

Chapter 5 Wizard based on natural language processing for Java programming language

Capítulo 5 Asistente basado en procesamiento del lenguaje natural para lenguaje de programación Java

LÓPEZ-GONZÁLEZ, Erika†*, PONCE-CAMACHO, Julio, MALDONADO-GARCÍA, Ángel and REYES-NAVA, Adriana

Tecnológico Nacional de México: Tecnológico de Estudios Superiores de Jocotitlán, División de Ingeniería en Sistemas Computacionales, Carretera Toluca-Atlacomulco km 44.8, Ejido de San Juan y San Agustín, Jocotitlán, Edo.

ID 1st Author: *Erika, López-González* / **ORC ID:** 0000-0001-7279-5111

ID 1st Co-author: *Julio, Ponce-Camacho*

ID 2nd Co-author: *Angel Maldonado-García*

ID 3rd Co-author: *Adriana, Reyes-Nava* / **ORC ID:** 0000-0002-4440-909X

DOI: 10.35429/H.2023.13.53.68

E. López, J. Ponce, A. Maldonado and A. Reyes

*erika.lopez@tesjo.edu.mx

A. Reyes, E. López and B. Hernández (AA. VV.) Computer Technology and Innovation. Handbooks-TI-©ECORFAN-Mexico, Mexico City, 2023

Abstract

Smart assistants are a technology that has become very popular today, due to the multiple functions they have, and somehow allow a natural interaction between devices and human beings. The objective is to develop an assistant based on NLP (Natural Language Processing) focused on the Java programming language, with the feature of guiding the user on the use of this language from reliable and validated sources, through speech recognition and speech synthesis by the assistant, allowing communication between the software and the user through natural language, in order to make the time in the consultation of information more efficient since it is considered that from the implementation of the speech to text engine and the subsequent processing of natural language, The search will be carried out on the web obtaining relevant results of precision. In the implemented methodology, the general requirements were defined, such as the voice recognition and identification modules, the search module and the internal storage structure, the design was based on a file-based controller view architecture, implemented through technologies such as PyQt5, Speech Recognition, Pyttsx3, Beautiful Soap among others, developed with the python programming language with the paradigm of object-oriented programming obtaining satisfactory results of precision in the searches by concepts and syntax carried out.

Virtual Assistant, Speech recognition, Paradigm, Web scraping

Resumen

Los asistentes inteligentes son una tecnología que en la actualidad se han vuelto muy populares, debido a las múltiples funciones con las que cuentan, y de algún modo permiten una interacción natural entre los dispositivos y los seres humanos. El objetivo es desarrollar un asistente basado en PLN (Procesamiento de lenguaje natural) enfocado al lenguaje de programación Java, contando con la característica de orientar al usuario sobre el uso de este lenguaje a partir de fuentes confiables y validadas, a través del reconocimiento de voz y síntesis de voz por parte del asistente, permitiendo la comunicación entre el software y el usuario a través de lenguaje natural, con el fin de hacer más eficiente el tiempo en la consulta de información ya que se considera que a partir de la implementación del motor de voz a texto y el posterior procesamiento del lenguaje natural, se realizara la búsqueda en la web obteniendo resultados relevantes de precisión. En la metodología implementada se definieron los requisitos generales como lo son los módulos de reconocimiento e identificación de voz, el módulo de búsqueda y la estructura interna de almacenamiento, para después realizar el diseño que es con base a una arquitectura modelo vista controlador basada en archivos, implementándose a través de tecnologías como PyQt5, Speech Recognition, Pyttsx3, Beautiful Soap entre otras, desarrollado con el lenguaje de programación python con el paradigma de programación orientada a objetos obteniendo resultados de 92% de precisión en las búsquedas por conceptos y sintaxis realizadas.

Asistente virtual, Reconocimiento de voz, Paradigma, Web scraping

1. Introduction

Smart assistants are a technology that can be implemented in many areas, from the development of an assistant with a particular function, to assistants with the purpose of meeting the general needs of a user. This technology has been present since 1960, which is why various works have emerged that have increased its development.

Education is a very important factor in the development of a person, therefore, the relationship between technology and education are two aspects that should be handled together for a good performance in learning various subjects. As mentioned in (Ssanyu, 2023), there are a wide variety of educational tools and several of them deal with aspects of language processors, which can be used according to the needs and interpretation of the results, in the case of the aforementioned quote he carried out a research focused on PAMOJA which is a framework of components based on Java and supports the construction of grammatical applications.

The objective of developing this research project is to implement an assistant based on NLP (Natural Language Processing) dedicated to learning the Java programming language, with the characteristic of guiding the user on the use of this language from reliable and validated sources, promoting the bases for a good development in learning through speech recognition and speech synthesis by the assistant. allowing communication between the software and the user through natural language.

It is considered that the topic of intelligent assistants is of great relevance today, and probably will have an even greater impact in the future, and being part of that development is what motivates the realization of this research, in order to implement the creation of this type of software.

Various contributions have been found regarding the aforementioned topic and each project has contributed in the field of virtual assistants, mentioned some in the year 2017 article by Arteaga Maza *et al* (Maza A., 2017), presents the implementation of a tutor assistant based on cognitive computing for use in visual learning environments, in order to improve the knowledge of students, resolving doubts. The project had only one interface developed through the Node.js framework, with the conversation service "IMB Watson", with keyboard requests, which reduced a little the efficiency of querying the information but fulfilled correctly based on the subject.

The field of assistants is not only in academics or personal help, it also covers more areas, an example is the assistant project implemented on a Raspberry Pi 2, with the ability to perform tasks that a user requests with the use of their own voice, obtaining a response to the request and executing an action such as turning on lights or devices (Díaz Fernández, 2018). The project was developed from a Python library "Speech Recognition", with which speech recognition was achieved.

The accuracy of this wizard was 88.66% according to the accuracy tests in recognizing commands, however, the requests it executed were very limited. As already mentioned, virtual assistants have many applications as mentioned in "Voice application, as a virtual assistant for the management of emotions in students of UT Tehuacán", a project that helped to identify emotions with the help of color codes where each one symbolizes a domain or kind of feeling, using voice commands. In this project, multiple tests were carried out on the Alexa Developer Console platform where results obtained were satisfactory, I also seek to contribute with an alternative tool for early detection of emotions, through its voice application information was provided to the interested party without having a physical approach (ORTEGA-GINES, 2019).

There have been projects focused on a specific sector, where the beneficiaries are a small region, as an example is the chatbot-type virtual assistant. In this research work for the degree, a chatbot-type virtual assistant was developed, having the characteristic of receiving requests through the keyboard, with the purpose of speeding up the reception of a request from PQR's (The System of Requests, Complaints, Claims and Suggestions) in the San Isidro corporation "Colegio Anglo Americano". This is done through a web administration module for user interaction, with the only shortcoming being limited word identification (García-Reina, 2018).

Voice recognition is a technology that speeds up the execution of various actions, an example of this is the thesis work "Voice Recognition Security System" written by the authors Eyra Pérez & Fernando Martínez *et al* (Pérez, 2013). In this dissertation, a security system was developed with the characteristic of the use of voice recognition so that the system can interact in a more fluid and precise way, in order to make fast and reliable decisions.

The design and implementation of this system was carried out MATLAB with algorithms to obtain audio input. An important factor to avoid errors was to be in an environment of 70 dB (Decibels), with the problem of lack of speech recognition when changing emotions as it had an error rate of 40%. Voice programming is also a potentially useful method for people with motor disabilities, as mentioned in (Nowrin, 2023), and is also challenging for a standard speech recognizer due to the variability in spoken programs.

Virtual assistants are intelligent tools that help users search for information in a conglomerate of web resources. The natural deployment of these is carried out on the web pages themselves, where they allow users to solve their doubts formulated in natural language using Artificial Intelligence techniques such as Natural Language Processing (NLP) (Jorge Matich, 2001). Also as mentioned in (Camacho-Alvarez, 2020), Natural Language Processing (NLP) has had enormous growth thanks to the large amount of data that can be obtained through the internet. NLP investigates the use of computers to process or understand natural language (LN) for the purpose of performing useful tasks.

Likewise, natural language processing is the ability of a machine to process communicated information through the use of natural language. It could be said that NLP consists of using a natural expression that can communicate with the computer directly, by written means or voice command, facilitating commands or requests with language (Moreira, 2020), in this way speech recognition is simply speech recognition allows a human being to communicate with a computer. Broadly speaking, the computer captures the voice signal emitted by a person through a microphone, converting it into digital information (Sánchez. G.A., 2021). To do this, it is important to identify Speech Synthesis, also known as text-to-speech (CTV), which consists of providing the system with the ability to convert a given text into speech.

The voice of the computer can be generated by joining the recordings that have been made, whether they are whole words, or phonemes, but always trying to make the sound produced seem as natural and intelligible as possible, correctly chaining the sounds within the speech (Cuentas Chavez, 2020), now the files will allow storage, so to speak, since a file is a set of related information, recorded in the secondary storage system and referred to by a name, finally, web scraping refers to the techniques that allow data to be collected or obtained from the web through automatic or manual methods.

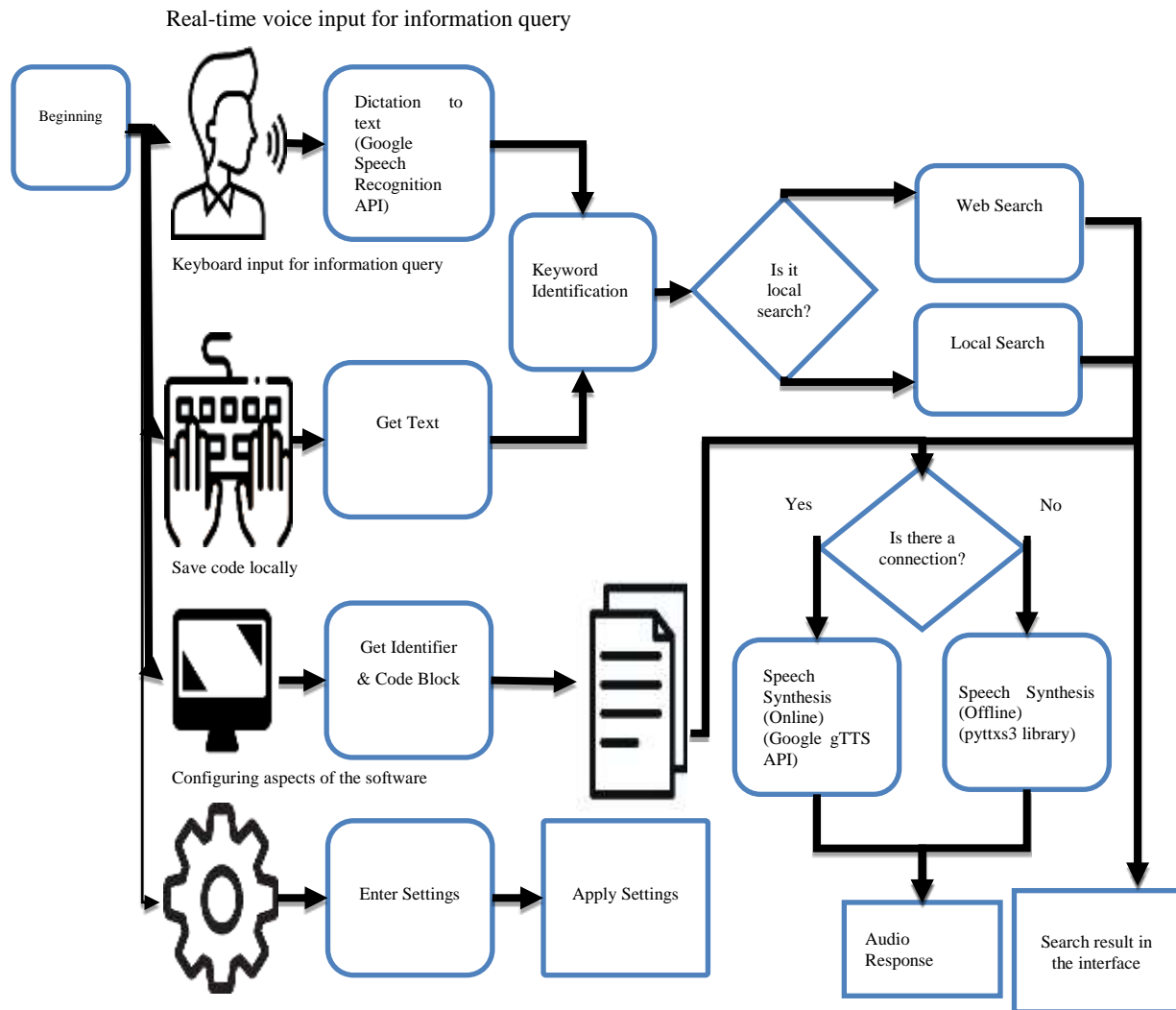
With web scraping, you can extract the content of an HTML from websites to filter the information and store it to later make use of the data of interest (Gonzalez. D, 2013). Some other variants of language processing can also be found, such as word processing, in the work (GARCÍA-AMARO, 2017) "Review of text preprocessing techniques for the automatic classification of tweets in Spanish " where techniques were applied to obtain text in tweets, which made it possible to classify them more easily to detect patterns and keywords. Likewise, the cleaning and normalization techniques reviewed in this research helped them to standardize and refine the content of each tweet.

2. Work Architecture

The architecture that will be implemented to carry out this project is based on the agile software methodology, SCRUM, because the field of development of this project requires a lot of programming, this methodology is ideal, taking into account its particularity of implementation of a development divided by "Sprint", which is equivalent to a certain amount of time in which it proposes to implement specific aspects. The methodology includes the planning stage where various activities will be developed that will serve as a guide for the beginning and culmination of the project, defining the time for the beginning of the development.

Determination of general requirements of the software, which is a fundamental aspect, since based on the product, the planning of each of the Sprints can be carried out. Sprint planning will allow you to achieve the objectives, and each sprint will trigger the implementation of the project, in which important activities are carried out that increase the overall progress of the project. In the design phase, the logical operation of the software is proposed, determining how the flow, the technical, logical and visual design will be. In the implementation, the coding of the software is carried out through the technologies and algorithms designed. Finally, the testing, documentation and evaluation of the software

Based on the requirements, the logical operation of the wizard was precisely defined, which has several modules that allow it to fulfill the purpose of assistance for the java language, which is shown in Figure 1.

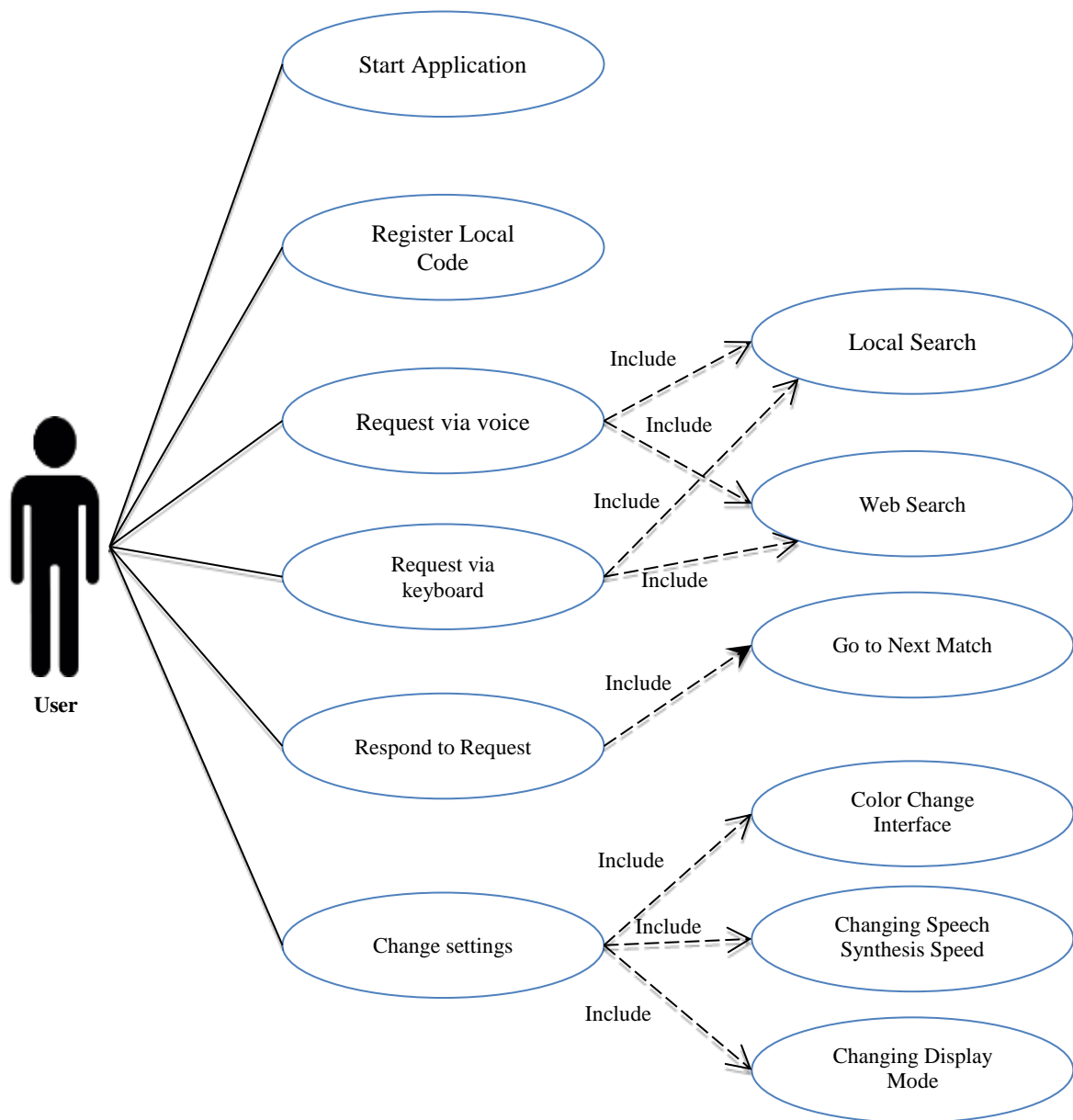
Figure 1 Logical Design

Source: Own Work

As can be seen in figure 1, you can see the diagram that represents the execution of the wizard, having different paths to follow depending on what the user decides to do.

In total there are 4 modules: voice dictation to carry out a query, whether web or local, consultation through the keyboard that shares instructions with the previous module; the local code storage module, in which based on an identified and a block of code can be stored in the local code file for quick queries, as well as the configuration module, in which the user simply determines parameters of the software of his preference and applies them.

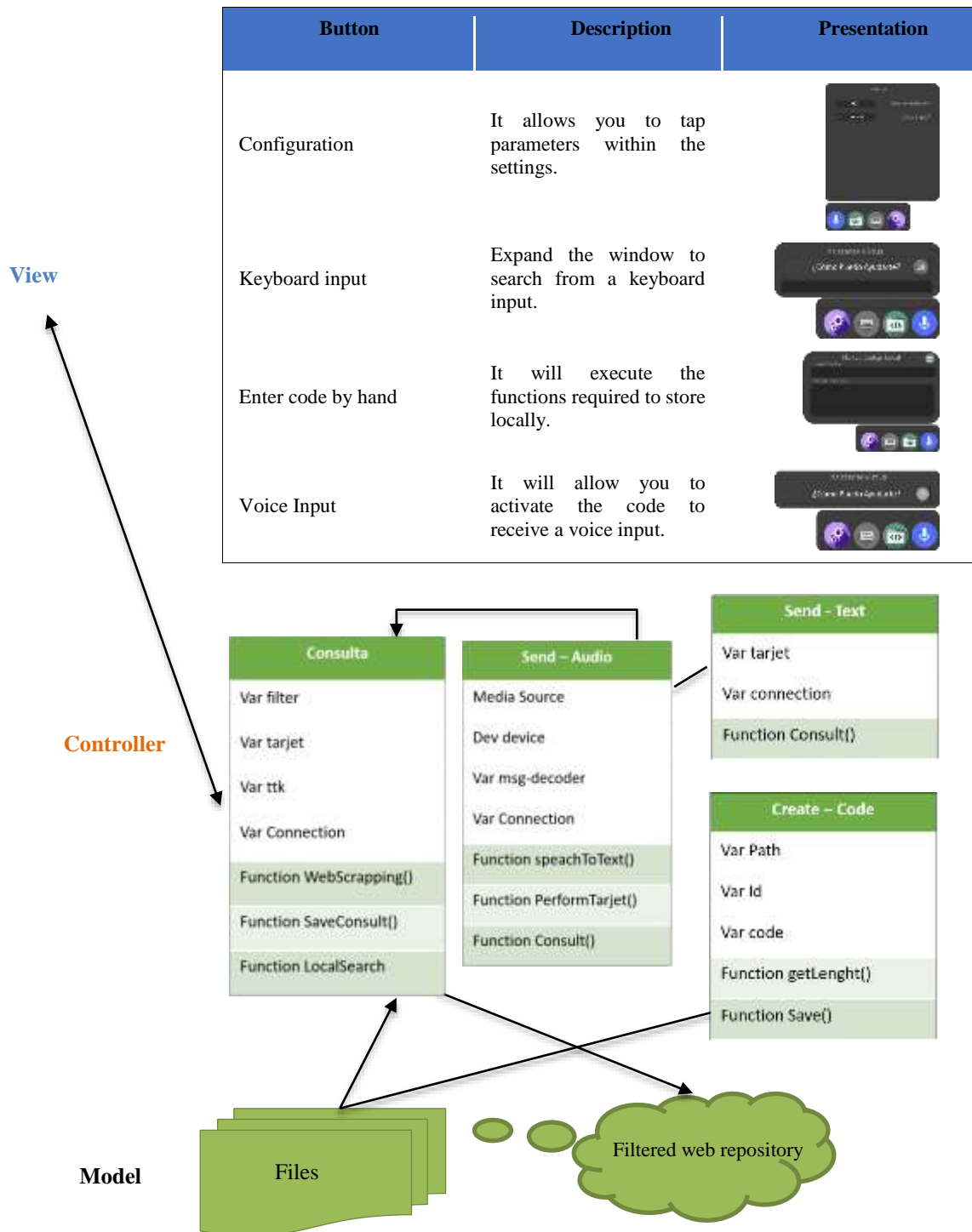
In figure 2 it can be seen that the software will not only present type of use, since among the interfaces it will allow to perform various types of actions with which the user can interact, so it can be seen that the main use cases are the use of local code registration, web and local search by code or by concept, as well as the configuration of software parameters.

Figure 2 Designing Use Cases

Source: Own Work

The model view controller is a design pattern that allows you to separate the database (Model), the view (GUI) and the controller that allows the connection between the previous elements. In this case, the model is the files stored locally, which are constantly consulted with programmed functions and displayed in view so that the user can access the information in a clear way. Figure 3 depicts the distribution of the wizard through this design pattern.

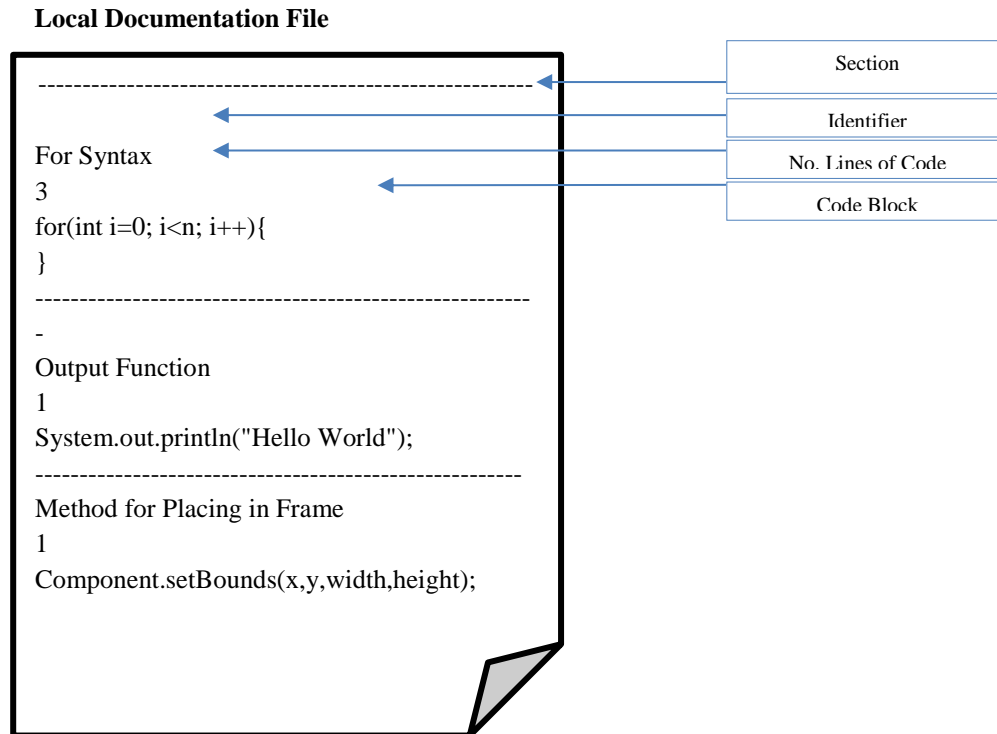
Figure 3 Model Controller View



Source: Own Work

The files in the project play the role of a database, since they will store useful information to maintain a good functioning in each of the software modules. In most of the files to be used Send-text are a list format, accessing through the name and getting the value in front of it, however, the local code file is different and follows the format that can be seen in figure 4, in which it is graphically explained how it works.

The codes are stored in 4 parts, the identifier which is the top, the number of lines of the code block (counting the sections) which will serve to extract only the code block or jump to another identifier in case it is not what you are looking for.

Figure 4 Local Code Format

Source: Own Work

3. Implementation

By default, Python has a large number of libraries and functions that allow you to develop correct and functional projects. Table 1 lists all the dependencies required for the coding of the project plus a previous description. These libraries are intended to develop advanced interfaces, implement speech synthesis and recognition, as well as Web Scraping search.

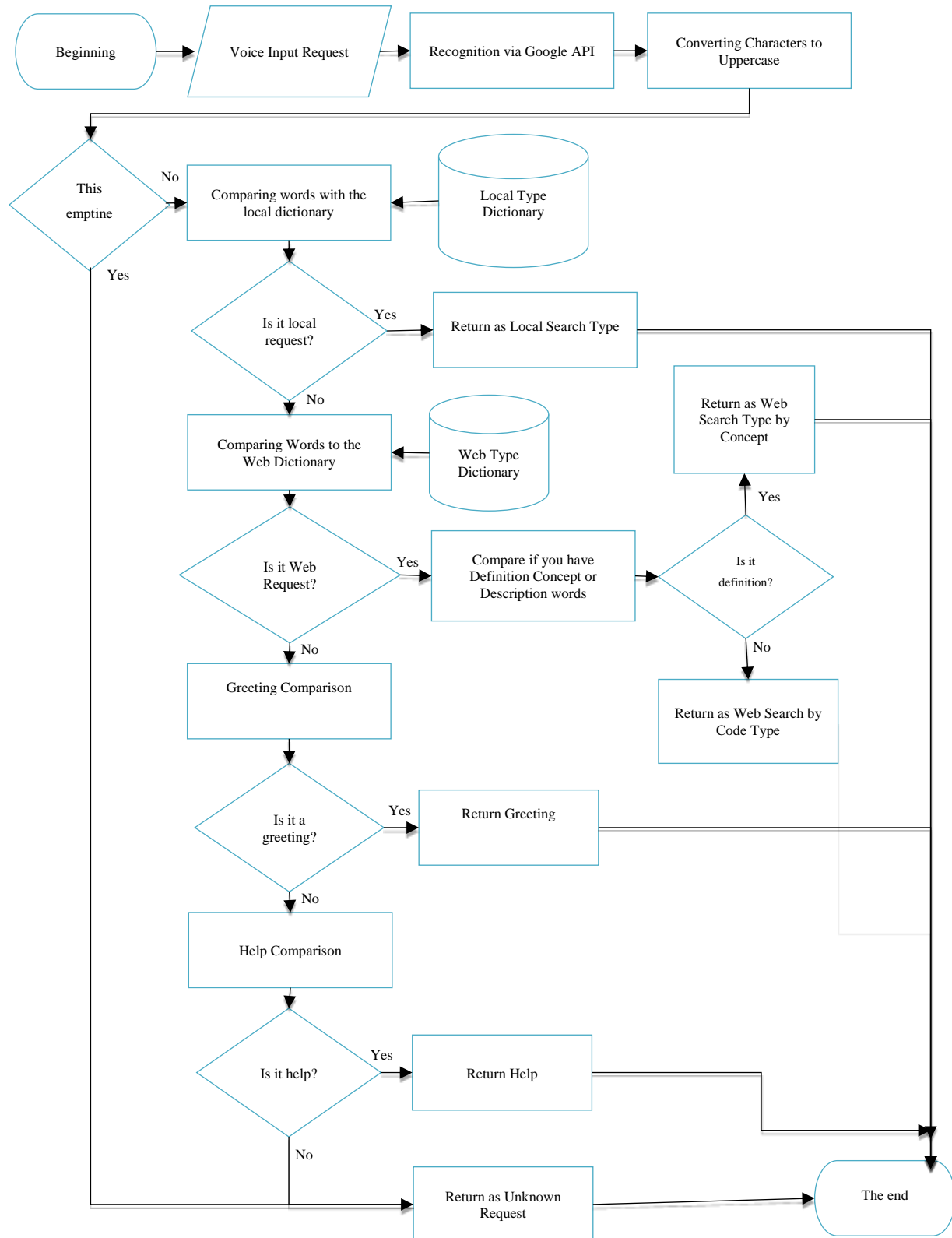
Table 1 Technologies implemented with Python

Package	Description
PyQt5	It is a set of modules of the Qt5 framework for the creation of advanced graphical interfaces
PipWin	A package that allows for the installation of packages with greater compatibility for Windows operating systems.
Speech Recognition	Library that allows speech recognition through various API's such as Google, Google Cloud, IBM, Sphinx, etc.
Pytsx3	Python package for offline text-to-speech.
Pyaudio	It provides links to make use of the audio input and output devices.
PlaySound	It allows the playback of audio files through the various audio devices.
Google	Allows connection to Google services.
Google Search	It allows access to and use of the Google search engine to make queries on the web.
Beautiful Soap	It extracts information based on specific tags in the HTML of a queried site.

One of the characteristics of a virtual assistant is the use of natural language processing, accepting inputs from the keyboard or through voice, in this case it was emphasized that requests could be made from both means. Figure 5 shows the flow that follows the identification of the requests made by a user, where first the voice input is made through the microphone, and then based on Google Speech the conversion of speech to text is carried out. Once obtained, the text is converted into capital letters to avoid problems when comparing the different strings that allow identification.

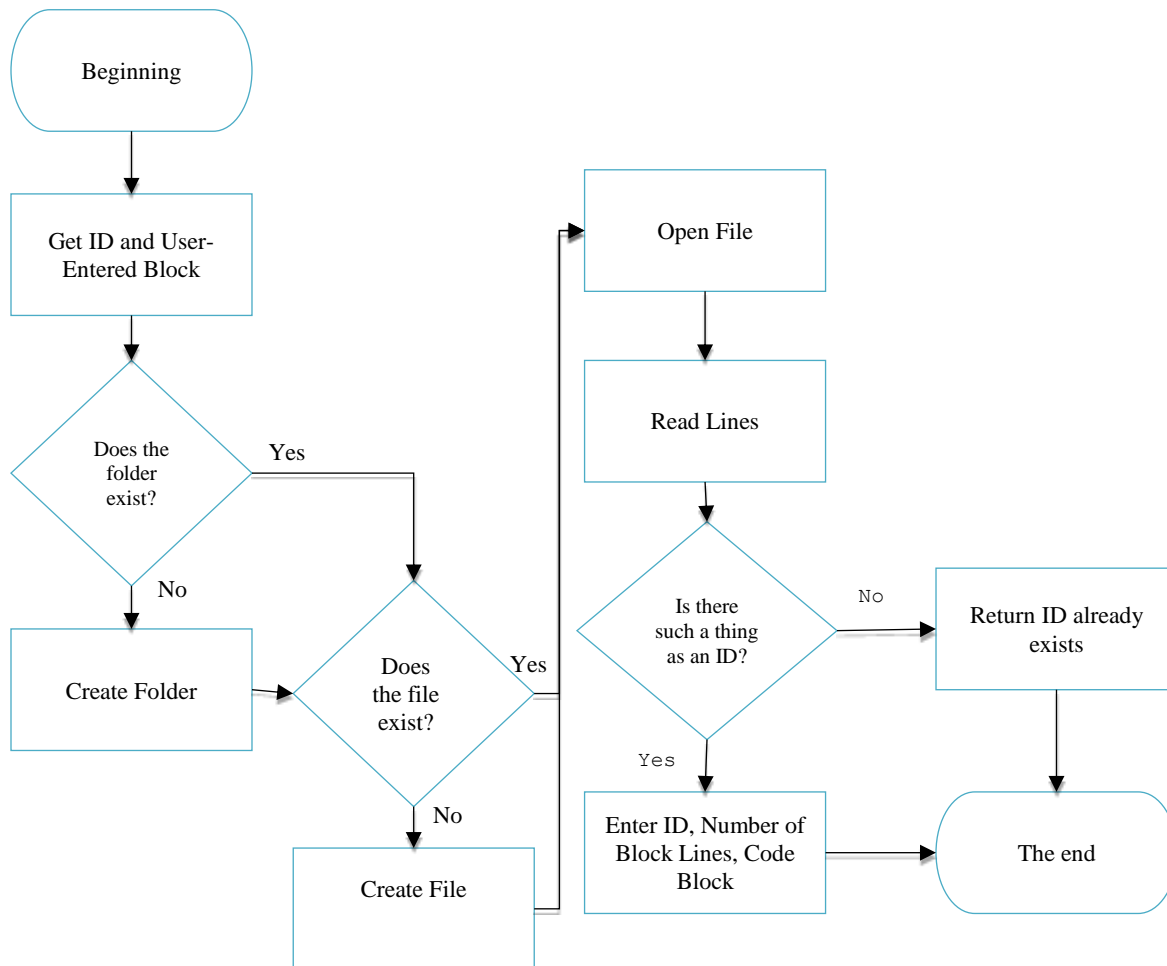
Once the text is obtained, it is verified that it has really been recognized, then the comparison is made with a dictionary of words to perform a local search, if there is no match of 80% it goes to the comparison of words for web search, otherwise it is verified if it is a greeting or a request for help. In case any of these do match, it returns a string to be able to process the request, cleaning up the words that are not useful and performing the next process with only the most important data.

Figure 5 Speech Recognition Operation Flowchart



Source: Own Work

To enter a code locally, to be able to have it available at all times, an identifier and a block of code are required, which will be stored in a folder and a file stored in the same directory of the executable. In figure 6 you can see the process that is carried out to add codes to these files, in which it is mainly verified if the folder and file exist, if not it creates them to later analyze the file, it is worth mentioning that a line-by-line analysis is not performed, since the local file makes use of a format that allows skipping lines to save time based on the number of lines of Codes already stored and only read the code identifier to determine if the identifier already exists and avoid duplicate values, if not, it is added to the end of the file.

Figure 6 Handling Local Code Injection

Source: Own Work

When consulting the local file to obtain some code that the user needs, it will be displayed in a window where the code block can be edited, so that based on two buttons the user can update the stored code or delete it.

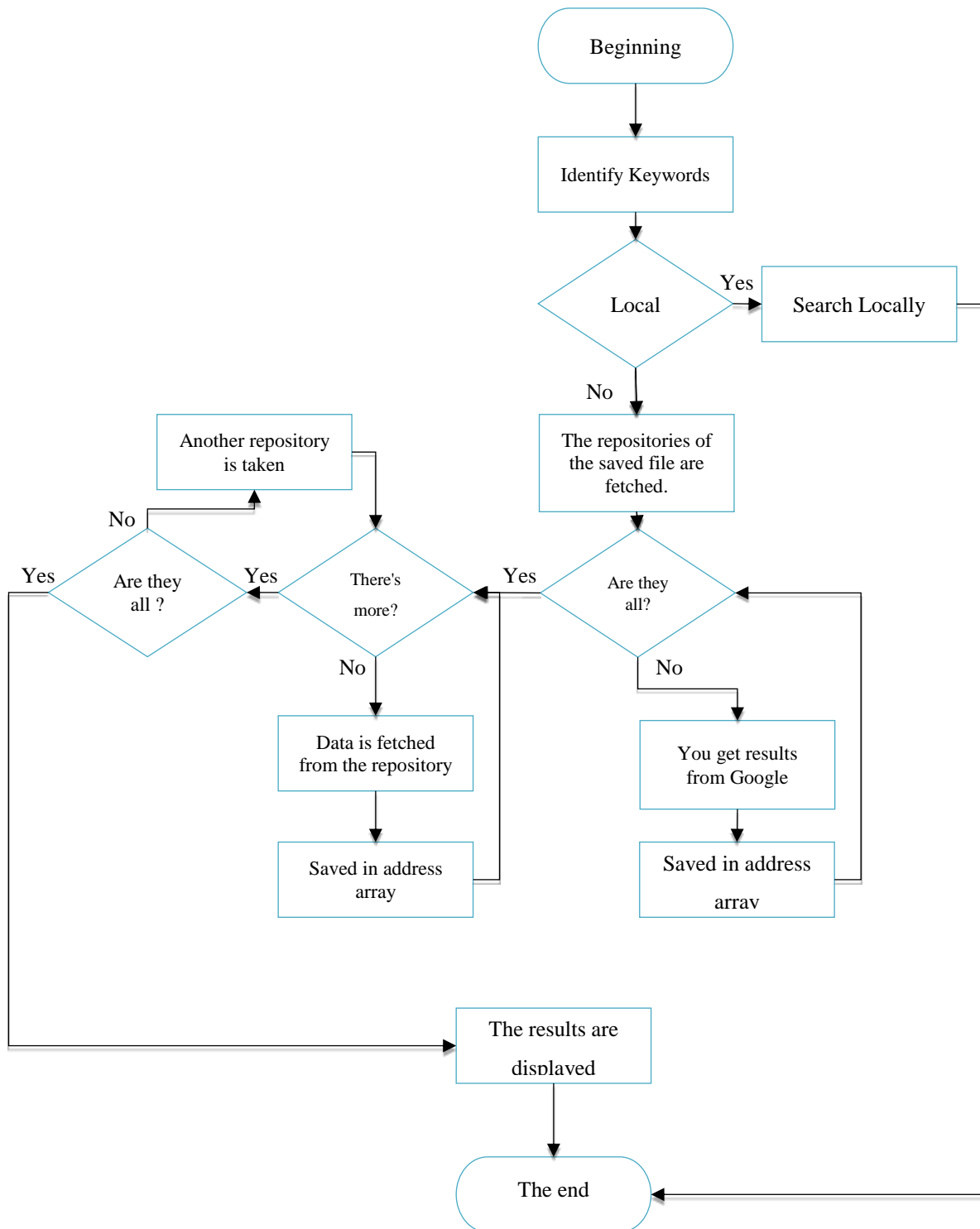
To delete locally stored codes, follow the following process:

- a) Beginning
- b) Open the local code file.
- c) Read the number of lines in the file and store them in a variable
- d) Open the file again and read by performing line breaks based on the number of lines per code block.
- e) If the ID to be deleted matches the parsed file, store the ID, the number of lines, and the code block in a string variable.
- f) Replace the text that matches the previous string in the variable where the lines of the entire file were stored.
- g) Overwrite the local code file with the result of the text replacement.
- h) The end

To update a local code, you run the algorithm to delete it and then add it back with the updated structure.

Likewise, in the search for data on the Internet, web scraping techniques were used from the Google search engine and beautiful Soup to obtain the information of the HTML tags. This algorithm can be seen in figure 7, in which the message is first identified, and then identified that it is a web search and based on the repositories that are listed in files, the search is carried out in those sites, taking the first two matches and performing the inspection to return it to the user.

Figure 7 Web Scraping Algorithm



Source: Own Work

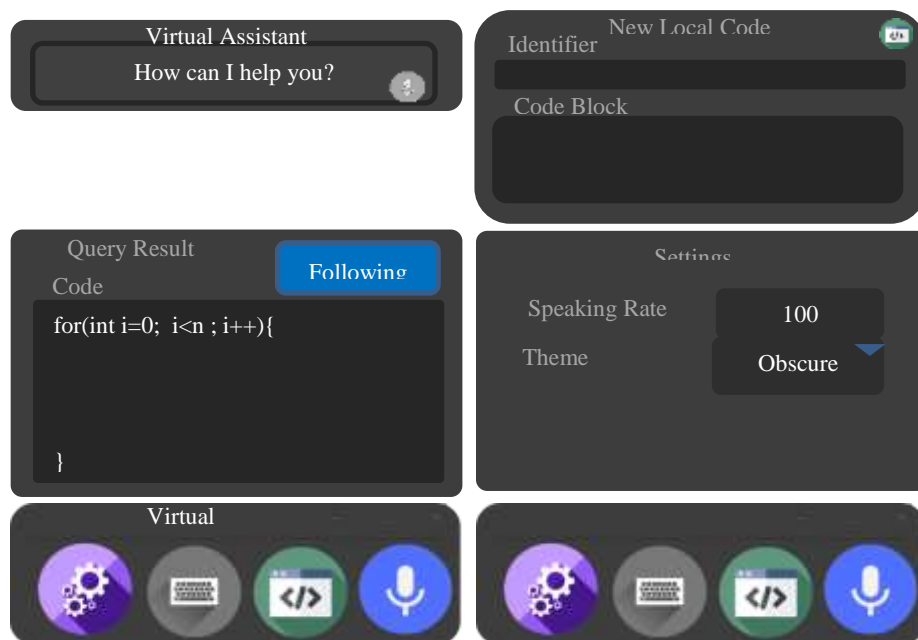
The coding of the graphical interface was developed based on the Qt5 framework through the modules of the PyQt5 library, which allows the development of advanced graphic components, making use of technologies such as CSS and HTML. Each of the windows has its own integrated action bar to perform the closing and minimization, this with the purpose that the interface is not visually affected when you have another theme or a different version of the operating system, avoiding display errors. And they are seen in Figure 8.

Figure 8 Software Interfaces

Source: Own Work

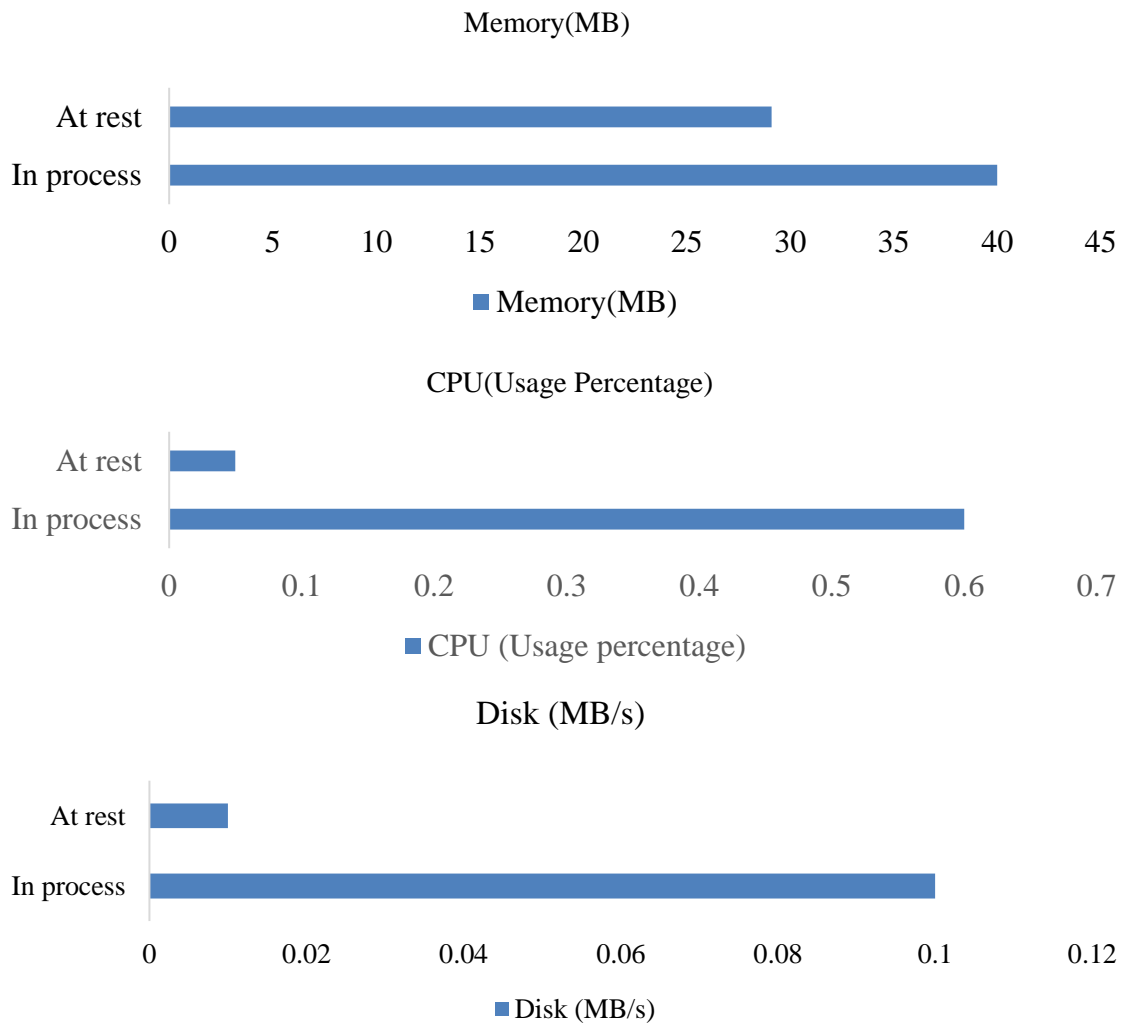
4. Discussion

Once each of the necessary modules has been developed and each of the interfaces has been coded, the final result is shown in Figure 9, being able to perform natural language processing via keyboard or voice and then make a query through the web based on techniques such as web scraping or a query in the local file.

Figure 9 How the Software Works

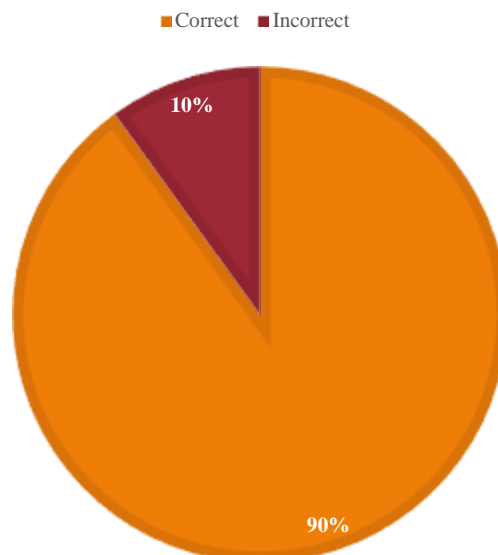
Source: Own Work

The consumption of the software is quite low for a hardware with 4 cores and 8 GB of RAM, consuming as little as 40 MB of RAM and 0.60% when processing a request from the user, In Figure 10, you can see graphs of the overall consumption of the software.

Figure 10 Consumption of developed software

Source: Own Work

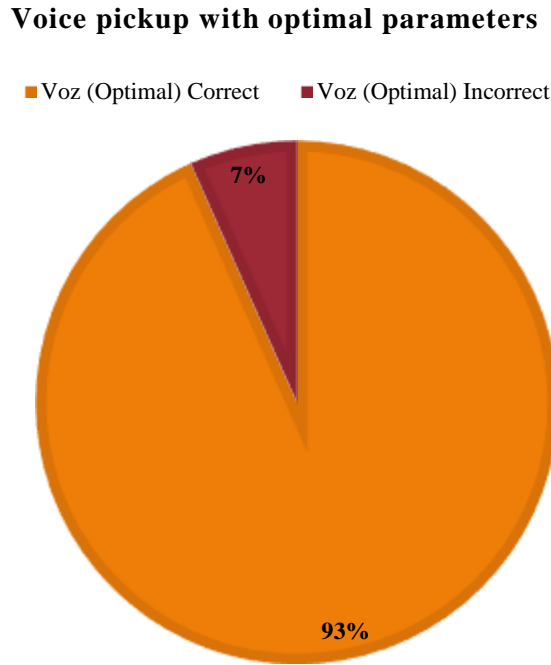
The developed software correctly fulfilled the proposed query functions, however, to corroborate the hypothesis, several tests were carried out with different types of environments. The first test was carried out with a sample of 30 situations in which internet queries were made about concepts and syntax related to java programming, and as can be seen in figure 11, 90% of these had satisfactory results, with an error rate of 10%.

Figure 11 Consultations made on the internet

Source: Own Work

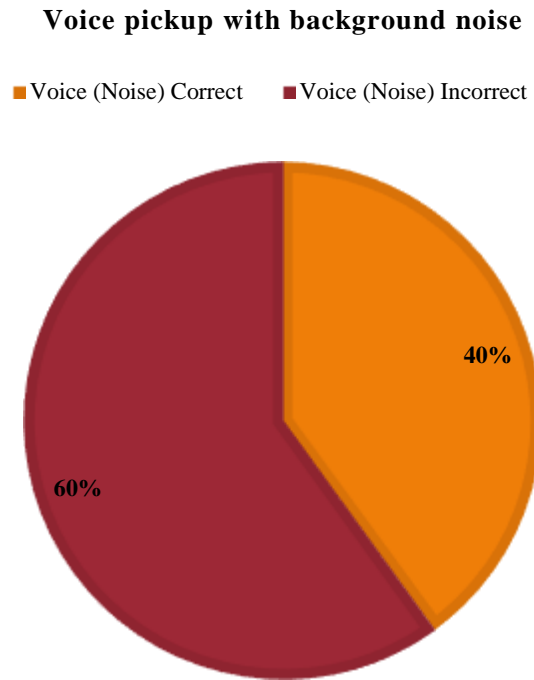
The internet query has the desired precision, another factor that can hinder the search for concepts is the voice due to the different noises that may arise, in this sense, the voice detection test was carried out with various parameters, in figure 12, it can be seen that in optimal conditions with little noise in the environment the accuracy is 93% in recognition, However, it is important to mention that if the environment is not controlled or there is a lot of noise, the accuracy percentage can drop to 40%, as seen in Figure 13.

Figure 12 Consumption of developed software



Source: Own Work

Figure 13 Speech recognition tests under different noise conditions.



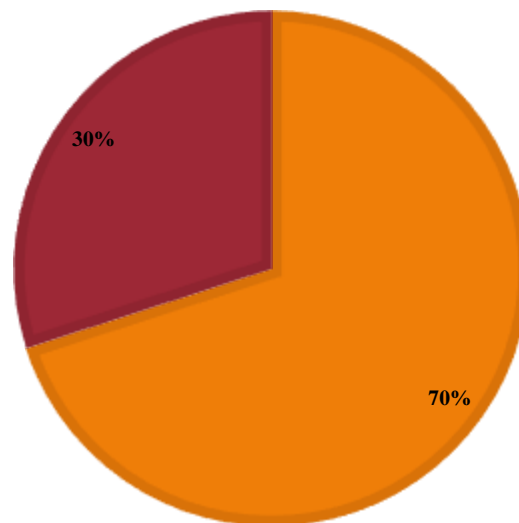
Source: Own Work

An equally relevant factor when implementing natural language processing is to recognize speech regardless of what type of emotion is dictated or at what speed the pronunciation of the words is performed, therefore, tests were also carried out with changes in the speed when the user dictates a request and as seen in figure 14, The accuracy achieved when it comes to recognizing a request at high speed is 70%, reducing up to 20% compared to when the query is presented under normal condition.

Figure 14 Recognition tests when changing the speed of the speaker's request

Voice pickup with speed change in dictated petition

■ Voice (spacing) Correct ■ Voice (spacing) Incorrect



Source: Own Work

5. Conclusions

Technology must be used for the growth of humanity's knowledge, in this case the intelligent assistant for learning the java language met the objectives, such as the search on the web through natural language recognition and its processing, locally with the implemented files and their respective identifiers, as well as more than acceptable results in the general set of implemented functionalities.

The main objective of this project was to develop an assistant that supports students who want to learn the java programming language, therefore, this software was more focused on searching for information on the internet or locally with an accuracy of 90% in the results so that the user obtains the information that is of help. And based on the tests carried out, this percentage of accuracy was adequately achieved, with a speech recognition of 93% under appropriate conditions.

The points to be improved in this project are especially in the validation of recognition in conditions where they are not favorable for the assistant, since, for example, in the capture of voice with background noise, the error rate increased by up to 60%, which is not optimal at all. Another aspect to consider is the implementation of an offline speech recognition engine in the future so that the assistant can understand the voice regardless of whether the device has an internet connection, this through technologies such as Sphinx, being that it would be very useful since the assistant can make queries locally based on the file that stores that data.

It can be concluded that the virtual assistant based on natural language processing for learning the Java language was done correctly, however, there are details that can be improved and that in future versions can be corrected or scaled the project to more programming languages since the java bases could be implemented in a correct and functional way.

Acknowledgement

We are grateful to the Tecnológico Nacional de México and Tecnológico de Estudios Superiores de Jocotitlán for the support given in carrying out this research project.

References

- Camacho-Alvarez, M. &.-A. (marzo de 2020). Procesamiento del lenguaje natural con Python. *Revista de computo aplicado*, 4(13), 24-28. doi:10.35429/JCA.2020.13.4.24.28.
- Cuentas Chavez, E. F. (2020). Desarrollo de una aplicación de seguridad vial usando el Sistema de Posicionamiento Global (GPS); para prevenir accidentes vehiculares”, Tesis profesional. Juliaca, Perú: Universidad peruana Unión. Obtenido de <http://hdl.handle.net/20.500.12840/4620>
- Díaz Fernández, C. (2018). Diseño e implementación de un asistente personal inteligente en español basado en reconocimiento de voz empleando Raspberry Pi. Madrid: Universidad Carlos III de Madrid. Obtenido de <http://hdl.handle.net/10016/29445>.
- GARCÍA-AMARO, J. F.-R.-A.-Á. (junio de 2017). Revisión de técnicas de pre-procesamiento de textos para la clasificación. *Revista de Cómputo Aplicado*, 1(2), 1-11. Obtenido de https://www.ecorfan.org/spain/researchjournals/Computo_Aplicado/vol1num2/Revista_de_Computo_Aplicado_V1_N2_1.pdf.
- García-Reina, L. F. (2018). Asistente virtual tipo ChatBot. Colombia: Universidad Católica de Colombia, Facultad de Ingeniería. Obtenido de <https://hdl.handle.net/10983/17726>.
- Gonzalez, D, L. A. (2013). Web scraping technologies in an API world”, *Articulo científico. Briefing in bioinformatics* Vol. 15., 788-790. doi:10.1093/bib/bbt026.
- Jorge Matich, D. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. Facultad regional rosario departamento de ingeniería química, 8-11. Obtenido de https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf.
- Maza A., D. S. (2017). Análisis, diseño e Implementación de un asistente tutor basado en computación cognitiva para su uso en entornos visuales de aprendizaje. Universidad técnica particular de la roja, Área técnica,. Obtenido de <https://dspace.utpl.edu.ec/bitstream/20.500.11962/20589/1/Arteaga%20Maza%2c%20Sleyder%20Daniel.pdf>.
- Moreira, D. C. (2020). Análisis del Estado Actual de Procesamiento de Lenguaje Natural. *Revista Ibérica de Sistemas e Tecnologías de Información, RSTI, N° E42*, 126-136. Obtenido de <http://www.risti.xyz/issues/ristie42.pdf>.
- Nowrin, S. &. (2023). Programming by Voice: Exploring User Preferences and Speaking Styles. *Proceedings of the 5th International Conference on Conversational User Interfaces, CUI*. doi:<http://doi.org/10.1145/3571884.3597130>.
- ORTEGA-GINES, H. B.-H.-O.-O. (2019). Aplicación de voz, como asistente virtual para el manejo de las emociones. *Revista de Tecnologías Computacionales*, 2-7. doi:10.35429/JOCT.2019.12.3.1.7.
- Pérez, E. M. (2013). tesis “Sistema de Seguridad Por Reconocimiento de Voz”. México. Obtenido de [https://tesis.ipn.mx/bitstream/handle/123456789/12309/Sistema%20de%20Seguridad%20por%20Reconocimiento%20de%20Voz%20\(Tesis%20de%20Ingenieria%20ESIME\).pdf](https://tesis.ipn.mx/bitstream/handle/123456789/12309/Sistema%20de%20Seguridad%20por%20Reconocimiento%20de%20Voz%20(Tesis%20de%20Ingenieria%20ESIME).pdf).
- Sánchez. G.A. (2021). Puesta a punto de un robot real autónomo inteligente usando ROS, Trabajo de fin de grado. Cartagena, Colombia: Universidad politécnica de Cartagena. Obtenido de <https://repositorio.upct.es/bitstream/handle/10317/9454/tfg-san-pue.pdf>
- Ssanyu, J. B. (2023). Enseñanza del procesamiento del lenguaje con el marco PAMOJA. *Ciencia de la programación informática*(102959), 229. doi:<https://doi.org/10.1016/j.scico.2023.102959>