Implementación de una Ontología y herramienta de consulta para un programa de Maestría en Ciencias de la Computación

Creation of a consulting tool and implementation of an ontology for a Master's Degree Program in Computer Sciences

Cecilia Reyes Peña[1], Mireya Tovar Vidal[1] iD , Concepción Stephanie Vázquez González[1]

[1] Faculty of Computer Sciences, Benemérita Universidad Autónoma de Puebla. Puebla, México
reyesp.cecilia@gmail.com, mireyatovar@gmail.com, stephanie.vazquez@gmail.com

Resumen. En este artículo, es representada una ontología manual para un programa de Maestría en Ciencias de la Computación construida con algunos elementos de las metodologías METHONTOLOGY, Grüninger y Fox, y Bravo. Se ha seguido un conjunto de pasos para identificar y representar la base del conocimiento del programa de Maestría, posteriormente son utilizados axiomas lógicos de primer orden y preguntas de competencia para evaluar la ontología. El desarrollo de un módulo en lenguaje Python es utilizado para la evaluación de la ontología a través de las preguntas de competencia definidas en la fase de diseño. Este módulo es lo suficientemente flexible para presentar preguntas predefinidas o definidas por el usuario en tiempo de ejecución y obtener resultados a las consultas que representan las preguntas de competencia. Elementos como el diagrama de jerarquía de clases y descripción de las relaciones y atributos son utilizados en la construcción de la ontología.

Palabras Clave: Ontología, Herramienta Python, Lenguaje SPARQL.

Abstract. In this paper, a manual ontology for a Computer Sciences Master program constructed, that uses some elements from the METHONTOLOGY, Grüninger and Fox, and Bravo's methodologies, is presented. A series of steps to identify and represent the Master's Degree program's knowledge base has been followed. Afterwards, first order logic axioms and competency questions to evaluate the ontology are used. The development of a module written in Python language is used for evaluating the ontology through competency questions defined during design phase. This module is flexible enough to present predefined or defined questions by the user in running time and to obtain results to the queries representing the competency questions. Elements as a hierarchy class diagram and a description of the relations and attributes are used in this ontology's construction.

Keywords: Ontology, Python tool, SPARQL language.

1. Introduction

These days the amount of information existing in the web grows exponentially, which in turn increases the problem of its organization. A way to reduce such problem is through the usage of the

Semantic Web, in which semantics for both, information and services, is defined, so that it allows the Web to understand and satisfy content requests made not only by users but also by machines [1]. Tim Berners Lee defines it as follows: "Semantic Web is not a separate Web but instead an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [2]. Ontologies are an important cornerstone of Semantic Web, for they provide a well-defined structure in order to present clear and precise information regarding a particular domain. Adolfo Lozano Tello lists the main elements for ontologies in its works [3]:

- Concepts: basic ideas intended to be formalized, *i.e.*, the knowledge about a subject.
- Relations: they represent the interaction and link between the concepts of a certain domain.
- Functions: a relation where an element is obtained through a calculation which involves several elements of the ontology.
- Instances: used to represent particular objects within a concept.
- Axioms: rules or norms declared upon relations in which the elements of the ontology should meet with.

In 2009, a model for ontology evaluation is proposed by E. Ramos, H. Nuñez y R. Casañas; it mentions that the most important aspects of an ontology are: (1) correct and legible (understandable) use of the ontology targeted for every audience language and for each one of the ontology's elements; (2) good taxonomy design (structure); (3) use of a proper coding language; (4) its applicability; etc. [4] Representation of information, regarding educational and academic institutions' information that uses ontologies, has been previously proved to be useful when describing information in an organized and structured way. These institutions' information also possess a well-defined and organized structure [5], [6]. The Semantic Web technologies have also proven to be really useful in the everyday life. When applied in daily life settings, these tools give us direct power when searching better and more accurate results [7].

In this work, the creation and implementation processes of an ontology, in the representation of a Master's Degree program knowledge-base, will be presented. This document uses as its main information source the study areas, professors, administrative personnel, students, and available courses of the master's program, as well as the relation each entity has between each other. For consulting and referencing this ontology, a graphic tool that contains some predefined questions, a console for SPARQL language and a graph about the structure of ontology have been created.

This work is mainly focused on users that do not know what to ask. The flexibility of the ontology allows them to search and navigate through information, without having a precise informational target, thanks to the language in which the ontology is implemented.

This document is organized as follows: Section 2 specifies the ontology design. First, the scenario for the competence area is presented through a taxonomy, so to get to a definition regarding ontology classes and the relations between them. After that, a series of questions expected to be answered through the ontology querying, are also defined. A formal definition is made for the classes and its attributes, as well as, for the description of the ontology's relations and axioms. Section 3 covers the formalization of the competency questions defined in Section 2, plus the results of such queries obtained through the consulting tool and its features. Section 4 presents the conclusions of this research.

2. Ontological implementation

Several methodologies for ontology construction have been defined, as mentioned by Corcho, M. Fernández-López and A. Gomez-Pérez [8]. In this work elements of the approach suggested by Grüninger and Fox [9], as well as, some suggested by the METHONTOLOGY methodology [10] and M. Bravo [6], are used. Grüninger and Fox's method is based on the design of systems with a knowledge base using first order logic. First, the scenario in which the ontology is applicable is defined, followed by the generation in natural language of the so-called "competency questions", whose objective is to determine the ontology reach. These questions and its corresponding answers are used to extract the main

concepts, as well as their relations, properties and axioms within the ontology. The formality of this method based on classical logic allows us to transform informal scenarios into computational models. The elements of METHONTOLOGY borrowed for the design are the following: taxonomy construction for the knowledge-base to be represented; class description tables; attributes and relations within classes; and rules and instance's attributes.

2.1 Scenario presentation

The use of ontologies to represent a knowledge-based within a certain domain has the purpose to facilitate the understanding of such domain, and to obtain better information on the subject. The relevant information about the Master's degree in Computer Science could be represented in a hierarchic diagram generated in the Protégé software [11]. Such diagram presents two existing knowledge generations and their application lines: Software Technologies and Computing Applied to Sciences and Engineering (see Figure 1). These two lines, in turn, comprise two specialization areas each one.
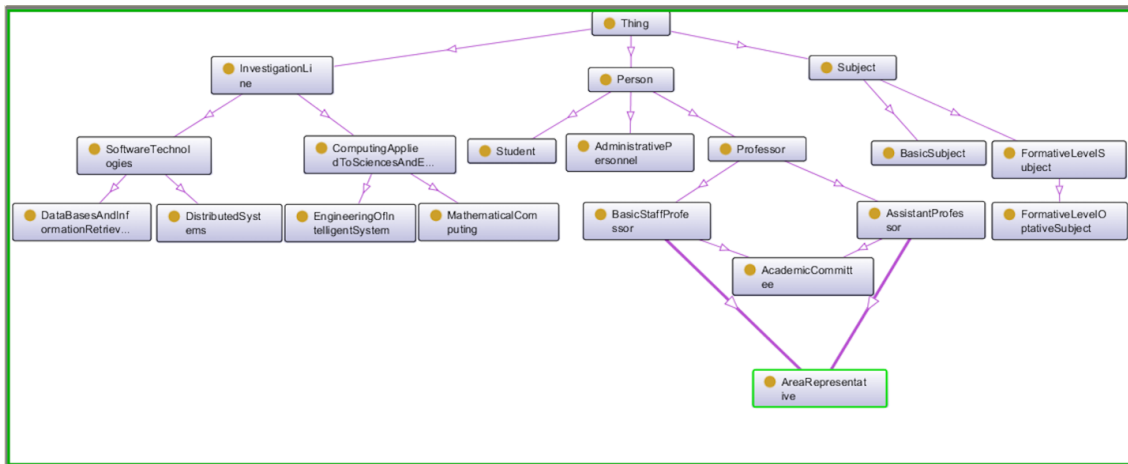
Figure 1. Hierarchy class diagram.

The professors within the Master's program belong to one of the aforementioned areas, and some of them also perform administrative duties such as Graduate School Coordinator, or Secretary of Investigation. Furthermore, there is administrative personnel, which does not make part of the academic personnel. Each student, like professors, belongs to an area; and the courses to be taken correspond to a given area, apart from three first semester courses that are given to all students regardless of their research area. The remaining subjects are part of the formative level within each area. All personnel have a unique ID that identifies them; for both students and academic personnel there is information regarding their highest academic degree obtained (Bachelor, Master, PhD), in addition to their personal data such as name, age and gender.

2.2 Competency questions

Competency questions are an important part in the ontology design steps because they allow defining the domain and scope of the ontology. The proposed ontology looks for answers to similar questions to the following:

1. Which students belong to the Data Bases and Information Retrieval area?
2. Which courses were taught in 2016?
3. To which area does the Graduate School Coordinator belong?
4. How many professors belong to the Software Technologies investigation line?
5. Which professors taught a basic subject?

6. Which professor gives the Data Mining subject?
7. How many administrative workers are there in the program?
8. How many students took the Programming subject?
9. What is the average age of the students in the Master's Degree program?
10. In which period (academic term) the course Information Retrieval was offered?

The ontology knowledge base must be capable to answer such questions. At this phase, the questions are presented in natural language.

2.3 Class definition

The following entities were found after an analysis of the scenario from the competency area (see Table 1). A mixed strategy was used (top-down and bottom-up) to identify the main concepts [10].

Table 1. Classes' definition.

Class	Definition
Investigation Line	This class groups the two main investigation lines into the Master's program.
Software Technologies	Subclass of Investigation Line, represents a line of generation and application of knowledge.
Computing Applied to Sciences and Engineering	Subclass of Investigation Line, represents a line of generation and application of knowledge.
Data Bases and Information Retrieval	Subclass of Software Technologies, represents a specialization area.
Distributed Systems	Subclass of Software Technologies, represents a specialization area.
Engineering of Intelligent Systems	Subclass of Computing Applied to Sciences and Engineering, represents a specialization area.
Mathematical Computing	Subclass of Computing Applied to Sciences and Engineering, represents a specialization area.
Person	This class includes all the different individuals that belong to the Master's program.
Professor	Subclass of Person, groups the academic personnel working in the Master's program.
Basic Staff Professor	Subclass of Professor, the personnel that belongs to some investigation area.
Assistant Professor	Subclass of Professor, it is the personnel that do not belong to some investigation area.
Academic Comitee	Subclass of Basic Staff Professor and Assistant Professor, in charge or the student admission program.
Area Representative	Subclass of Basic Staff Professor and Assistant Professor, in charge of solving area related problems.
Administrative Personnel	Subclass of Person, groups all the people with an administrative position, they can be also a Professor at the same time.

Class	Definition
Student	Subclass of Person, contains the students into the Master's program.
Subject	This class groups all the subjects taught at the Master's program.
Basic Subject	Subclass of Subject, groups all the subjects that are taught to all the students regardless of their area.
Formative Level Subject	Subclass of Subject, groups all the subjects that are taught to all the students according to their area.
Formative Level Optative Subject	Subclass of Formative Level Subject, groups all the subjects that are taught to all the students according their area, but the content of the course is chosen by the corresponding academy.

2.4 Relations and attributes description

Within the ontology, there are certain regarding the instances of the ontology itself. To begin to describe these constraints it is necessary to consider the necessary objects' attributes to complement the information (See Table 2).
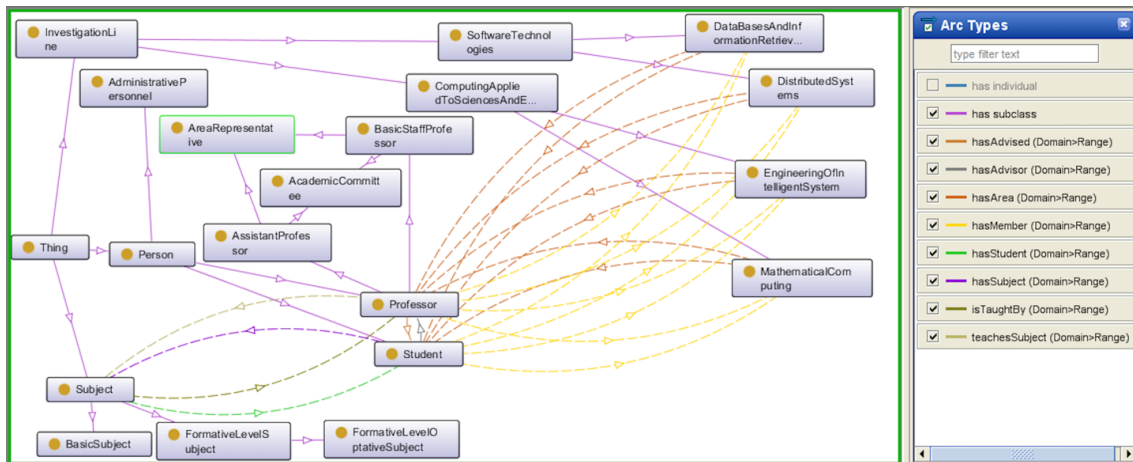
Table 2. Objects' attribute description.

Attribute	Class	Type	Range	Cardinality
hasName	Person, Subject	String	Unbounded	1:1
hasDegree	Person	String	Unbounded	1:1
hasAge	Student	Integer	Positive Integer	1:1
hasEmail	Person	String	Unbounded	1:N
hasPeriod	Subject	String	Spring, Fall	1:1
taughtInYear	Subject	Integer	Positive Integer	1:1
hasGeneration	Student	Integer	Positive Integer	1:1
hasPosition	Administrative Personnel	String	Unbounded	1:N

Just like objects have attributes, the classes have relations with each other. The following table describes on detail the relations showed before (See Table 3). These relations can be visually observed in the following diagram, the relations can be represented as a graph where each class is presented as a node, and the edges between nodes are the so-called relations between classes (see Figure 2).

Table 3. Classes' relation description.

Relation	Domain	Range	Cardinality
hasMember	Data Bases and Information Retrieval, Distributed Systems, Engineering of Intelligent Systems, Mathematical Computing	Student, Professor	1:N
hasArea	Student, Professor	Data Bases and Information Retrieval, Distributed Systems, Engineering of Intelligent Systems, Mathematical Computing	1:1
hasSubject	Student	Subject	1:N
hasStudent	Subject	Student	1:N
isTaughtBy	Subject	Professor	1:1
teachesSubject	Professor	Subject	1:1
hasAdvisor	Student	Professor	1:1
hasAdvised	Professor	Student	1:1

Figure 2. Relations between classes diagram.

The axioms defining the rules for the ontology are listed in the following table (See Table 4).

Table 4. First Order Logic Axioms of the Ontology.

| Name	Description	Expression	Class or Relation
Sole Role	A student cannot be a professor at the same time	NOT[EXIST(Student(?x) AND Professor(?x))]	Student, Professor
Sole Contract	A basic staff professor cannot be an assistant professor at the same time	NOT[EXIST (BasicStaffProfessor(?x) AND AssistantProfessor(?x))]	Basic Staff Professor, Assistant Professor
Sole Subject	A subject can be taught just once time in a unique period and year	IF(Subject(?x), Subject(?y) AND ?x.hasName=? y.hasName) THEN (? x.hasPeriod!=? y.hasPeriod Or ?x.taughtInYear!=? y.taughtInYear)	Subject
Related Tutorial	A professor can be advisor just to a student from the same area	IF(Professor(?X), Student(?Y) AND has Advisor (?Y,?X) THEN (?x.hasArea=? y.hasArea)	Professor, Student

3. Results

This section first presents the formalization of the competence questions formulated in section 2.2. These have been formalized in natural language, since they are going to be presented in such way to the user of the ontology (See Table 5).

**Table 5.** Competency questions formalization.

Formalization in Natural Language
Which students belong to X area?
Which courses were taught in X year?
To w*hich* area does the administrative personnel and professors belong?
How many professors belong to X investigation line?
Which professors taught a basic subject?
Which professor gives X subject?
How many administrative personnel there are in the program?
How many students took X subject?
What is the average age of some kind of person in the Master's program?
At *which* period the course X was offered?

In order to validate the correct construction of the manual ontology, it is necessary to answer the previous questions and to verify the veracity of those results. A consulting tool (*Ontholopy*) was made using the Python programming language in order to provide to the user a useful and easy-to-use tool for such queries. The ontology is fed to the consulting tool using a RDF file [12]. The consulting tool has a well-defined list of the main competency questions. They are presented to the user using natural language for a better and easier understanding, moreover, the consulting tool also provides a way to introduce new questions made by the user. These questions are formalized in SPARQL language and introduced to the tool to obtain the related answer. SPARQL is a standardized language for RDF queries, normalized by the World Wide Web Consortium (W3C) [13]. Once the consulting tool is initialized a screen like the following is presented to the user (see Figure 3).
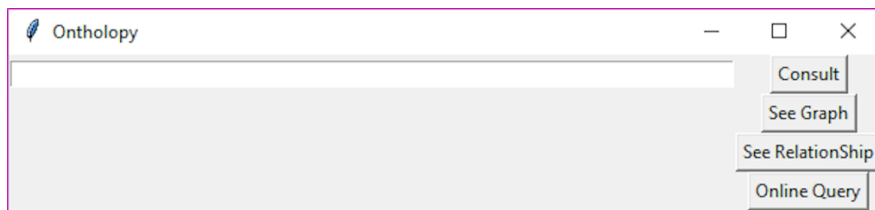
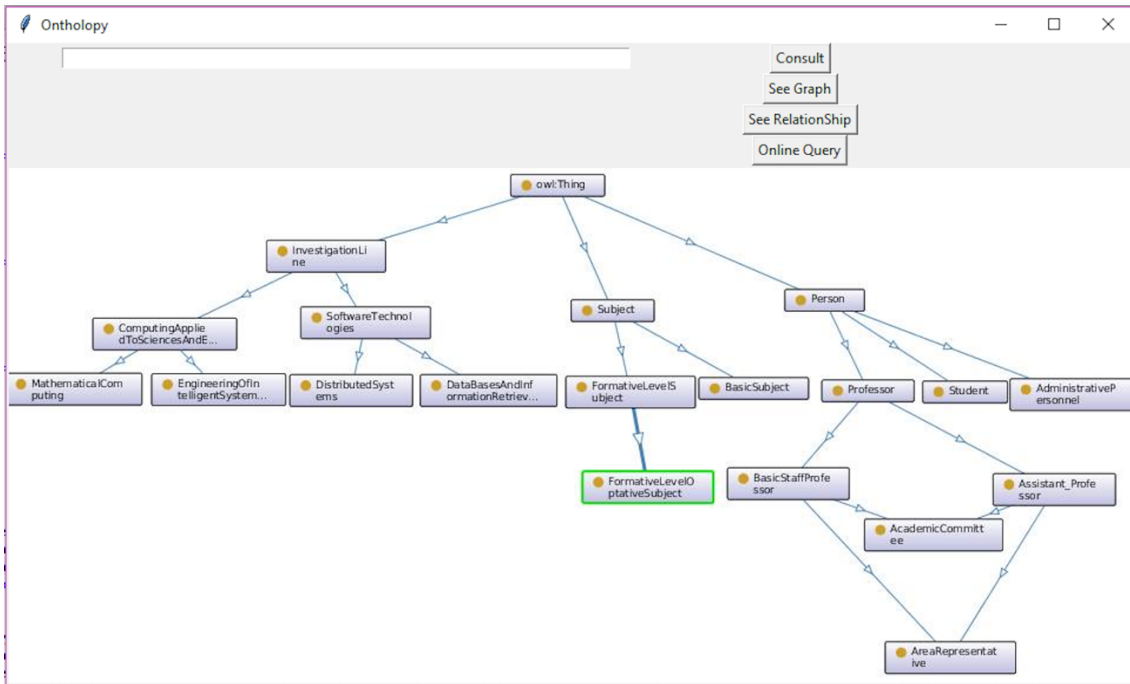**Figure 3.** Consulting Tool GUI.

This window contains the list of the well-defined competency questions, the different options can be browsed using the arrow keys ↑ or ↓. Once the desired question is selected, it is necessary to click over the "Consult" (Do Query) button and the correspondent answer will be displayed (see Figure 4).
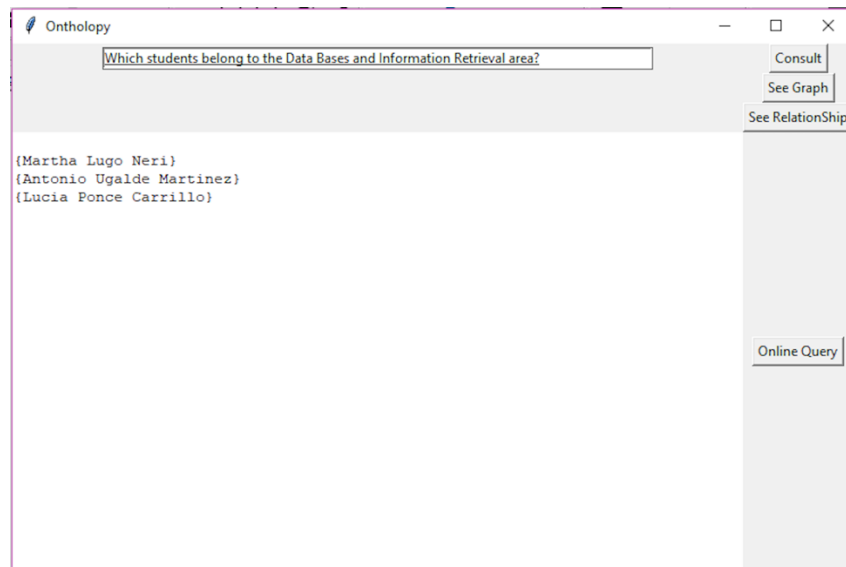
Figure 4. Answer to the competency question is displayed.

It must be highlighted that the tool also includes a button named "See Graph". This element shows a graph that corresponds to the knowledge-base of the ontology, where the subjects and objects are represented by nodes and its hierarchy is represented by the edges of the graph (see Figure 5).

Figure 5. Ontology's corresponding graph.

When the user defines their own questions the way to introduce them to the consulting tool is through the "Online Query" option. When this option is selected, a new blank text box is showed. In this text box the user can type in SPARQL language the new query (see Figure 6).

Figure 6. Customized query for the ontology.

4. Conclusions

Ontologies are powerful tools for information description. Using them to model the information of any structured environment, allows us to answer questions related to the competence area, and to infer knowledge that will also be important for the users of said ontology. Furthermore, the integration of such data with content for Semantic Web, can bring new benefits from intelligent agents, and this information can be accessible for a larger number of users globally.

References

[1] K. Loudon, *Developing Large Web Applications: Producing Code That Can Grow and Thrive*, First Edit. O'Reilly Media, Inc., 2010.

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Sci. Am.*, vol. 284, no. 5, pp. 34–43, 2001.

[3] A. Lozano Tello, "Ontologías en la Web semántica," in *I Jornadas de Ingeniería Web'01*, 2001, p. 4.

[4] E. Ramos, H. Núñez, and R. Casañas, "Esquema para evaluar ontologías únicas para un dominio de conocimiento," *Enlace*, vol. 6, no. 1, pp. 57–71, 2009.

[5] J. A. Baez and M. Tovar, "Un modelo ontológico para representar la organización de una unidad educativa," in *Avances recientes en Ciencias Computacionales -CiComp 2016*, 2016, pp. 54–61.

[6] M. Bravo, F. Martínez-Reyes, and J. Rodríguez, "Representation of an Academic and Institutional Context Using Ontologies," *Res. Comput. Sci.*, vol. 87, pp. 9–17, 2014.

[7] L. Feigenbaum, I. Herman, T. Hongsermeier, E. Neumann, and S. Stephens, "The Semantic Web in Action," *Sci. Am.*, vol. 297, no. 6, pp. 90–97, 2007.

[8] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, "Methodologies, tools and languages for building ontologies. Where is their meeting point?," *Data Knowl. Eng.*, vol. 46, no. 1, pp. 41–64, 2003.

[9] M. Grüninger and M. S. Fox, *Methodology for the design and evaluation of ontologies*. 1995.

[10] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer, 2006.

[11] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, Jun. 2015.

[12] Python, "Python Software Foundation. Python Language Reference, version 3.6.1," 2017. [Online]. Available: http://www.python.org.

[13] W. W. W. Consortium, "SPARQL 1.1 Query Language, 2013.," 2013. [Online]. Available: https://www.w3.org/TR/sparql11-query.

About the authors

Cecilia Reyes Peña
Current student of the Master in Computer Sciences at the Benemérita Universidad Autónoma de Puebla and specializing in Data Bases and Information Retrieval Area. Puebla, México.

Mireya Tovar Vidal
SNI I researcher in the Faculty of Computer Sciences at the Benemérita Universidad Autónoma de Puebla. Puebla, México.

Stephanie Vázquez González
Current student of the Master in Computer Sciences at the Benemérita Universidad Autónoma de Puebla and specializing in Data Bases and Information Retrieval Area. Puebla, México.