

# Sistema de consulta flexible utilizando dispositivos móviles.

## Aplicación a la búsqueda de restaurantes.

José Enrique Pons Frías<sup>1</sup> y Amparo Vila Miranda<sup>2</sup>

*Resumen*— En este trabajo se implementa un entorno de consulta flexible, también denominada consulta con preferencias, en un entorno de computación ubicua. El usuario, equipado con un dispositivo móvil, tal como un PDA o smartphone realiza una consulta a una base de datos. Los parámetros de la consulta están personalizados con las preferencias de cada usuario. Por lo tanto, los resultados que se obtienen están más ajustados a las preferencias del usuario. En esta aplicación, hemos implementado la consulta flexible de restaurantes.

*Palabras clave*— Computación ubicua, consulta flexible, bases de datos difusas.

### I. INTRODUCCIÓN

Los seres humanos manejamos información frecuentemente de manera imprecisa. Por ejemplo, en lugar de manejar cantidades específicas a la hora de referirnos a una distancia, preferimos utilizar el término *cerca* o *lejos*. Se ha realizado mucho esfuerzo a la hora de tratar de capturar esta semántica en los sistemas de información. Las bases de datos difusas han tenido gran aceptación a la hora de modelar la imprecisión. Sin embargo, existe la necesidad de que tanto la representación en la base de datos como los resultados de las consultas se ajusten a las preferencias del usuario. Diferentes usuarios pueden tener una interpretación distinta de un mismo concepto. Aún más, un mismo usuario, en diferentes contextos puede necesitar una interpretación diferente de los mismos conceptos.

Imaginemos un usuario que busca un restaurante con un tipo determinado de cocina y en el que el usuario se desplaza andando hacia el restaurante. Es evidente que la representación del concepto *cerca* puede ser un intervalo, p.ej [0, 100] metros. Sin embargo, si el mismo usuario se desplaza en coche, la representación más adecuada del concepto *cerca* puede ser el intervalo [0, 1500] metros.

Por lo tanto, en este trabajo presentamos una arquitectura diseñada para dispositivos móviles con capacidades de consulta flexible ajustada a preferencias de usuario. El diseño de la base de datos es específico para restaurantes. Los datos en la base de datos son conocidos con precisión, mientras que las consultas que especifica el usuario pueden contener algún tipo de imprecisión.

El resto del trabajo está organizado de la siguiente manera: La sección II presenta los principales con-

ceptos teóricos así como trabajos previos. La sección III describe la propuesta del sistema. Comienza con la descripción de las restricciones que se han de contemplar. Sigue con la arquitectura del sistema y finaliza con los detalles de implementación del mismo. Finalmente la sección IV describe las conclusiones y líneas de trabajo futuro.

### II. PRELIMINARES

A continuación realizaremos una breve introducción a las bases de datos difusas explicando tanto la representación de incertidumbre como la consulta con imprecisión. Mostraremos también un listado con los modelos de bases de datos más utilizados. Seguidamente haremos un recorrido por los dispositivos móviles, sus características y sus limitaciones que guiarán tanto el diseño como la implementación de la aplicación.

#### A. Bases de datos difusas

Estas bases de datos se llaman así por utilizar como base la lógica difusa (conjuntos difusos, introducidos por Zadeh [1] y su conexión con la teoría de la posibilidad [2]). Existen numerosas propuestas que son una extensión al modelo relacional [3], [4], [5], [6], [7], [8], [9] introducido por Codd [10], [11] en los setenta.

En los posteriores años, se realizó una prolífica investigación sobre las bases de datos relacionales. Podemos encontrar varios modelos con sus respectivas álgebras: [12], [13], [14], [15], [16], [17], [18]. De todos ellos, *GEFRED* (A generalized model for fuzzy relational databases) [19] es el que aglutina diferentes tipos de datos propuestos en modelos anteriores. Su implementación es FS<sub>F</sub>SQL [20], [21] y se ejecuta sobre el sistema de bases de datos Oracle, aunque también hay una implementación para PostgreSQL.

Se han propuesto diferentes sistemas de consulta para bases de datos difusas como [22], [23], *SQLf* [24] y más recientemente [25]. En cuanto a implementaciones, encontramos las siguientes: *Freedom-0* [26], *SQLf* [27]. En [28] encontramos una explicación del diseño e implementación de aplicaciones utilizando bases de datos difusas.

En lo que sigue utilizaremos el modelo GEFRED [19] y los conceptos de implementación para bases de datos relacionales incluidos en [20].

Los principales objetivos de una base de datos difusa son dos:

<sup>1</sup>Dpto. de Ciencias de la Computación e I.A. Universidad de Granada, e-mail: jpons@decsai.ugr.es

<sup>2</sup>Dpto. de Ciencias de la Computación e I.A. Universidad de Granada, e-mail: vila@decsai.ugr.es

*Representación* En este modelo se distinguen tres tipos principales de datos:

*Tipo 1* : Es cualquier tipo de datos que puede ser consultado de manera flexible. Suelen ser tipos de datos numéricos que no contienen ningún tipo de imprecisión. P. ej., El precio medio de un restaurante puede ser almacenado de manera precisa, pero el usuario puede querer consultarlo especificando un concepto, como por ejemplo *barato*. Dicho concepto, para un usuario determinado puede considerarse como el intervalo  $[0, 20]$  euros.

*Tipo 2* : Datos difusos que tienen un dominio ordenado. Se trata de tipos de datos cuyo dominio subyacente suele ser numérico. Por ejemplo, el precio de un restaurante puede no ser conocido con exactitud. En este caso, se puede representar como un intervalo o una distribución de posibilidad de diversos tipos (triangular, trapezoidal). Además se suelen definir una serie de operadores (como la igualdad difusa) para poder comparar los distintos valores de este tipo de datos. En este caso, se puede considerar una distribución de posibilidad como la que se muestra en la figura y que representamos como  $[0, 10, 15, 20]$

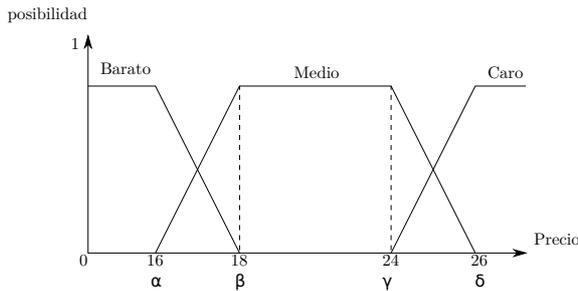


Fig. 1. Distribución de posibilidad asociada al precio. Hay tres etiquetas lingüísticas: barato, precio medio, y caro.

*Tipo 3* : Datos difusos sobre un dominio no ordenado. Este tipo de datos tiene un dominio que no es ordenado. En este caso, sólo se puede definir la operación de equivalencia entre clases. Un ejemplo de este tipo de datos puede ser el tipo de cocina de un restaurante. Podemos tener las siguientes clases: tradicional, italiana, japonesa, comida rápida. Se suele definir una tabla de equivalencia (ver tabla I para cada una de las clases de modo que sea posible obtener un grado de similitud entre clases.

*Consulta* El usuario puede especificar en la consulta cualquiera de estos tres tipos descritos con anterioridad. En el lenguaje de consulta se suele proporcionar los operadores de igualdad difusa, mayor, menor que, así como operadores extendidos como por ejemplo, mucho mayor que y mucho menor que.

Hay que hacer notar la diferencia de semántica que tienen los datos que se almacenan en la base de datos con respecto a los datos que se especifican en una

TABLA I

EJEMPLO DE UN TIPO DE DATOS NO ORDENADO, COMO ES EL TIPO DE COCINA.  $C_1 = \text{PIZZERÍA}$ ,  $C_2 = \text{HAMBURGUESERÍA}$ ,  $C_3 = \text{CHINA}$ ,  $C_4 = \text{COMIDA RÁPIDA}$

	$C_1$	$C_2$	$C_3$	$C_4$
$C_1$	1	0	0	.6
$C_2$	0	1	0	.8
$C_3$	0	0	1	.4
$C_4$	.3	.6	.4	1

consulta de usuario:

*Almacenamiento en Base de Datos* Cuando almacenamos un elemento (objeto) en la base de datos, utilizando alguno de los tipos 2 o 3, lo que estamos indicando es que el objeto solo tiene un valor, por ejemplo el precio medio es un sólo valor, pero que por algún motivo, se desconoce con exactitud. Esto nos lleva a una *interpretación disjunta* de los valores que se almacenan. Si se almacena una distribución de posibilidad como la que se muestra en la figura 1, dada por  $[16, 18, 24, 26]$ , lo que estamos indicando es que el precio medio del restaurante es o bien 10, o bien 11, ... y así hasta 20 euros.

*Especificación en la Consulta* Cuando el usuario especifica una distribución de posibilidad en la consulta lo que quiere es que se le devuelvan todos los valores que se indican en dicha distribución. Esto es, nos encontramos ante una *interpretación conjuntivista* de los valores de un conjunto.

Las distintas interpretaciones de un conjunto difuso han sido especificadas por Dubois en [29].

### B. Dispositivos móviles

La capacidad de los dispositivos de computación móviles ha aumentado últimamente, hasta el extremo de ser auténticos ordenadores de bolsillo, con procesadores bastante potentes. Hoy podemos encontrar teléfonos móviles, que proporcionen la funcionalidad de un reproductor de MP3, o la posibilidad de editar un archivo de texto con un programa igual de potente que el que utilizamos para editar en nuestro ordenador de sobremesa, incluso el poder realizar fotos de calidad.

Además, las posibilidades de conexión de nuevos periféricos es casi infinita, debido en parte a los nuevos estándares de comunicación (BlueTooth, WiFi), etc, que posibilitan conectar nuestro dispositivo móvil, a cualquier otro aparato que utilice esta tecnología de interconexión, como por ejemplo un GPS, una impresora, ..., etc.

Como vemos, la tendencia es aglutinar los servicios proporcionados por diversos dispositivos en uno solo. Es común a día de hoy, encontrar smartphones con conexión a Internet, ya sea por wifi o por 3G así como GPS.

En la actualidad encontramos los siguientes sistemas operativos principales para dispositivos

móviles:

*Android* Sistema operativo desarrollado por Google y basado en el kernel de Linux. Las aplicaciones para este sistema operativo se ejecutan en un framework Java.

*Windows Phone* Sistema operativo desarrollado por Microsoft. Las aplicaciones para este sistema han de ser desarrolladas con Microsoft Silverlight o con Microsoft XNA.

*iOS* Sistema operativo móvil de Apple. Se trata de un sistema operativo basado en Linux. Para desarrollar aplicaciones en este sistema hay que utilizar el framework de Apple.

*Blackberry OS* Este sistema operativo es el que incorporan los dispositivos Blackberry muy ligados al mundo empresarial. Para desarrollar aplicaciones, también es necesario utilizar el framework propio de desarrollo.

*Symbian OS* Es un sistema operativo propietario usado principalmente por los teléfonos de la compañía Nokia. La principal ventaja de este sistema es que puede ejecutar código Java.

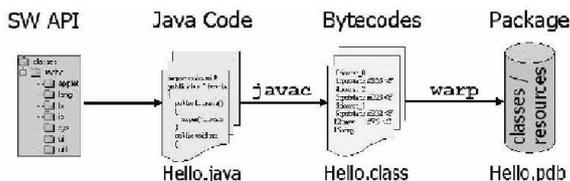


Fig. 2. Proceso de compilación en SuperWava.

Dada la gran variedad de sistemas operativos que existen en la actualidad, hay varias opciones a la hora de desarrollar una aplicación para un dispositivo móvil: o bien conocer de antemano la arquitectura para la que se desarrolla y realizar el desarrollo con el correspondiente framework, o bien utilizar un framework que nos permita desarrollar para distintos sistemas operativos. En nuestro caso, escogemos la segunda opción. Existe un framework Java multiplataforma, llamado *SuperWaba*[30] que nos permite con un mismo código desarrollar aplicaciones para: Android, Linux, Windows, Blackberry, Ios y Symbian. La figura 2 ilustra el proceso de compilación.

### III. PROPUESTA

A continuación explicaremos la arquitectura que hemos definido para el sistema, así como la estructura que tiene el sistema de información.

#### A. Arquitectura del Sistema

El sistema tiene una arquitectura con dos componentes (ver figura 3):

- Servidor: Se trata de una máquina que contiene una versión completa de la base de datos. En nuestro caso, la base de datos es MySQL.
- Cliente: Es el dispositivo móvil, que debe de estar equipado con un receptor GPS. Dicho dispositivo exporta del servidor una versión (en

XML) reducida de la base de datos. Por ejemplo, si el servidor contiene una base de datos de toda España, y el cliente se mueve por Granada, la base de datos que contiene el dispositivo móvil es una versión que contiene únicamente los restaurantes de Granada.

Hay que remarcar que el funcionamiento de este sistema no requiere una conexión constante entre el cliente y el servidor, y por lo tanto no podemos decir que se trate de una arquitectura cliente-servidor típica. La finalidad de este comportamiento es minimizar la cantidad de datos que se transfieren así como lograr la autonomía del cliente dentro de una localización geográfica determinada.

También es posible exportar aquellos datos que satisfagan las preferencias indicadas en un determinado perfil de usuario. Aunque esta funcionalidad no se encuentra implementada en esta versión.

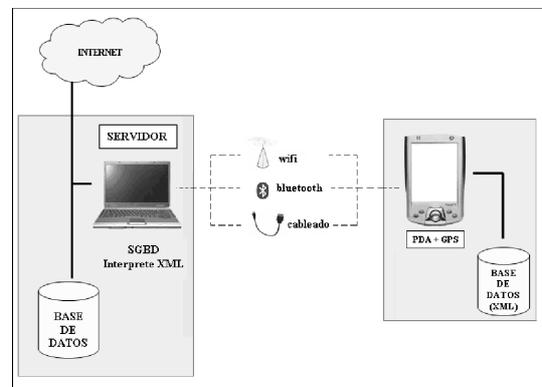


Fig. 3. Arquitectura de la aplicación.

#### B. Sistema de Información

Para implementar nuestro sistema, necesitamos construir un sistema de información que nos permita representar con suficiente detalle los elementos de los que consta el sistema (en nuestro caso, restaurantes y perfiles de usuario). A continuación describiremos por un lado, los elementos del sistema de información que representa la base de datos con los datos de restaurantes, y por otro lado la información que contienen los perfiles de usuario.

##### B.1 Base de Datos

Como resultado del análisis de diversas guías de restauración, se realiza un filtrado de los campos más representativos de un restaurante. También indicamos entre corchetes si ese tipo de datos es difuso (tipos 1 a 3) tal y como hemos indicado en la sección II-A:

- Nombre, así como una breve descripción.
- Precio medio [Tipo 1], así como precio mínimo y máximo.
- Dirección: Se trata de un tipo de dato complejo que dividimos de la siguiente manera:
  - Nombre de la calle.
  - Teléfono para reservas, email, página web.

- Localización GPS [Tipo 1]: Esto a su vez es un dato compuesto por dos coordenadas: latitud y longitud.
- Servicios adicionales: Son atributos de caracter booleano que indican si se dispone o no del servicio: Servicio de párking, Lugar pintoresco, Fumadores, Zonas para niños, Jardín, Aire acondicionado, Bodega, Acceso minusválidos.

## B.2 Perfil de Usuario

La intención del perfil del usuario es que el usuario especifique qué es lo que entiende por determinados valores, por ejemplo por cerca y barato, de modo que se puedan resolver consultas del tipo:

“Quiero obtener todos los restaurantes que se encuentren cerca y que sean baratos”.

Los elementos que forman parte del perfil de usuario son los siguientes:

**Distancia** : Se especifica mediante valores de tipo 2 las siguientes tres etiquetas: ‘Cerca’, ‘Media’, ‘Lejos’. La relación entre las etiquetas se ilustra en la figura 4. En el perfil del usuario se almacenan los valores  $[\alpha, \beta, \gamma, \delta]$ .

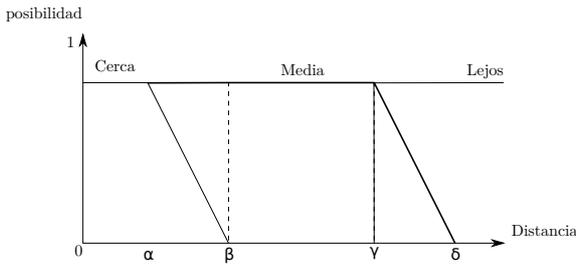


Fig. 4. Distribución de posibilidad asociada a la distancia. Hay tres etiquetas lingüísticas: cerca, medio, y lejos.

**Precio** : Se especifica mediante valores de tipo 2 las siguientes tres etiquetas: ‘Barato’, ‘Medio’, ‘Caro’. (Ver figura 1). Aquí también se almacenan los valores  $[\alpha, \beta, \gamma, \delta]$  en el perfil del usuario.

**Tipo de cocina** : El usuario introduce sus preferencias con respecto al tipo de cocina (tipo 3, ver tabla I).

**Otros servicios** : Los campos adicionales que especificamos para los restaurantes en el apartado anterior, se consideran en el perfil, así, un “plan” de salida romántico consistiría en seleccionar las opciones de bodega y lugar pintoresco. Sin embargo, si salimos con niños, quizás un plan más adecuado sería que el restaurante disponga de zonas para niños y jardín.

## B.3 Resolución de consultas

Una consulta está compuesta por los valores que se indican en el perfil del usuario, por lo tanto, se puede definir de la siguiente manera:

$$\tilde{Q} = \{q_D, q_P, q_C, q_S\} \quad (1)$$

$$\Omega = \{\omega_D, \omega_P, \omega_C, \omega_S\} \quad (2)$$

Donde cada uno de los elementos de  $\tilde{Q}$  son:

$q_D$  Es la etiqueta lingüística que se escoge para la distancia. (Cerca, Media, Lejos)

$q_P$  Es la etiqueta lingüística que se escoge para el precio. (Barato, Medio, Caro).

$q_C$  Es el tipo de cocina que busca el usuario. ( $C_1, \dots, C_4$  indicados en tabla I).

$q_S$  Es la combinación booleana de los tipos de servicios que escoge el usuario.

El vector  $\Omega$  es un vector de pesos. El usuario indica el peso específico de cada uno de los atributos dentro del cómputo total. Es necesario que la suma de todos los elementos de  $\Omega$  sume uno.

El proceso de ejecución de la consulta consta de dos pasos:

1. Cálculo del grado de cumplimiento de cada uno de los elementos de la consulta ( $q_D, \dots, q_S$ ).
2. Agregación de cada uno de estos grados en un grado total de satisfacción.

Pasamos a detallar cada uno de los procesos:

**B.3.a Cálculo del grado de cumplimiento.** Para calcular los grados de cumplimiento de los atributos precio y distancia, necesitamos calcular el grado de pertenencia de un elemento a un conjunto difuso definido por  $f_s = [\alpha, \beta, \gamma, \delta]$ . Dado un elemento  $x \in X$ , su pertenencia al conjunto difuso dado por  $f_s$  está determinado por la función  $\mu_T(x) \in [0, 1]$ :

$$\mu_T(x) = \begin{cases} 1 & \text{if } x \in [\beta, \gamma] \\ 0 & \text{if } x > \delta \vee x < \alpha \\ \frac{x-\alpha}{\beta-\alpha} & \text{if } x \in [\alpha, \beta] \\ \frac{\delta-x}{\delta-\gamma} & \text{if } x \in [\gamma, \delta] \end{cases} \quad (3)$$

Por cada registro  $r$  de la base de datos reducida, se evalúan de la siguiente manera:

**Distancia** : Denotamos por  $r_D$  la distancia en metros desde el restaurante hasta la posición GPS del usuario. El cálculo del grado de cumplimiento de la distancia viene dado por:  $\mu_{q_D}(r_D)$ .

**Precio** : El precio medio del restaurante es  $r_P$ . El grado de cumplimiento del precio es:  $\mu_{q_P}(r_P)$ .

**Tipo de Cocina** : En este caso,  $r_C$  es el tipo de cocina del restaurante y  $q_C$  el tipo de cocina indicado en el perfil de usuario. El grado de cumplimiento del tipo de cocina viene dado por la tabla I. Notamos por  $\mu_{q_C}$  al grado de cumplimiento para el atributo cocina.

**Servicios** : El conjunto de servicios que tiene un restaurante es un vector booleano  $V_{r_S}$ . El conjunto de servicios que prefiere el usuario viene indicado en el vector  $V_{q_S}$ . El grado de cumplimiento del atributo servicios, viene dado por la siguiente expresión:  $\mu_{q_S} = \frac{\sum_{i=1}^n V_{r_S}[i] \wedge V_{q_S}[i]}{n}$

**B.3.b Grado de satisfacción global.** Para calcular el grado de satisfacción global de la consulta, basta

con calcular la suma ponderada de cada uno de los grados de satisfacción:

$$\mu_{\tilde{Q}}(r) \rightarrow [0, 1] = \sum_{i=1}^{i=n} \omega_i \cdot \mu_i \quad (4)$$

$$\begin{aligned} \mu_{\tilde{Q}}(r) = & \omega_D \cdot \mu_{q_D}(r_D) + \\ & \omega_P \cdot \mu_{q_P}(r_P) + \\ & \omega_C \cdot \mu_{q_C}(r_C) + \\ & \omega_S \cdot \mu_{q_S}(r_S) \end{aligned} \quad (5)$$

### C. Caso de uso

A continuación ilustraremos el funcionamiento de la aplicación con un ejemplo de caso de uso. La pantalla principal de la aplicación es la que se muestra en la figura 5. Aquí se pueden introducir las coordenadas GPS a mano, o bien obtenerlas del receptor GPS incorporado. Además, se puede especificar qué se desea buscar, restaurantes baratos y cercanos, por ejemplo. El resto de valores son leídos del perfil del usuario en ese momento.

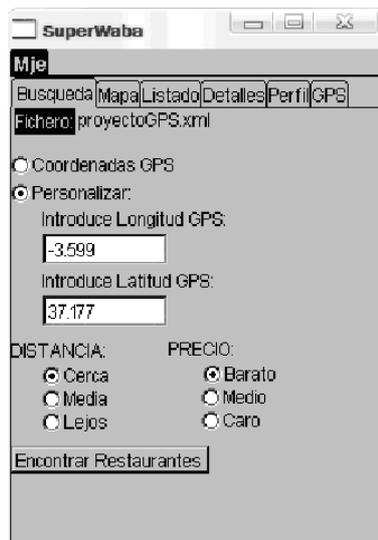


Fig. 5. Pantalla principal de la aplicación.

La figura 6 muestra un ejemplo del perfil del usuario para el atributo distancia, así como la asignación del peso  $\omega_D$  para este atributo concreto.

El resultado de una consulta es un listado de restaurantes ordenado según las preferencias del usuario, tal y como muestra la figura 7. Si se pulsa sobre un restaurante, se puede ver el detalle del mismo, con información sobre la dirección, el horario, el precio medio y los servicios de los que dispone (figura 8).

Finalmente, es posible ver dónde se encuentra cada restaurante en la vista de mapa (figura 9). Se indican con distinto color la situación del usuario y la situación del restaurante.

## IV. CONCLUSIONES

En este trabajo hemos mostrado como realizar una consulta con preferencias. Las preferencias

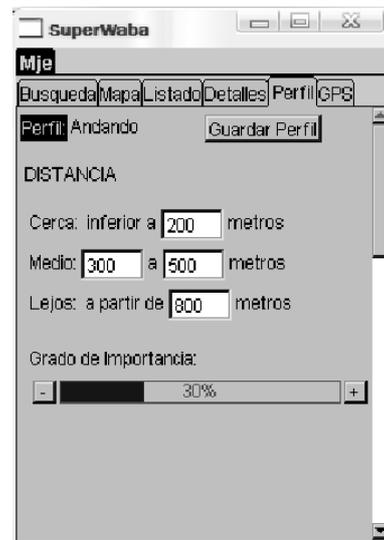


Fig. 6. Perfil de usuario. Asignación del grado de importancia.



Fig. 7. Resultado de una consulta. Listado de restaurantes.

las expresa el usuario por medio de un perfil de usuario en el que se describen mediante conjuntos difusos unos valores aproximados para determinados atributos, como la distancia, el precio, etc. Se proporciona una implementación en un dispositivo móvil, todo el código y la documentación se encuentran disponibles en la siguiente dirección: <http://code.google.com/p/mje-pda>.

Como trabajo futuro, planteamos acceder a fuentes de información adicionales, así como extender la consulta con preferencias a otros ámbitos con un dispositivo móvil.

## AGRADECIMIENTOS

Parte de los investigadores reciben fondos del proyecto de investigación TIN2008-02066 y la ayuda BES-2009-013805.

## REFERENCIAS

- [1] Lotfi A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

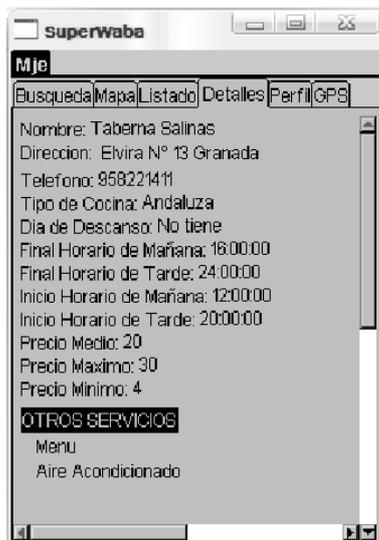


Fig. 8. Resultado de una consulta. Detalle de un restaurante.



Fig. 9. Resultado de una consulta. Mapa con los restaurantes.

[2] Lotfi Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[3] E. F. Codd, "Extending the database relational model to capture more meaning," *ACM Transactions on Database Systems*, vol. 4, pp. 397–434, 1979.

[4] Tomasz Imieliński and Jr. Witold Lipski, "Incomplete information in relational databases," *J. ACM*, vol. 31, no. 4, pp. 761–791, Sept. 1984.

[5] Yannis Vassiliou, "Null values in data base management a denotational semantics approach," in *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*, New York, NY, USA, 1979, SIGMOD '79, pp. 162–169, ACM.

[6] Yannis Vassiliou, "Functional dependencies and incomplete information," in *Proc. 6th Int. Conf. on Very Large Data Bases*, New York. ACM, Ed., Montreal, Ont. Canada, Oct 1980, pp. 268–269.

[7] John Grant, "Null values in a relational data base," *Inf. Process. Lett.*, vol. 5, no. 5, pp. 156–157, 1977.

[8] Carlo Zaniolo, "Database relations with null values," in *Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, ACM New York, Ed., Los Angeles, Calif., March 1982, pp. 27–33.

[9] Guy De Tré, Rita M. M. De Caluwe, and Henri Prade, "Null values revisited in prospect of data integration," in *ICSNW*, Mokrane Bouzeghoub, Carole A. Goble, Vipul Kashyap, and Stefano Spaccapietra, Eds. 2004, vol. 3226 of *Lecture Notes in Computer Science*, pp. 79–90,

Springer.

[10] E. F. Codd, "A relational model of data for large shared data banks," *Commun. ACM*, vol. 13, no. 6, pp. 377–387, June 1970.

[11] J. D. Ullman, *Principles of Database Systems*, Computer Science Press, 1982.

[12] M. Zemankova-Leech and A. Kandel, "Fuzzy relational databases - a key to expert systems," *Journal of the American Society for Information Science*, vol. 37, pp. 272–273, 1984.

[13] Billy P. Buckles and Frederick E. Petry, "Extending the fuzzy database with fuzzy numbers," *Information Sciences*, vol. 34, no. 2, pp. 145–155, 1984.

[14] Billy P. Buckles and Frederick E. Petry, "A fuzzy representation of data for relational databases," *Fuzzy Sets and Systems*, vol. 7, no. 3, pp. 213–226, 1982.

[15] H. Prade and C. Testemale, "Representation of soft constraints and fuzzy attribute values by means of possibility distributions in data bases," *The Analysis of Fuzzy Information*, vol. II, 1987.

[16] Henri Prade and Claudette Testemale, "Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries," *Information Sciences*, vol. 34, no. 2, pp. 115–143, 1984.

[17] Elke A. Rundensteiner, Lois W. Hawkes, and Wyllis Bandler, "On nearness measures in fuzzy relational data models," *International Journal of Approximate Reasoning*, vol. 3, no. 3, pp. 267–298, 1989.

[18] Sujet Sheno and Austin Melton, "Proximity relations in the fuzzy relational database model," *Fuzzy Sets and Systems*, vol. 100, Supplement 1, no. 0, pp. 51–62, 1989.

[19] M. A. Vila, J. C. Cubero, J. M. Medina, and O. Pons, "A logic approach to fuzzy relational databases," *International Journal of Intelligent Systems*, vol. 9, pp. 449–460, 1994.

[20] Jos'e Galindo, Juan Medina, Olga Pons, and Juan Cubero, "A server for fuzzy sql queries," in *Flexible Query Answering Systems*, Troels Andreassen, Henning Christiansen, and Henrik Larsen, Eds. 1998, vol. 1495 of *Lecture Notes in Computer Science*, pp. 164–174, Springer Berlin / Heidelberg.

[21] José Galindo, "New characteristics in fsql, a fuzzy sql for fuzzy databases," in *Proceedings of the 4th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering Data Bases*, Stevens Point, Wisconsin, USA, 2005, AIKED'05, pp. 4:1–4:9, World Scientific and Engineering Academy and Society (WSEAS).

[22] P. Bosc, M. Galibourg, and G. Hamon, "Fuzzy querying with sql: extensions and implementation aspects," *Fuzzy Sets Syst.*, vol. 28, pp. 333–349, December 1988.

[23] J. Kacprzyk and A. Ziolkowski, "Database queries with fuzzy linguistic quantifiers," *IEEE Trans. Syst. Man Cybern.*, vol. 16, pp. 474–479, May 1986.

[24] P. Bosc, O. Pivert, and K. Farquhar, "Integrating fuzzy queries into an existing database management system: An example," *International Journal of Intelligent Systems*, vol. 9, no. 5, pp. 475–492, 1994.

[25] Guy De Tré and et al. Zadrozny, "Dealing with Positive and Negative Query Criteria in Fuzzy Database Querying Bipolar Satisfaction Degrees," in *Proceedings of 8th Int. Conf. FQAS*, Denmark, 2009, pp. 593–604, Springer Verlag Berlin.

[26] Motohide Umamo, "Freedom-0: A fuzzy database system," *Fuzzy Inf and Decis Processes*, pp. 339–347, 1982, cited By (since 1996) 36.

[27] Leonid Tineo, Marlene Goncalves, and Juan Carlos Eduardo, "A fuzzy querying system based on sqlf2 and sqlf3," in *CLEI2004*, Mauricio Solar, David Fernández-Baca, and Ernesto Cuadros-Vargas, Eds., Sept. 2004, pp. 845–851.

[28] Jose Galindo, *Fuzzy Databases: Modeling, Design, and Implementation*, IGI Publishing, Hershey, PA, USA, 2006.

[29] Didier Dubois and Henri Prade, "The three semantics of fuzzy sets," *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 141–150, 1997.

[30] SuperWaba Br., "http://www.superwaba.com.br/," .