Publicaciones en Ciencias y Tecnología. Vol.12, Nº2, Julio-Diciembre (2018) 97–104

Artículo de investigación

Dynamic construction of fuzzy queries on a database of projects

Construcción dinámica de consultas difusas sobre una base de datos de proyectos

Livia Borjas[a], Leonid Tineo[b]

[a]*Instituto Universitario de Tecnología Dr. Federico Rivero Palacio. Venezuela*
[b]*Universidad Simón Bolivar. Caracas, Venezuela.*

Recibido: 15-03-2018 Aceptado: 12-10-2018

Abstract

In this paper an application for evaluation and control of software projects is presented. The novelty of this application is that it has been developed using an extended database management system with fuzzy logic. In addition to the usual tasks of a project control tool, this application allows to evaluate the management of a project, taking into consideration the benefits of fuzzy queries.

Key words: SQLf, fuzzy querying, database application, DBMS, project management.
UNESCO Code: 120304- Artificial intelligence

Resumen

En este trabajo se presenta una aplicación para evaluación y control de proyectos de software. La novedad de esta aplicación es que ha sido desarrollada usando un sistema gestor de bases de datos extendido con lógica difusa. Además de las tareas habituales de una herramienta de control de proyectos, esta aplicación permite evaluar la gestión de un proyecto, aprovechando las bondades de consultas difusas.

Palabras clave: SQLf, consultas difusas, aplicación en bases de datos, SGBD, gestión de proyectos.
Código UNESCO: 120304- Inteligencia Artificial

1. Introduction

Nowadays, the Projects management Systems awakens great interest since the demand of these projects has increased, as well as the number of companies that have dedicated to providing this service. Therefore, the authors of this article are interested in the implementation of an automated tool aimed at supporting required functionalities for Project management systems. The management of project is required to perform the following functions: −−Administration of projects, tasks, resources and expenses; −−Selection of technical resources and leadership; and −−Control and follow-up of work flow.

Impresa: ISSN 1856-8890. Dep. Legal pp200702LA2730. Digital: ISSN 2477-9660. Dep. Legal ppi201402LA4590.
Email addresses: livacaro7@gmail.com (Livia Borjas), leonid@usb.ve (Leonid Tineo)

In general terms, a tool supporting the management and administration of developing projects carries out the same functions regardless the type of project being executed, i.e. the tasks to be considered for a software projects are the same as those considered for a city planning project. Therefore, it is considered that the contributions achieved in this context might be rapidly used in other contexts of interests for the management and administration of projects in general.

Furthermore, this research intends to offer a more interesting contribution to the Projects management. The administration of tasks, resources and expenses as well as the control and follow-up of the workflow are functions that are already automated through very popular tools that have been used a long time ago.

However, since these tools do not respond to information requests such as: *Given the context of the project and considering the deadline and budget, which is the best group of programmers to achieve this?; How long is time required to deliver a functional prototyping to the customer?; What action should be executed by a manager when evidencing delays in the delivery to a particular customer?*. These are only some value questions in this context that are difficult to express in a traditional query language including change request and unplanned events causing delaying in project delivery.
Conditions involved in these questions criteria could not directly be expressed through traditional logic, they require fuzzy logic. Therefore, we propose to use an extended database management system with fuzzy logic. Such database management systems allow the expression of queries, among others options, involving fuzzy conditions. This type of query is known as fuzzy queries.
This research addresses to the design and implementation of an application for evaluation and control of software projects provided with a fuzzy querying interface over its classical database, enhancing decision-making.
This paper is organized as follows: Section 2 concerns to the motivation of the reported work. Section 3 gives a background in fuzzy querying to relational databases. Section 4 describes our Control and Evaluation System for Software Projects. Finally, we out the concluding remarks of this work.

2. Motivation

The use of Boolean Logic restricts the treatment of imprecision or uncertainty in database management systems (DBMS). The queries do not get prospective results, because these queries are restrictive and they do not contemplate important elements that could be considered to analyze data. A solution to this problem is using fuzzy sets in DBMS [1] to express non-crisp data and gradual (or flexible) requirements. In previous works, we have concentrated our efforts in the study and implementation of fuzzy querying languages on relational databases since many organizations store their information in relational databases.

There are some different proposals of fuzzy set based flexible querying languages [2, 3, 4]. One of the most representatives is SQLf [2] a fuzzy extension of the standard SQL that allows using fuzzy conditions in any place where SQL allows a regular one. Some real implementations of SQLf were presented in the work [5]. One of such flexible querying systems is named SQLfi (SQLf on the Internet).

Some previous research works deal with real life applications for fuzzy databases and flexible querying systems [6, 7, 8, 9]. They explore the actual advantages of using an extended database management system with fuzzy logic, such as SQLfi.

The main motivation of this paper is present an application that we have developed as an experience with SQLfi. We have selected as case study for this development a Project management system because this is a tool that would be useful for the sake of computer science engineer.

3. Fuzzy querying

The idea of the database arose from the necessity of storing great amount of information. Databases allows saving data so that multiple users and applications for a long period of time can access them engine tool. Data are usually accessed through classical queries, which are based on Boolean conditions. This type of queries has an inflexibility problem, due to the fact it sometimes does not show all the expected results. Additionally, it is unable to obtain the responses that are close to meet the specific conditions that might be of interest to users. Furthermore, there is no discrimination among different responses.

For purposes of sorting out this issue, research works have been developed aimed at managing the queries based on users' preferences. From these works arose the database flexible query system based on fuzzy logic named SQLfi [8]. This query system promises to be a very useful tool for managing databases since it offers users the facility of expressing

their preferences in the query, in addition to make decisions based on information obtained, since the responses are discriminated by these preferences.

Fuzzy logic is a powerful tool aimed at dealing with preferences. Through this tool, users might express their requirements by using conditions based on the use of linguistic terms, so software projects do not need to force users to establish inflexible sentences. Moreover, the system knows users' preferences and makes the discrimination of the responses. Semantically, this tool is ideal and in some areas it represents the most general tool to manage expressions over flexible conditions in database queries.

Fuzzy Logic [10] is based in fuzzy set theory. A fuzzy set F [11] is a special kind of collection of elements from a classical set X (universe) where membership is defined by a gradual function μ_F on X that ranks on the Real interval $[0, 1]$. Here the degree 0 means the total exclusion while 1 means total inclusion. Elements with membership degree in the open interval (0,1) are said to be in the border of the fuzzy set, they are partially included. As the degree approaches 1, the statement "the element is included in the set" is more certain. Fuzzy sets give a mathematical meaning to vague linguistic terms of natural language. Such meanings are stablished according to the context and human preferences.

The main advantage of fuzzy queries is that they allow expressing user preferences in a natural way, giving discriminated answers. However, upon selecting a membership function for fuzzy sets, it is difficult to apply the suitable quantifier in the natural language. The simple fact of mistakenly specifying any membership function shall cause fails in the result of the query.

It is difficult to compare the two types of databases mentioned above, since they have different objectives. Conventional databases intent to extract new information different from the information already stored through the use of deduction rules, while fuzzy databases intent to store and manage inaccurate or insecure information.

Additionally, both types are not mutually exclusive since they do not allow for making both things, i.e. deduct new information based on inaccurate information.

In connection with the form both software projects are programmed, there may be can be a comparison point, since to implement a Traditional Database it is only required to know the use and management of the communication language with the SQL database, while to program a fuzzy query the programmer has necessarily to know the definition of different fuzzy terms applied to a fuzzy query and the form it is executed by the compiler, and benefit from the database in order to construe the data obtained from the fuzzy query.

A scheme illustrating the execution of a fuzzy query is presented in Figure 1. First, we observe in this scheme that there is a layer where the fuzzy query is written. This is where the SQLf query language [2] is used to specify user preferences through extended Data Definition Language (DDL) with the definition of fuzzy terms and the Data Manipulation Language (DML) extension for query operations (SELECT) with the possibility of expressing fuzzy conditions [12]. Second, we can observe a layer consisting of the fuzzy DBMS. In this work, we use SQLfi which is a logical layer responsible for translating the sentences of the extended language into sentences of the DBMS's native language. In a third layer, the data repository used by the DBMS appears.

There is also a difference form the point of view of users, since for a query in a Traditional Database only the field of the query is requested, while for a query in a fuzzy database, it is required to select query criteria as well as the definition of each criteria, which is more expensive as to the resources and time of the response.

Although a classical query seems to be easier, it does not provide all the results required to meet all our preferences. However, although fuzzy query seems to be more complex, it offers more results that in spite of not being within the range specifically indicated provide us with those that will better cover our requirements.

4. Control and evaluation system for software projects

For showing some benefits offered by fuzzy queries, an experimental application namely SISCEPI was developed. This is a control and evaluation system for software development projects based on fuzzy logic resulting in a tool that allows for evaluating the capacity of managers when developing a project, as well as evaluating the advantages provided by the Fuzzy Database Managers for spreading their use to make them of public knowledge. Additionally, it is a tool that allows and facilitates decision-making upon executing a project since it provides the manager with a set of criteria that will allow for measuring the quality of projects, the proper management of resources used, involved costs, etc

SISCEPI is the Projects management application that allows defining and controlling in a simple way the main elements involved in the administrative monitoring of work, projects, tasks, individuals and expenses. This system was developed at Universidad Simón Bolívar, where the studies on fuzzy managers continue to be performed. For the system development, we used SQLfi, on top of Oracle RDBMS.

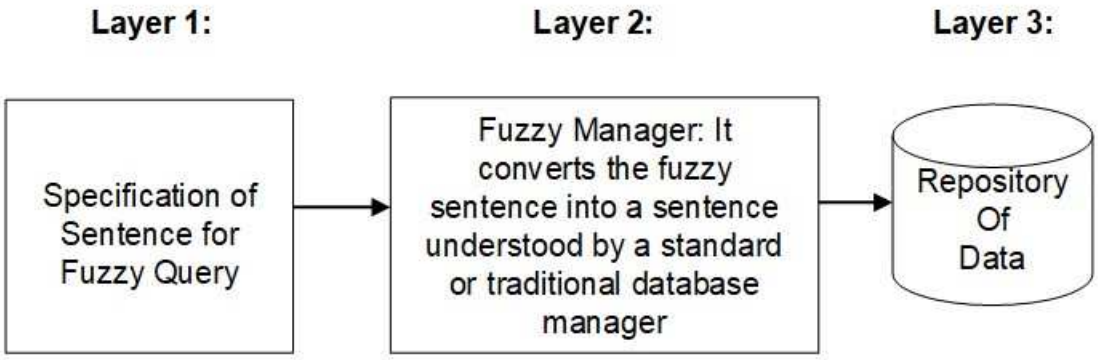

Figure 1: Fuzzy Query Processing

With regards the software developing tool, the Java Multiplatform Language was used. The methodology implemented in the development was the OMT (Object Modeling Technique) [13], due to the advantages offered with respect to the development of the system, since it adapts to current and future requirements related to software engineering.

Certain query evaluation mechanisms were proposed for SQLf. One of them is focused on extending a classical DBMS through a logical layer that makes the processing of fuzzy query on the result of precise query extracting the rows from the database.

The simplest processing mechanism that might be used for fuzzy queries is named naïve strategy. This mechanism consists of calculating the satisfaction degree on each row of the query relation and then selecting the rows desired by the user.

In first place an analysis of the problem domain was performed and a sound architecture for the system was established. Then, a design model was made based on the analysis model, where certain details were added for the subsequent implementation of the system.

The element used to illustrate fuzzy terms is a trapezium, whereby the satisfaction level from 0 to 1 (Axis Y), and the range of the query (Axis X) are shown. For purposes of illustrating this concept, following is the creation of three predicates for the Duration attribute (Figure 2) and three predicates for the Budget attribute (Figure 3), used in SISCEPI.

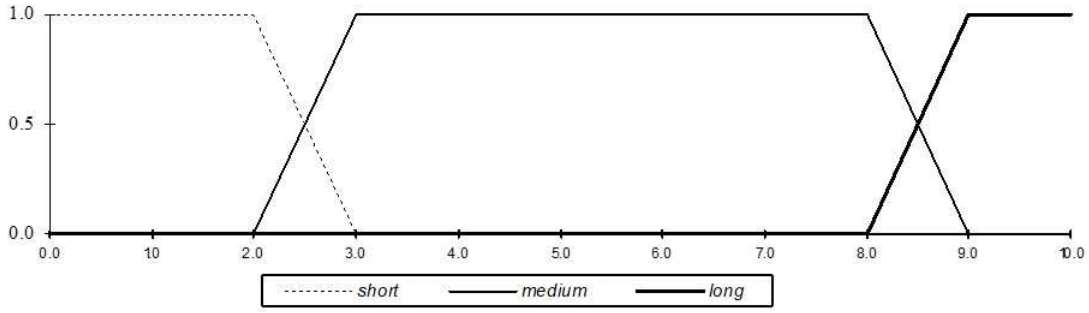

Figure 2: Fuzzy predicates on Duration (U.M: months)

The SQLf-DDL statements for specifying predicates of Duration attribute are:

```
CREATE FUZZY PREDICATE ShortDuration ON (1..10) AS (0,0,2,3)
CREATE FUZZY PREDICATE MeddiumDuration ON (1..10) AS (2,3,8,9)
CREATE FUZZY PREDICATE LongDuration ON (1..10)
AS (8,9,infinit,infinit)
```

In Figure 4, we show a user interface with the steps followed upon making a fuzzy query: First, the attributes that will be part of the fuzzy query are selected. These attributes correspond to the evaluation criteria of the project, namely: duration, size, budget, work team, progress, risk, cost and feasibility. For each these attributes, the user defines fuzzy predicates according to own preferences.

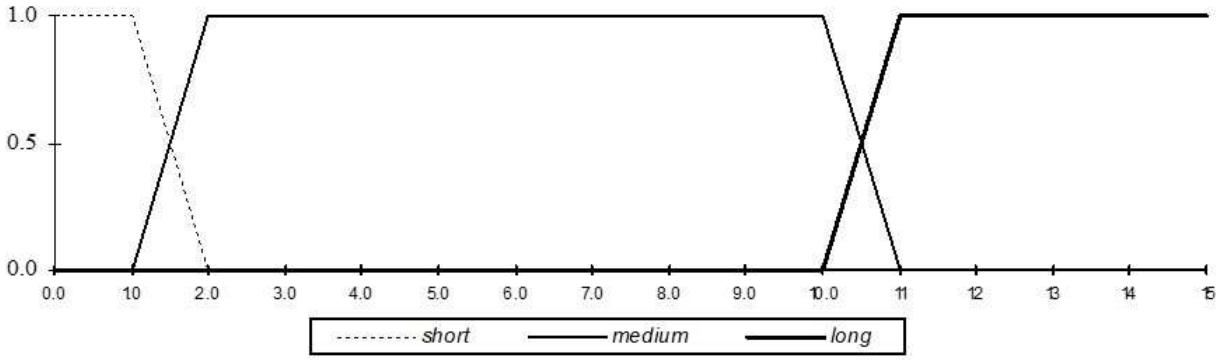

Figure 3: Fuzzy predicates for Budget (U.M:millions)

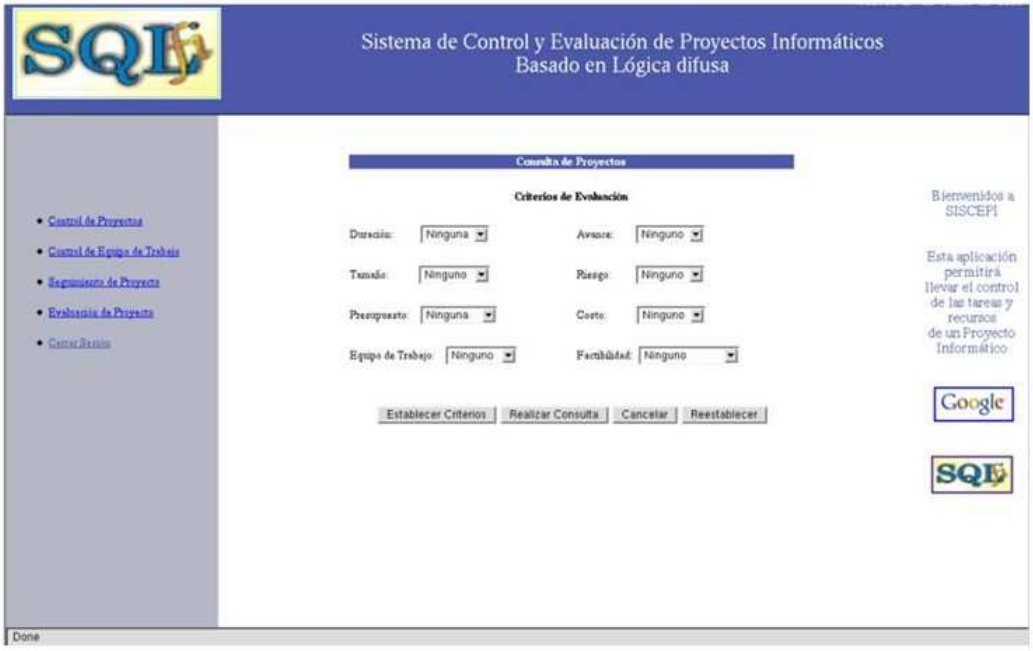

Figure 4: User interface: making a fuzzy query

Figure 5 presents a screen upon making the query. Below you will find certain criteria to make such query, where user selects the predicate of his preference and then establish the meaning of such predicate. In Figure 5, the second step for the construction of the query is shown. On this screen, once the evaluation criteria have been selected, the user defines their preference for each criterion. In this case, an interface is used that allows setting these preferences in an intuitive way. This becomes necessary because the theory of fuzzy sets is little known. The movement of the range bar to the left shows preferences towards the low levels (short predicate), while moving this bar closer to the right indicates a preference towards elevated levels (long predicate). If the bar remains in the middle, a medium level is indicated (medium predicate). The definition of these predicates using the SQLf-DDL is generated automatically by SISCEPI.

Once the user has selected the desired values of the criteria and the calibration, the system generates the fuzzy query considering all these parameters. The calibration corresponds to the level of satisfaction expected in the consultation and indicates the way to choose the best answers. There are two types of calibrations:

- Quantitative Calibration, where the user select the best k responses by means of the satisfaction degree of the query to the established criteria, and
- Qualitative Calibration that indicates the choice of answers according to membership degree thrown by the re-

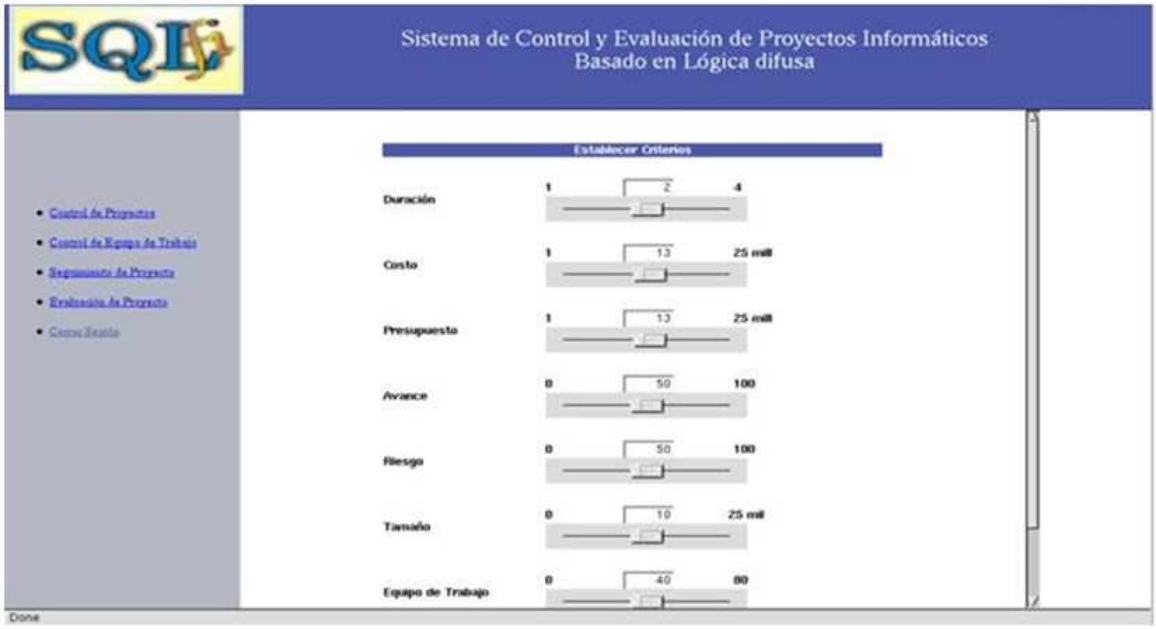

Figure 5: User interface: preferences selection

sponse according to whether it is greater than or equal to a minimum level (α) of expected satisfaction, which is a value in the interval [0, 1].

The generated query has the following format

```
SELECT projectID FROM PROJECTS WHERE
        duration IS preferenceDuration AND
        size IS preferenceSize AND
        budget IS preferenceBudget AND
        workTeam IS preferenceWorkTeam AND
        progress IS referenceProgress AND
        risk IS preferenceRisk AND
        cost IS preferenceCost AND
        feasibility IS preferenceFeasibility
WITH CALIBRATION [k|a|k,a]
```

The constants indicated in italics (preference...) represent the fuzzy predicates that were selected by the user when establishing his preferences on the screen shown in Figure 5. After executing this query in the fuzzy DBMS, the system displays one screen with the list of projects resulting from this query, as shown in Figure 6. The satisfaction degree of each answer to the fuzzy query is included in a column named Precision.

5. Conclusions

Existing tools for Project Management do not respond to some information requests that involve vague criteria expressing user preferences. The added value of the work here supported is the dynamic construction of queries about project control data based on fuzzy logic and natural language. It would be important to mention that fuzzy logic and natural language can be applied in other business areas such as: the identification of traits in very diverse data sets, the population of a country, the ranking of service providers, the evaluation of personnel, among others. In this sense, based on the contribution presented in this paper, similar experiences could be made for these other areas.

Traditional database management systems (DBMS) restrict the treatment of imprecision or uncertainty. A solution that has been proposed for this problem is using fuzzy sets in DBMS to express imperfect data and vague requirements. There are a few of fuzzy set based flexible querying languages, being SQLf one of the most representatives. There is an

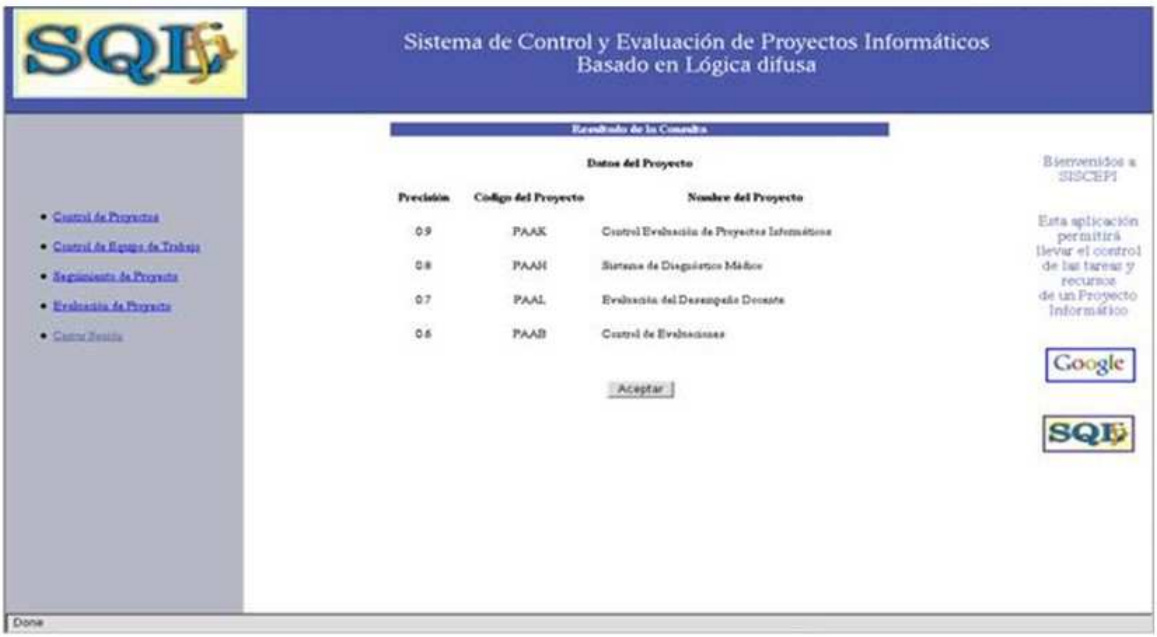

Figure 6: User interface: fuzzy query results

extension of DBMS for fuzzy queries named SQLfi. It is a logical layer on top of a relational DBMS that runs on several actual DBMS such as ORACLE, Postgres, MySQL and others.

An experimental application namely SISCEPI was developed using SQLfi. SISCEPI is control and evaluation system for software development projects based on fuzzy logic. This tool enable the configuration of dynamic queries for pulling out the required information for supporting the managerial decision making processes with regards Project management cost, resources, quality and so forth.

Additionally, the solution provides a graphical end user interface, which build the fuzzy query in back ground mode. The application pulls out answer based on preferences defined by the end-user. Such answers are likely helpful in decision-making support. We have shown here only some on these interfaces, but there are many in similar way for different requirements.

As ever, the management of project system SISCEPI provides all usual functionalities of this kind of applications, such as administration of projects, tasks, resources and expenses; selection of technical resources and leadership; Control and follow-up of work flow; and others likewise. Nevertheless, we do not present them here because they are rather traditional and lack interest for our purpose to illustrate the benefits of fuzzy querying.

Acknowledgements

We give thanks to Eng. Juan Carlos Eduardo, SQLfi program developer who helped us a lot. This work has been made with all the heart: “Whatever you do, work at it with all your heart, as working for the Lord, not for men” Colossians 3:23 (New International Version)

References

[1] E. Cox. Relational database queries using fuzzy logic. *Artificial Intelligent Expert*, 10(1):23–29, 1995.
[2] O. Pivert; P. Bosc. *Fuzzy Preference Queries to Relational Databases*. Imperial College Press, London, 2012.
[3] J. Galindo. Introduction and trends to fuzzy logic and fuzzy databases. in: Handbook of research on fuzzy information processing in databases. In *Prototipo experimental para consultas difusas. Caracas, Venezuela*, pages 174–180. Hershey, PA, USA: Information Science, 2008.
[4] Zongmin Ma: Li Yan. A literature overview of fuzzy conceptual data modeling. *Journal of Information Science and Engineering*, 26(2):427–441, 2010.
[5] A. Aguilera; L. Borjas; R. Rodríguez; L. Tineo. Experiences on fuzzy DBMS: Implementation and use, 2013.

[6] C. Aranda; J. Galindo. Gestión de una agencia de viajes usando bases de datos difusas y FSQL. In *Turismo y tecnologías de la información y las comunicaciones: Nuevas tecnologías y calidad*, pages 65–77. TuriTec'99., 1999. OnLine.
[7] M. Goncalves; L. Tineo. A web tool for web document and data source selection with SQLfi. In *Proc of the 9th International Conference on Enterprise Information Systems*, pages 65–77. ICEIS 2007, 2007.
[8] M. Goncalves; L. Tineo. SQLfi y sus aplicaciones. *Avances en Sistemas e Informática*, 5(2):33–40, 2008.
[9] L. Borjas; A. Fernández; J. Ortiz; L. Tineo. FDBMS application for a survey on educational performance. In *Bhowmick S.S., Küng J., Wagner R. (eds) Database and Expert Systems Applications. DEXA 2008. Lecture Notes in Computer Science*, pages 753–760. Springer, Berlin, Heidelberg, 2008. OnLine.
[10] L. Zadeh. Fuzzy logic - a personal perspective. *Fuzzy Sets and Systems*, 281:4–20, 2015.
[11] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
[12] R. Rodríguez; L. Tineo. Elementos gramaticales y características que determinan aplicaciones con requerimientos difusos. *Revista Tekhné*, pages 50–64, 2009.
[13] J. Rumbaugh; M. Blaha; W. Premerlani; F. Eddy; W. Lorensen. *Object-Oriented Modeling and Design*. Prentice Hall, 1990.

Sobre los autores

Livia Borjas
Ingeniero en Computación. Profesora en el Instituto Universitario de Tecnología Dr. Federico Rivero Palacio, Venezuela. Correo: LivaCaro7@gmail.com - ORCID

Leonid Tineo
Ingeniero en Computación. Magister Scientiarum en Computación. PhD. en Computación. Profesor e Investigador en la Universidad Simón Bolívar, Caracas, Venezuela. Adscrito al Departamento de Computación y Tecnología de la Información. Correo: leonid@usb.ve - ORCID