

Final Project

Programming languages

By

Jorge Armando Guzmán Flores

A01208480

Chabot developed in prolog to know the proposals of the candidates for the presidency of Mexico 2018

Index

Solution.……………………………… 3

Problematic………………………….. 3

Results……………………………….. 7

Set up instructions…………………... 7

Conclusions…………………………....7

Reference…………………………….. 9

Problematic

The elections to elect the president of Mexico will take place on July 1, 2018, this entails a great responsibility in all Mexicans to elect the best candidate, since he will represent the country for the next 6 years. Mexico has a very broad history of oppression and corruption to the people, so it is important to be informed about candidates for presidency and seek the common good.

One of the biggest problems is the number of votes registered in the country compared to the population there are, currently there are approximately 87 million Mexicans of whom not all vote, in the last votes only 36 million Mexicans voted. The problem is how they get resources for the campaigns, multiplies the total number of citizens and multiplies by 65% of the minimum wage, hence 30% is distributed equally and the rest is given as a percentage of the number of votes of each party, each vote costs the parties so it is important that citizens are aware of who will be our representatives, what are their proposals and so finally be able to decide who is our vote, as they vote without knowing or to know the proposals of the candidates is to vote blindly and worsen the situation of the country.

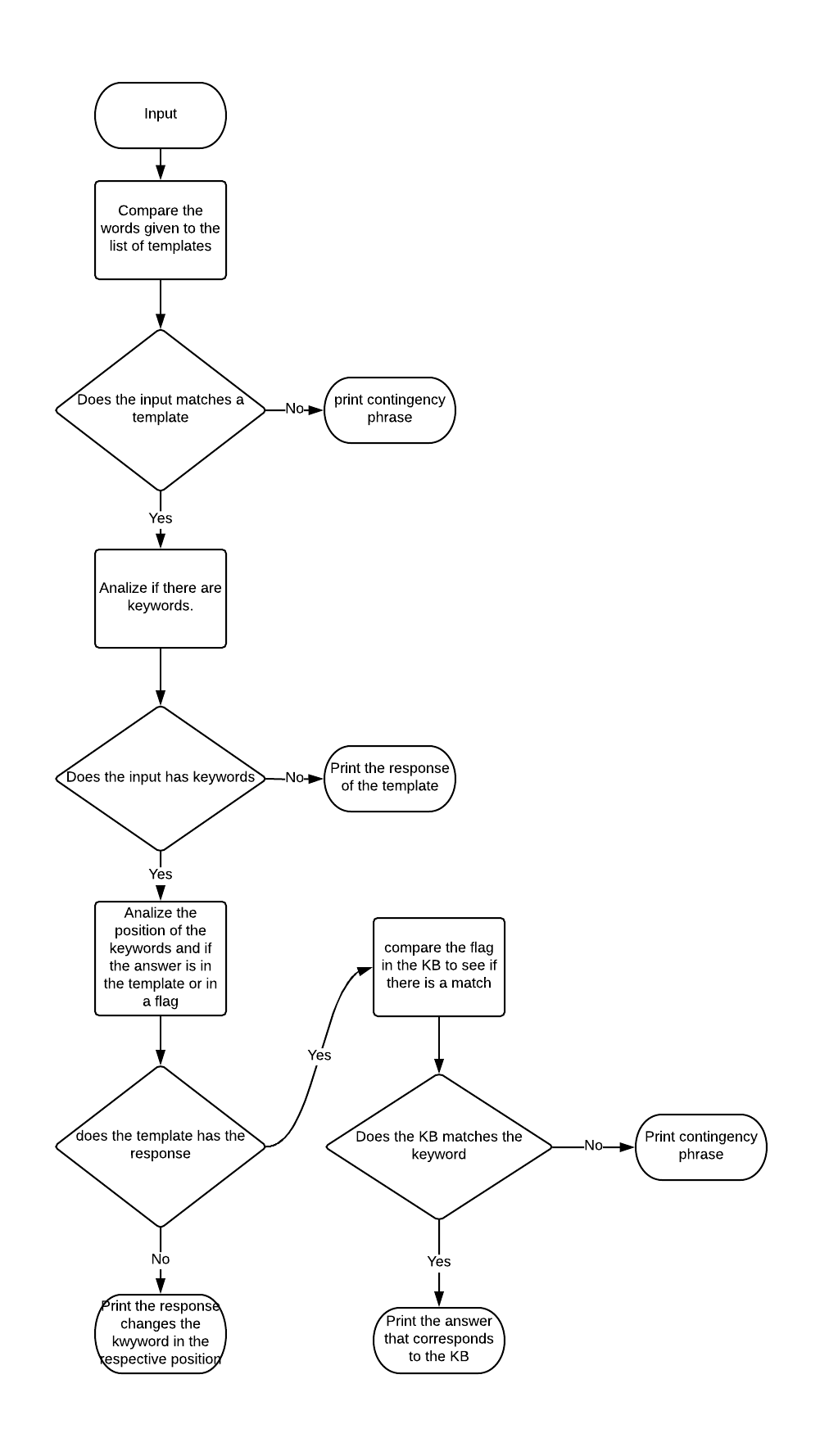
Solution

A solution that I propose to be able to help the Mexican population to know who to vote for is to generate a chatbot that is able to tell you the proposals of the candidates separated by the most common sectors, so whoever decides to use this program will be able to know them without having to read the files that send to the INE with more than 60 pages, to carry out this solution it will be necessary to read these documents, remove the proposals and categorize them in sectors so that the user can obtain the information he desires in a more dynamic and fast way.

To fulfill this objective, the program will be done in prolog, the reason of this decision is due to the paradigm seen in class "logic paradigm". The logical programming revolves around the concept of predicate, or relationship between elements. The logic that is sought in the system is that given a number of key words generates a response to those words belong, if requested information of a candidate, the system will be able to take the name of the candidate, which is our predicate , and look for the elements that relate to this in order to generate an output with the desired response.

The basis for the chat to work by generating answers that match what the user asks is based on compare word by word entered against a base of responses according to each input. The keywords allow you to search for matches in the different lists and, if necessary, change these words by the location where the answer carries the keyword.

The following diagram explains the flow of the system.



The function that compares the input with the templates is match.

In the stimulus the key words are in the predicates, the rest of the words are only atomic terms, when a non-atomic data is found the atom function ignores and continues the comparison of the words, if a template is found that is the same , the function will return the answer and a list with the indexes of the keywords, if any.

The substitute function is the one that cracks the magic in the system, its task is to change the keywords in the corresponding positions of the answers, in this way there is congruence in the context.

That is not all this function also detects the flags, answers based on the KB of proposals, and will look for the functions that belong to that flag to be able to compare the keywords with the KB and thus be able to generate a response based on the KB or in a contingency response.

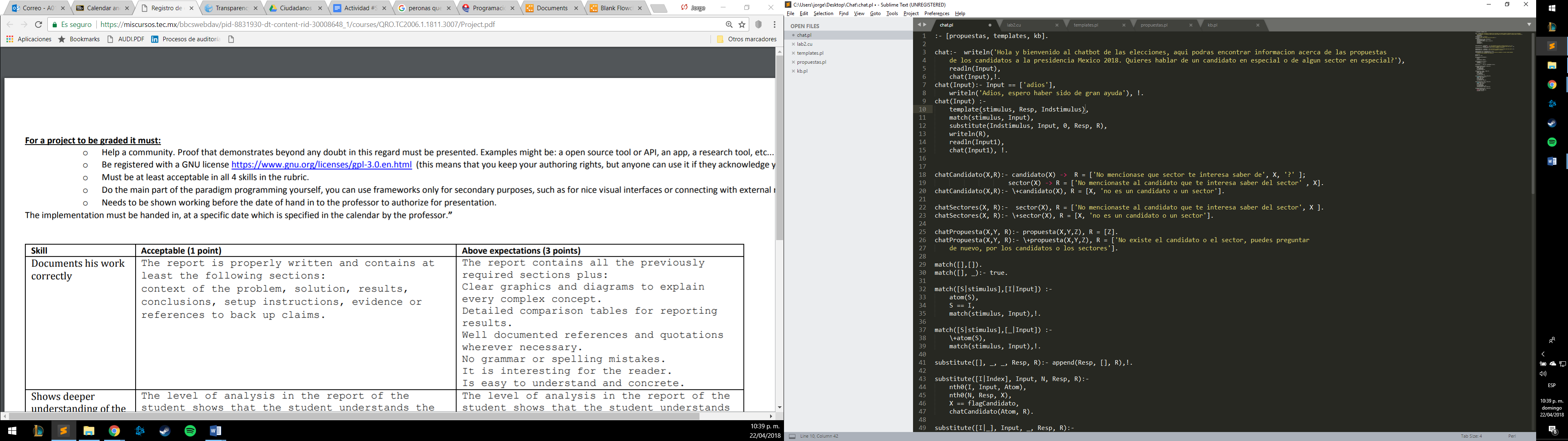
It is important to mention that this function also uses prolog functions, they are mentioned below:

nth0: returns the element of a list at a given position.

select: replaces the value of an item in one list with another based on its values.

append: concatenates two lists. In this case, the words of the answer are concatenated, along with the word replaced in the position where there is a number, word by word.

Finally, the following code shows how is the process of interaction between the user and the system:



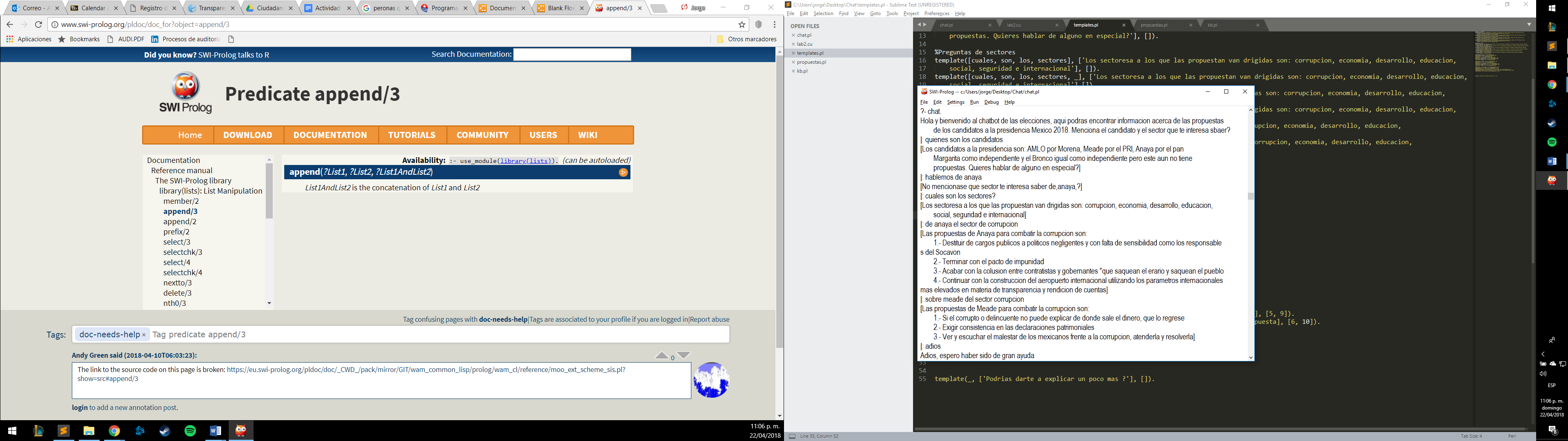
The "readln" function converts the user's queries into a list. Then the program will choose the first template of the knowledge base, then the stimulus of that template, and the user's query (Input), are entered the "match" function. If the stimulus does not match the query, Prolog will start the backtracking, choosing the following template and repeating the operation. The last template of the knowledge base makes the "match" function return true if none of the other stimuli of the other templates matches the user's query. This prevents the Eliza loop from breaking and ends the program abruptly.

Set up instructions

<https://www.dropbox.com/sh/bze007uss3b6zgd/AACY7BdhNY0CKya16VgtiqZZa?dl=0>

Results

If the installation process is followed correctly you can have a conversation as shown below:



Conclusions

The implementation of this chat shows the apparent intelligence of a system layers to maintain a conversation but really everything is a logical system that works through the coincidence of the user's queries and the stimuli is done term by term, all this thanks to the logical programming, which allows us to work around predicates and thus be able to generate logical systems, although the way to learn a new paradigm is not easy, the fact of creating projects of this complexity helps us to understand more and more that the way to program It is important to select the correct paradigm for better efficiency. This system can be greatly improved if the necessary time is dedicated and more templates are generated that help us to cover all the possibilities of answering each question.

I believe that what has been done is enough to understand and learn the basic concept of a logical system that is able to work through keywords to generate predetermined answers or questions, I would like to be able to carry out projects of this type later on, where the impact may be greater, since now I see the endless things that can be done with the logical paradigm.

Reference

<http://www.politify.mx/infografias/propuestas-de-los-candidatos-la-presidencia-de-mexico-2018>

<https://polemon.mx/las-diez-propuestas-de-amlo-para-cambiar-a-mexico>

<http://www.margaritazavala.com/el-mexico-que-queremos/>

<https://www.ricardoanaya.com.mx/plataforma-del-frente>

<http://meade18.com/?gclid=Cj0KCQjwh7zWBRCiARIsAId9b4rd_qZ_SToL-fbQZEE9aoYACYofawgFsPT5kuoPfuShrAnpLKx3XW8aAt8SEALw_wcB>

Diversos acuerdos del Instituto Nacional Electoral. 2017 2018. México.

• Álvarez Arredondo, Ricardo. El sistema de partidos políticos en México. Un esquema residual del autoritarismo. H. Cámara de Diputados XLI LegislaturaMiguel Ángel Porrúa. México, 2012

<http://www.swi-prolog.org/pldoc/man?predicate=atomic/1>

<http://www.swi-prolog.org/pldoc/man?predicate=nth0/3>

<http://www.swi-prolog.org/pldoc/man?predicate=select/3>

http://www.swi-prolog.org/pldoc/doc\_for?object=append/3

Propuestas

<https://www.buzzfeed.com/yuririaavila/estas-son-las-propuestas-de-los-candidatos-presidenciales?utm_term=.ek1PP4Kn6P#.akjggLJVog>

<http://repositoriodocumental.ine.mx/xmlui/bitstream/handle/123456789/94386/CGex201801-5-rp-unico-a2.pdf>

<http://repositoriodocumental.ine.mx/xmlui/bitstream/handle/123456789/94343/CG2ex201712-22-rp-5.1-a2.pdf>

<http://repositoriodocumental.ine.mx/xmlui/bitstream/handle/123456789/94367/CG2ex201712-22-rp-5-2-a2.pdf>

<http://www.margaritazavala.com/el-mexico-que-queremos/>