo a distancia $r$ de un elemento $q$ dado.

**Bases de datos temporales** [16, 11], que incorporan al tiempo como una dimensión, por lo que permiten asociar tiempos a los datos almacenados. Existen tres clases de bases de datos temporales en función de la forma en que manejan el tiempo: *de tiempo transaccional (transaction time)*, donde el tiempo se registra de acuerdo al orden en que se procesan las transacciones; *de tiempo vigente*, que almacenan el momento en que el hecho ocurrió en la realidad (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 modifica para actualizar el conocimiento de la realidad pasada, presente o futura, pero esas modificaciones se realizan generando nuevas versiones de los mismos estados.

**Bases de datos métrico-temporales** [3, 4, 15], que permiten almacenar objetos no estructurados con tiempos de vigencia asociados y realizar consultas por similitud y por tiempo en forma simultánea. Formalmente un *Espacio Métrico-Temporal* es un par $(U, d)$, donde $U = O \times N \times N$, y la función $d$ es de la forma $d : O \times O \rightarrow R^+$. Cada elemento $u \in U$ es una triupla $(obj, t_i, t_f)$, donde $obj$ es un objeto (por ejemplo, una imagen, sonido, cadena, etc) y $[t_i, t_f]$ es el intervalo de vigencia de $obj$. La función de distancia $d$, que mide la similitud entre dos objetos, cumple con las propiedades de una métrica (positividad, simetría y desigualdad triangular). Como un ejemplo de aplicación podemos mencionar una base de datos de rostros de delincuentes y cada foto tiene una intervalo de vigencia asociado, que representa el intervalo de tiempo en que el delincuente tenía el aspecto representado en esa foto; en este caso sería de interés, dada una foto y un intervalo de tiempo, poder recuperar de la base todos aquellos rostros parecidos al dado en el intervalo de tiempo especificado. Formalmente una *consulta métrico-temporal* se define como una 4-upla $(q, r, t_{iq}, t_{fq})_d$, tal que $(q, r, t_{iq}, t_{fq})_d = \{o/(o, t_{io}, t_{fo}) \in X \wedge d(q, o) \leq r \wedge (t_{io} \leq t_{fq}) \wedge (t_{iq} \leq t_{fo})\}$

Una forma trivial de resolver una consulta métrico-temporal, sin realizar un barrido secuencial sobre todos los elementos de la bases de datos, es construir un índice métrico agregándole a cada objeto el intervalo de tiempo de vigencia del mismo. Luego, ante una consulta $(q, r, t_{iq}, t_{fq})_d$ primero se utiliza el índice métrico para descarta aquellos objetos $obj$ que están a distancia mayor que $r$ de $q$; posteriormente se realiza un barrido secuencial sobre el conjunto de elementos no descartados por el paso anterior a fin de determinar cuáles objetos son realmente respuesta a la consulta, es decir, cuáles tienen un intervalo de vigencia que se superpone con $[t_{iq}, t_{fq}]$.

La desventaja que tiene esta solución trivial es que no se usa la componente temporal para mejorar el filtrado en el índice; en este proceso sólo se aprovecha la componente métrica. Una mejor estrategia es que durante el proceso de búsqueda se utilice tanto la componente métrica como la componente temporal para descartar elementos.

Varios índices métrico-temporales se han propuesto en este ámbito; todos ellos han tomado como base el Fixed Height Queries Tree, un índice para espacios métricos. El Fixed-Height FQT (FHQT) [2] construye un árbol a partir de un ele-

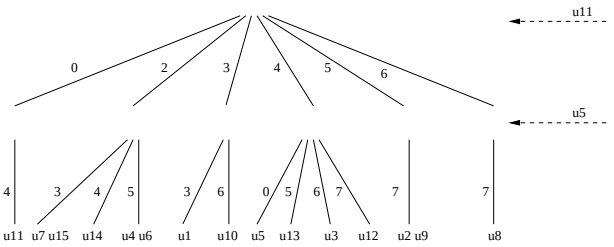


Figura 1: *Un ejemplo de un FHQT sobre un conjunto de 15 elementos*

mento $p$ (pivote) que puede ser elegido arbitrariamente, o mediante algún procedimiento de selección de pivotes [7], del universo $U$. Para cada distancia $i$ se crea el conjunto $C_i$ formado por todos aquellos elementos de la base de datos que están a distancia $i$ de $p$. Luego, para cada $C_i$ no vacío se crea un hijo del nodo correspondiente a $p$, con rótulo $i$, y se construye recursivamente un FHQT teniendo en cuenta que todos los subárboles del mismo nivel usarán el mismo pivote como raíz. Este proceso recursivo se continúa hasta lograr que todas las hojas estén en un mismo nivel y tengan menos de $b$ elementos, siendo $b$ un valor fijado previamente. La figura 1 muestra un ejemplo de un FHQT conjunto de $15$ elementos en los que se ha elegido $u_{11}$ como pivote en el primer nivel y $u_5$ como pivote del segundo nivel. Ante una consulta $(q, r)_d$, se comienza por la raíz y se descartan todas aquellas ramas con rótulo $i$ tal que $i \notin [d(p, q) - r, d(p, q) + r]$ siendo $p$ el pivote utilizado en la raíz. La búsqueda continúa recursivamente en todos aquellos subárboles no descartados, utilizando el mismo criterio.

Damos a continuación una breve reseña de los índices métricos-temporales que se basan en el FHQT:

**FHQT-Temporal** [15]. Este índice es una adaptación del Fixed Height Queries Tree (FHQT) en la que se agrega un intervalo de tiempo en cada nodo del árbol. Este intervalo representa el período máximo de vigencia para todos los objetos del subárbol cuya raíz es dicho nodo. En cada nodo hoja, este intervalo es el período total de vigencia de los objetos que contiene. Para cada nodo interior, el intervalo se calcula tomando el tiempo inicial mínimo, y el tiempo final máximo de sus hijos. Cuando se realiza una consulta métrico-temporal se procede de la siguiente manera: en cada nivel del árbol se filtran los subárboles hijos por el intervalo de tiempo de la consulta y luego de acuerdo a la distancia entre la consulta y el pivote. Al llegar al último nivel, se realiza una búsqueda secuencial sobre las hojas que no fueron descartadas seleccionando los objetos que cumplen con las condiciones temporales y de similitud.

**Historical-FHQT** [4]. Consiste en una lista de instantes válidos donde cada uno contiene un FHQT correspondiente a todos los objetos vigentes en dicho instante. Esta estructura es eficiente en bases de datos métrico-temporales en las que los objetos tienen vigencia en un solo instante de tiempo. Los FHQT tienen distintas profundidades en función de la cantidad de elementos que deban indexar. La cantidad de pivotes utilizada en un árbol se calcula como $\lceil log_2(|o_i|) \rceil$ donde $|o_i|$ es la cantidad de objetos vigentes en el instante $i$. De esta manera se evita que haya árboles con mayor profundidad de la necesaria, con el fin de que la estructura no tenga un costo excesivo en almacenamiento. Las consultas métrico-temporales se efectúan de la siguiente manera: en primer lugar se seleccionan los instantes incluidos en el intervalo de consulta. Luego se realizan consultas por similitud usando cada uno de los FHQT correspondientes, y finalmente se unen los conjuntos resultantes.

**Event-FHQT** [14]. Consiste en una lista de intervalos de tiempo válido consecutivos de tamaño fijo. Cada intervalo contiene un FHQT que indexa los objetos vigentes en el primer instante de dicho intervalo. Las hojas del FHQT contienen listas de eventos que indican los cambios que se produjeron entre dos intervalos. Presenta una ventaja respecto del Historical-FHQT, y es que no necesita duplicar los objetos vigentes en más de un instante de tiempo. Ante una consulta métrico-temporal primero se filtran los intervalos de la lista que se intersectan con el intervalo de consulta, luego por cada intervalo se realiza la consulta por similitud sobre el FHQT, se recorren las listas de eventos para determinar que objetos cumplen la restricción temporal de la consulta, por último se unen

los conjuntos resultantes y se compara cada elemento de ese conjunto con la consulta. .

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Nuestra principal línea de estudio e investigación es el desarrollo de índices métrico-temporales eficientes. El trabajo en curso se puede resumir en los siguientes puntos:

- Se sabe que la dimensionalidad de un espacio métrico afecta el desempeño de los índices [10]. En bases de datos métrico-temporales podría suceder que la dimensión de un conjunto de elementos en el instante $i$ sea distinta a la dimensión del conjunto de elementos en otro instante $j$ y en ese caso las decisiones tomadas con respecto a la construcción del índice deberían variar de un instante a otro. Por esta razón, un aspecto interesante a estudiar es el concepto de dimensionalidad aplicado a bases de datos métrico-temporales con el fin de encontrar una definición que se adecue a este nuevo modelo de bases de datos y que permita comprender mejor el desempeño de los índices.

- En base al punto anterior, se puede diseñar un índice híbrido que permita tener distintos índices métricos en distintos instantes de tiempo, según sea la dimensionalidad del conjunto de elementos almacenados en cada instante.

- Los índices desarrollados hasta el momento se basan en el supuesto de que la memoria principal tiene capacidad suficiente como para mantener tanto el índice como la base de datos. Si esto no es así, la cantidad de accesos a memoria secundaria realizados durante el proceso de búsqueda es un factor crítico en la performance del índice [18]. Nos proponemos explorar técnicas de paginado que sean aplicables a los índices métrico-temporales a fin de lograr que los mismos resulten eficientes también en memoria secundaria.

- Otro aspecto interesante a estudiar es el referido al espacio necesario para mantener el índice, dado que esto decide si el índice se mantendrá en memoria principal o en memoria secundaria. Una forma de reducir el espacio utilizado es tratar de reutilizar subárboles: si un subárbol del instante $i$ está también en el instante $j$ (con $j > i$), entonces el instante $j$ debería reutilizar el subárbol del instante $i$ en lugar de crearlo de nuevo. Esto implica diseñar un algoritmo que permita detectar subárboles isomorfos.

## 3. RESULTADOS OBTENIDOS/ESPERADOS

Se espera contar con índice eficiente métrico-temporal en memoria secundaria que sea eficiente tanto en los tiempos de respuesta como en el espacio ocupado por el mismo.

## 4. FORMACIÓN DE RECURSOS HUMANOS

El trabajo desarrollado hasta el momento forma parte del desarrollo de dos Tesis de Maestría en Ciencias de la Computación, una de ellas fue defendida y aprobada en marzo del corriente año. Se cuenta con el asesoramiento del Dr. Gilberto Gutiérrez, de la Universidad del Bio Bio, Chile. El grupo cuenta además con dos alumnos becarios que se están iniciando en las temáticas desarrolladas por el grupo.

## REFERENCIAS


[1] R. Baeza-Yates. Searching: an algorithmic tour. In A. Kent and J. Williams, editors, *Encyclopedia of Computer Science and Technology*, volume 37, pages 331–359. Marcel Dekker Inc., 1997.

[2] R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixed-queries trees. In *Proc. 5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.

[3] De Battista, A. Pascal, G. Gutierrez, and N. Herrera. Búsqueda en bases de datos métricas-temporales. In *Actas del VIII Workshop de Investigadores en Ciencias de*

*la Computación*, Buenos Aires, Argentina, 2006.

[4] De Battista, A. Pascal, G. Gutierrez, and N. Herrera. Un nuevo índice métrico-temporal: el historical fhqt. In *Actas del XIII Congreso Argentino de Ciencias de la Computación*, Corrientes, Agentina, 2007.

[5] S. Brin. Near neighbor search in large metric spaces. In *Proc. 21st Conference on Very Large Databases (VLDB'95)*, pages 574–584, 1995.

[6] W. Burkhard and R. Keller. Some approaches to best-match file searching. *Comm. of the ACM*, 16(4):230–236, 1973.

[7] B. Bustos, G. Navarro, and E. Chávez. Pivot selection techniques for proximity searching in metric spaces. In *Proc. of the XXI Conference of the Chilean Computer Science Society (SCCC'01)*, pages 33–40. IEEE CS Press, 2001.

[8] E. Chávez and K. Figueroa. Faster proximity searching in metric data. In *Proceedings of MICAI 2004. LNCS 2972, Springer*, Cd. de México, México, 2004.

[9] E. Chávez, J. Marroquín, and G. Navarro. Fixed queries array: A fast and economical data structure for proximity searching. *Multimedia Tools and Applications (MTAP)*, 14(2):113–135, 2001.

[10] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[11] C. S. Jensen. A consensus glossary of temporal database concepts. *ACM SIGMOD Record*, 23(1):52–54, 1994.

[12] I. Kalantari and G. McDonald. A data structure and an algorithm for the nearest point problem. *IEEE Transactions on Software Engineering*, 9(5):631–634, 1983.

[13] G.Ñavarro. Searching in metric spaces by spatial approximation. In *Proc. String Processing and Information Retrieval (SPIRE'99)*, pages 141–148. IEEE CS Press, 1999.

[14] A. Pascal, A. De Battista, G. Gutierrez, and N. Herrera. Indice metrico-temporal event-fhqt. In *Actas del XIIII Congreso Argentino de Ciencias de la Computación*, La Rioja, Argentina, 2008.

[15] A. Pascal, De Battista, G. Gutierrez, and N. Herrera. Procesamiento de consultas métrico-temporales. In *XXIII Conferencia Latinoamericana de Informática*, pages 133–144, San José de Costa Rica, 2007.

[16] B. Salzberg and V. J. Tsotras. A comparison of access methods for temporal data. *ACM Computing Surveys*, 31(2), 1999.

[17] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40:175–179, 1991.

[18] J. Vitter. External memory algorithms and data structures: Dealing with massive data. *ACM Computing Surveys*, 33(2):209–271, 2001.