**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Electricidad y Computación**

**Artificial Intelligence**

System for recommending the best Ecuadorian tourist route from Guayaquil to Quito, based on the shortest route

July 14, 2017

**Group #2: Joe E. Cruz, Wellington A. Martinez, Erick J. Rocafuerte**

**1. INTRODUCTION**

The most important Highway in Ecuador, is the route between Guayaquil, the city with the highest density and Quito the capital of the Ecuador, and as on any other highway exist accidents, and many other disadvantages.

This problem can affect the tourism and the communication between these cities.

The solution proposes the use of an Expert System with the aid of the hill climbing algorithm to avoid the routes when has natural incidents.

It’s important to say that the hill climbing algorithm uses local results to take a decision and when the algorithm don’t find a better result, the execution of the program end.

**2. THE REQUIREMENT ANALYSIS**

The route Guayaquil Quito has at least four ways to arrive to Quito, and on the way has much intermediate options to select, but only two highways are selected by the drivers, those are the Panamerican Highway, and the route by Santo Domingo de los Tsachilas.

Other issue is the high accident rate, this is the summarize of a lot problems on the way instead in cars and drivers.

The accident and its values of damage are the result of many factors as climate, state of the way, historical data about the accidents in the way, reckless drivers, so we did an investigation to experts, as Agencia Nacional de Transito ANT, INHAMI, OIAT.

At the beginning of the execution the program must be set a collection of accidents in the way between two cities, and then the program establish a new route between Guayaquil and Quito, underestimate the way who has the event.

Then the program will show the result of the execution and the steps to select that option in graphical format.

**3. PROBLEM**

We are developing Expert System for recommending the best Ecuadorian tourist route from Guayaquil to Quito, based on the shortest route, the problem is to obtain the shortest route based on the local facts of each city to be visited, will implement the Hill Climbing algorithm using local maxima, ie will see the local conditions of each city in order to determine what is the next Better state, this way when searching the local maximum we will find the shortest route. We will need the help of an expert to determine under which conditions or rules will be chosen which is the shortest route, as these do not necessarily depend on the distances between cities but also on the local conditions of these, factors such as mudslides, heavy rains, broken bridges, strikes, road accidents. That could make that there is no route by certain ways. The problem in itself seeks to find the shortest route between Guayaquil and Quito, this despite the difficulties that arise between the different cities.

**4. GENERAL FEASIBILITY ANALYSIS**

**Analysis of the tool**

Jess has a big advantage over the others, this is the possibility to integrate to an IDE, this for educational purposes is important for fast development applications.

The integration to the IDE include check of mistakes, typing errors and suggest of Jess Function.

Other advantages is the use of a JAVA language to implement more modules around the Expert System, for example the adquisicion module and the result module.

The Jess languaje has the same modules of any expert system, the adquisicion module, resul module, engine and knowledge database.

**5. ACQUISITION OF PRELIMINARY KNOWLEDGE AND METHODOLOGY**

Our system that we will develop is intended to replicate the knowledge of an expert, by means of rules that allow us to choose the shortest route between Guayaquil and Quito, based on the data of the local conditions of roads between the cities to visit, for these we will use Jess In order to execute the rule engine, in turn the Hill Climbing algorithm will be implemented to find the local maxims, applying the rules obtained above will be able to obtain the shortest route according to the local conditions, in order to prove the correct execution of the System will make several iterations of the algorithm, in each interaction will change the facts or local conditions of some cities and the algorithm will give us as a conclusion the shortest route, in turn will develop a conclusion module where the route will be shown and the reasons by which These decisions were taken and that conclusion was reached.

**HILL CLIMBING:** The AI ​​techniques we’ll use are the Hill Climbing search algorithm, which is a search algorithm that will give us the best route, which uses a heuristic function to determine the next best state. The algorithm works like this: the first closer node is chosen, whereas in steeper climbs all successors are compared and the one closest to the solution is chosen. The algorithm will not necessarily find the global maximum, but it can converge to a local maximum. One of the characteristics of the algorithm is that it will end when there are no improvements, that is to say, this problem will stop when one of the roads between the cities that it has to visit is hampered by some of the events such as mudslides, heavy rains, broken bridges , Strikes, road accidents, which affect roads between cities.

* Uses a breeding technique iterative
* Beginning from a state (point Current) in the search space
* If the new state is better, Transforms into the current state , if not, Another neighbor state is selected and Evaluated.
* The method ends when there is no Improvements, or when a predefined number of iterations.

**Developing:** A graphical interface will be implemented using the Java programming language, in order to maintain interaction with the user. This interface will have a connection to a database with all the cities that are villages between Guayaquil and Quito, and with the routes that exist between it, a list of events that could be given to the execution of the program, these events could do That the paths between cities are closed and there are no roads, so that the algorithm we use will stop and show the local maximum that found, then we will show the scheme that we will use:

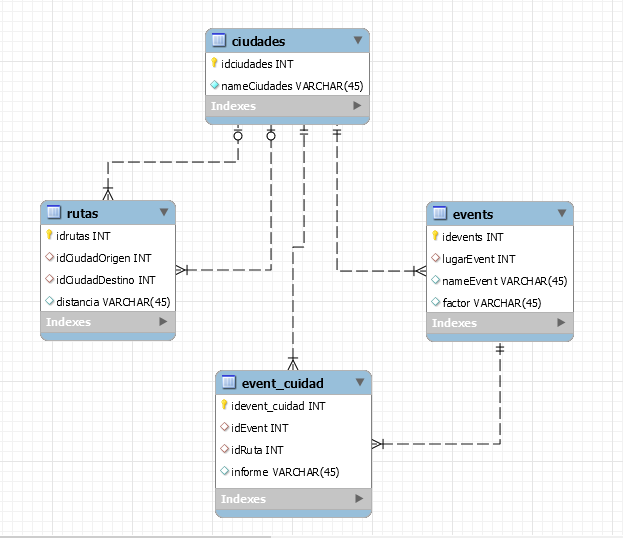


Fig 1.

Ciudades only have the name of itself, the class rutas, has the origin and finish city, the distance between both cities, the name of the route and a little description.

When the routes be ready are join in event\_ciudad this is the result of associate the routes with its possibility of have an accident.

As we are developing an intelligent system, that is, an expert system, rules based on expert recommendation will be generated. The rules engine of Jess will use these rules to give us a recommendation on which is the next best state, this next best state among the Neighboring cities is the one that will need the hillclimbing algorithm to give us local maxima. These rules are based on a procedural system of If, Then. These rules consider the different initial events to the trip and the events that occur during the trip to determine a recommendation of which is the best state between the neighboring cities.

Will be shown below an event diagram that will allow us to understand the execution of the expert system:

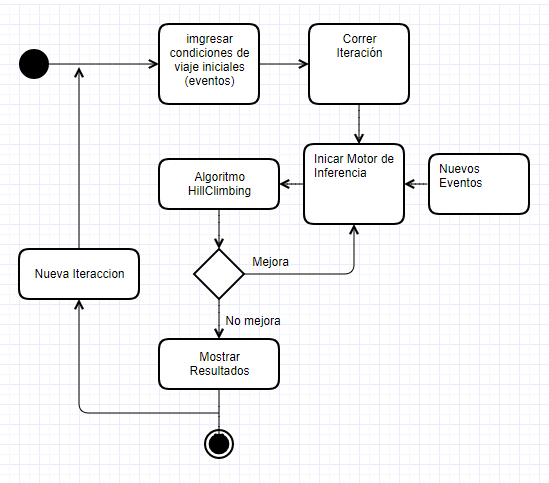


Fig 2..

**6.1 IMPLEMENTATION**

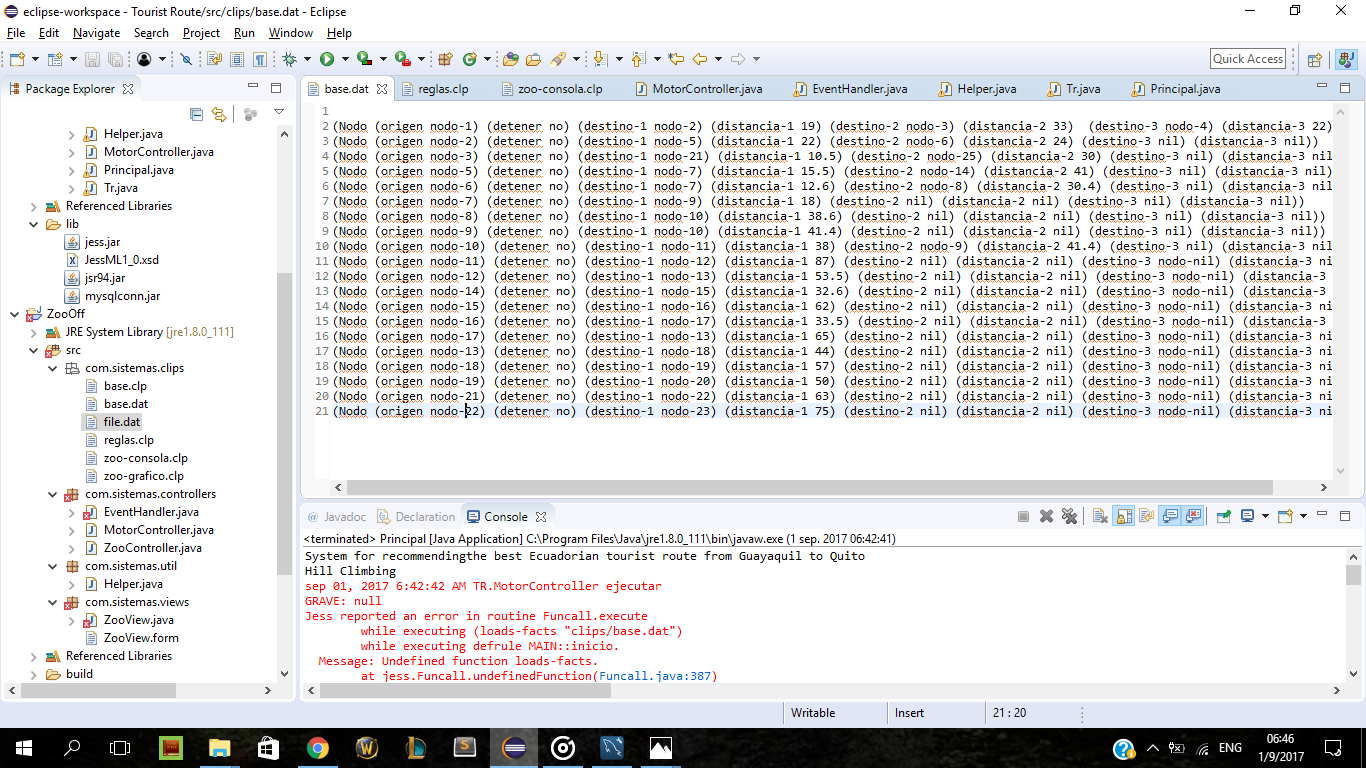
**Stages:**

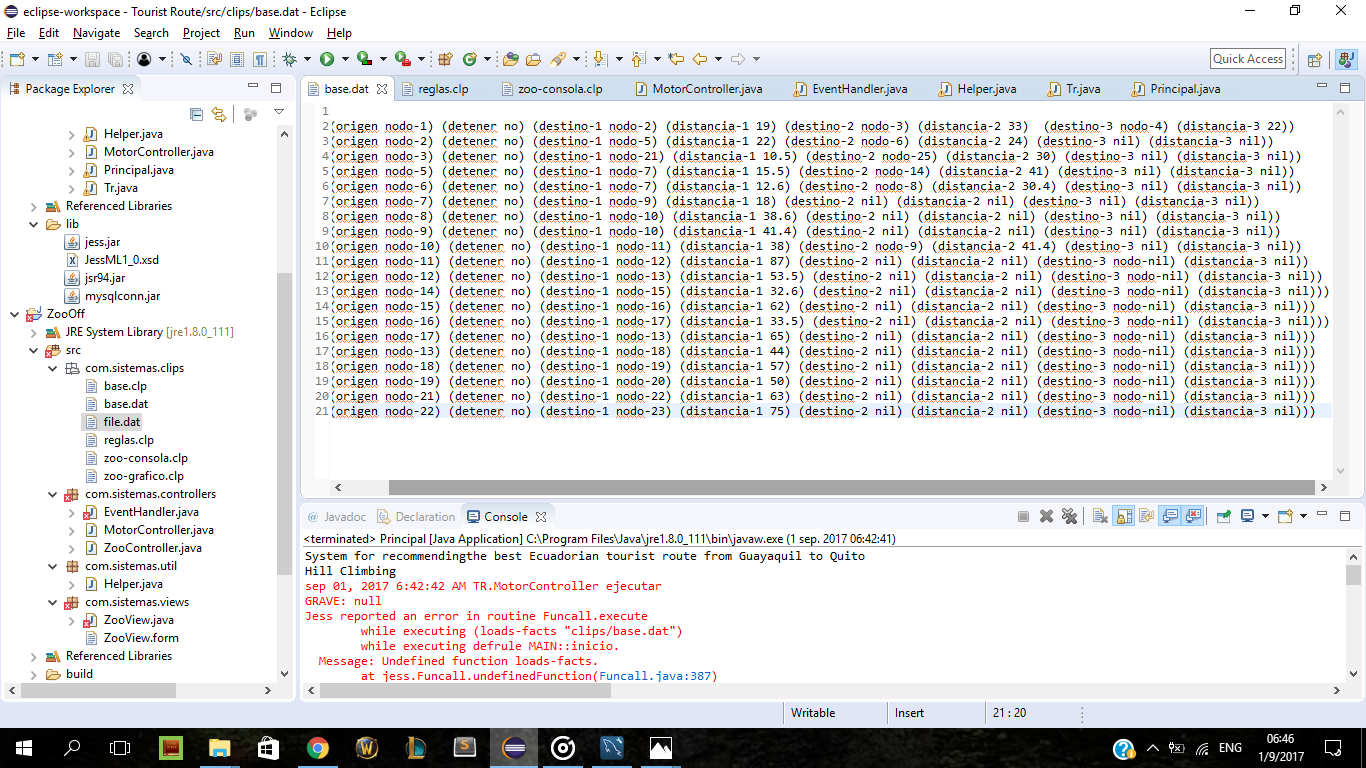
**6.1.1. Knowledge Base**

* the name of the cities.
* The distance between the adjacent cities that comprise the route.
* Information provided by an expert, based on certain rules, to infer knowledge.
* Information obtained from the ANT about traffic accidents.
* Information obtained from INAMHI about the weather of those routes.

**6.1.2. Database**

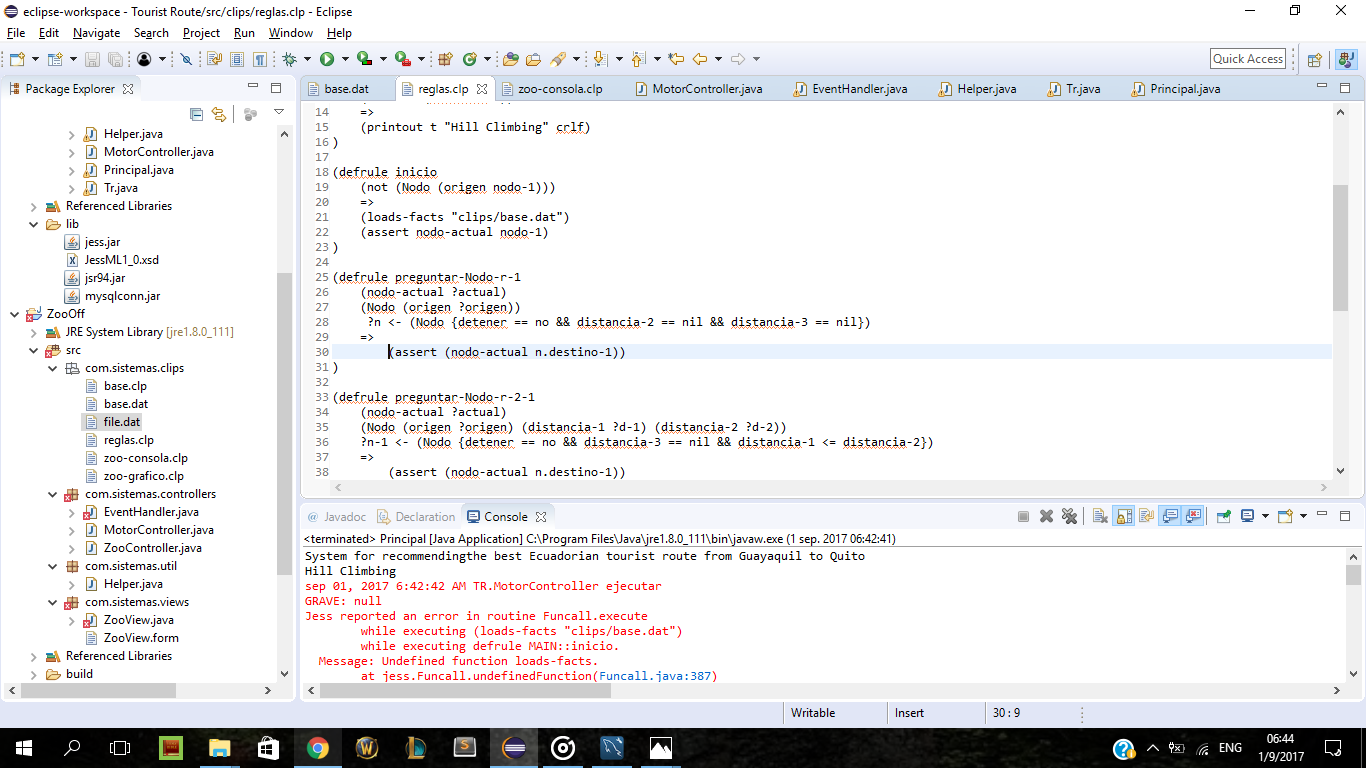
We use MySQL to storage the data provided by the Knowledge Base, because is common to use in projects using JAVA, and for the easy compatibility between both because are made for the same company Oracle.

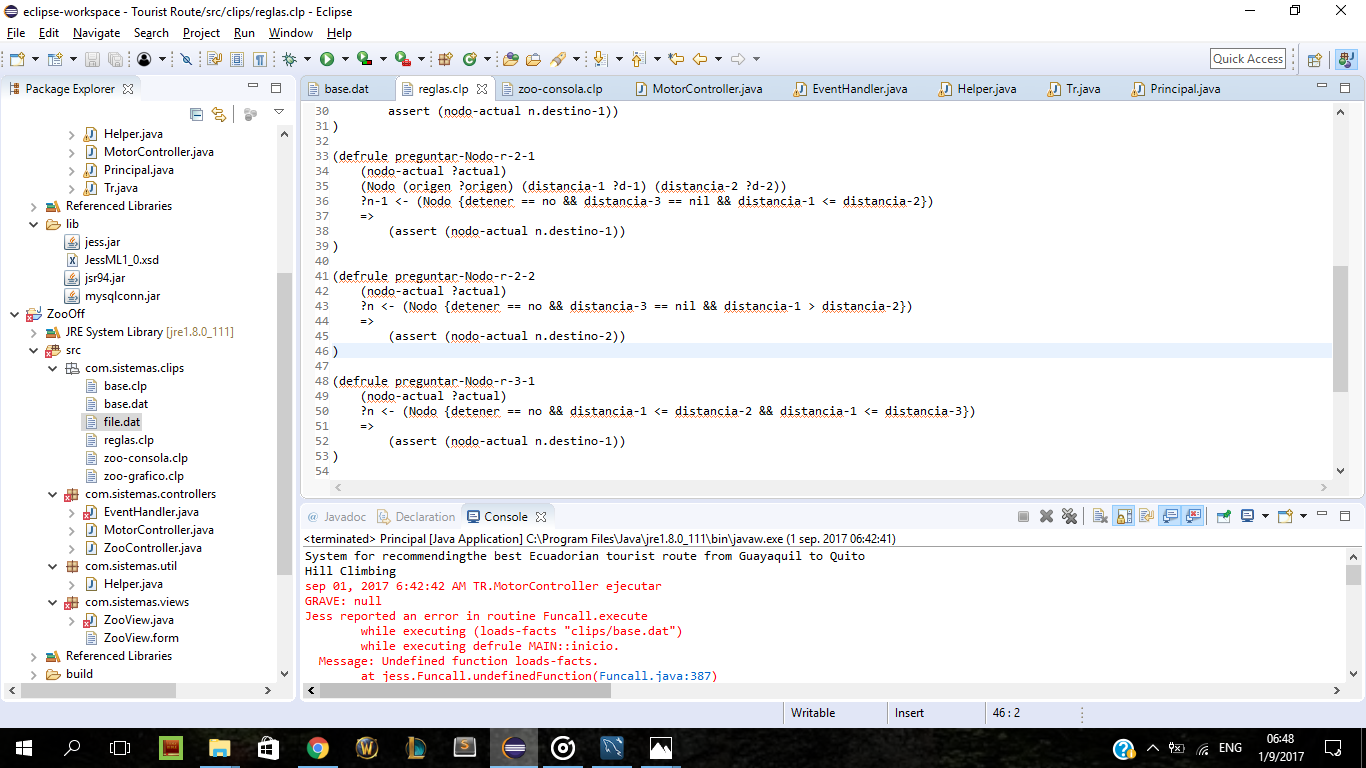


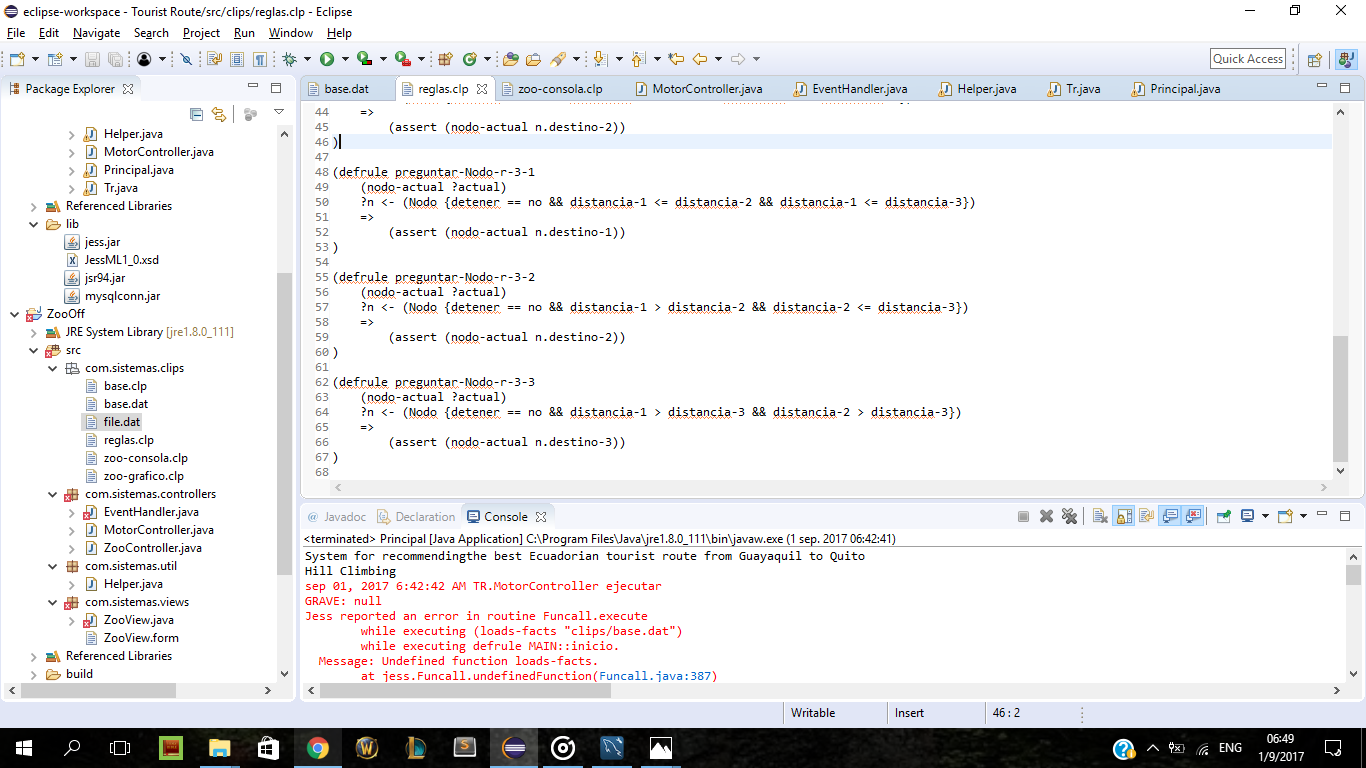


**6.1.3. Inference Engine**

The inference engine use for many expert systems include Jess is RETE algorithm, it uses tuples of facts and rules, if the facts match with the set of rules the program take a decision given by the rule.







**6.1.4. User Interface**

Using the advantages of JESS, we will use the GUI graphic user interface provided by java.

**6.1.5. Conclusions Module**

It shows the shortest route to the user, and in case it requires it will show he how obtain this conclusion.

**6.2 TOOLS**

* We will use this tools:
* Java Virtual Machine
* Java Development Kit
* Jess v7.2
* MySQL Server (SQLite)

**7. PROTOTYPE DESCRIPTION**

**8. CONCLUSION**

* The importance of learned at least one Functional language.
* New options to develop a different kind of system using new techniques.
* Interoperability between a Functional Language and a Structured Language
* Use of knowledge in Artificial Intelligence specifically Expert System in a common system.

**9. REFERENCES**

1. Welcome to Common-Lisp.net!. (2017). Common-lisp.net. Retrieved 14 July 2017, from <https://common-lisp.net/>
2. Welcome to Common-Lisp.net!. (2017). Common-lisp.net. Retrieved 14 July 2017, from <https://common-lisp.net/>
3. Friedman-Hill, E. (2017). *Jess® The Rule Engine for the Java™ Platform* (7th ed., pp. 3-15). New Mexico: Sandia National Laboratories.
4. *MySQL :: MySQL Connector/J 5.1 Developer Guide :: 6.1 Connecting to MySQL Using the JDBC DriverManager Interface*. (2017). *Dev.mysql.com*. Retrieved 8 July 2017, from https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html