

De acuerdo con nuestra arquitectura MVC (Modelo – Vista – Controlador) la vista tendrá una interfaz minimalista en la que solamente contemos con cuatro funciones en una ventana reducida, ubicada (de manera inicial) en la esquina inferior derecha, las funciones de cada uno de los botones es la siguiente:

1. **Configuración:** La configuración hace referencia a los ajustes visuales que se le pueden aplicar a la interfaz, como lo es el tema y los colores que tiene, aspectos de la voz y el micrófono, como lo es, la reducción de ruido y la velocidad del narrador.
2. **Entrada por teclado:** El teclado tiene una función fundamental para dispositivos (sobremesa) que no tengan micrófono de manera nativa, ya que nos permite utilizar el asistente sin la necesidad de ingresar voz.
3. **Introducir código a mano:** El poder almacenar código de manera local es una función que se desea almacenar para poder ampliar los horizontes del usuario ya que, al introducir el código a mano, se abre la posibilidad de que el usuario comience a crear sus propias funciones que se adaptan a las necesidades de cada usuario.
4. **Entrada por voz:** Es la función principal a la que está orientada el asistente, a que permitirá al usuario usarlo mientras se ejecuta otro programa, sin necesidad de escribir con el asistente, agilizando el desarrollo por parte del usuario.

El controlador tendrá diferentes funciones que nos permitirán conectar con el modelo, procesar la información y posteriormente desplegarlos al usuario, al tiempo que el usuario también podrá almacenar en el modelo con ayuda de las funciones especificas en el controlador para guardar de manera local el código consultado o el código escrito por el usuario. Dentro de las funciones que destacan tenemos:

1. **Consulta:** Que tiene como parámetros el *Filter* que hace referencia al conjunto de páginas que fueron seleccionadas para poder hacer la búsqueda de la información para ser mostrada en la aplicación, posteriormente tenemos el *Tarjet* que es el objetivo de la búsqueda, en este caso el conjunto de palabras clave previamente analizadas y validadas, después encontramos el *TTK* que es el tiempo para determinar si el usuario está conectado a internet y una vez validado podrá tomar dos caminos dependiendo el caso, webScrapping (online) o localSearch (offline), para finalizar tendremos *connection* que se refiere a la conexión con nuestro modelo local (fichero) y en línea (web).

También contará con funciones que nos permitirán hacer la búsqueda mediante parámetros en internet *webScrapping* así como funciones para retornar el resultado (*returnWeb*) o guardar el código de manera local (*saveConsult*).

1. **Send-Audio, Send-Text:** En principio comparten la misma lógica, con la diferencia que el audio utilizará un par de módulos extra, los cuales son *speachToText* y *performTarjet*, que se encargarán de convertir la voz a texto y a encontrar las palabras clave de la consulta.
2. **Create-Code:** El principal uso de esta función es permitir que el usuario guarde su código de manera local sin importar si está conectado a internet o no, ya que se almacenará en el fichero local establecido. Recibirá parámetros como ruta del fichero (path), identificador de código (id) y el código como tal (code), a la vez que tendremos una función para obtener el largo del código para poder optimizar la búsqueda local (getLenght).

Para finalizar tendremos el modelo, que se compone de dos conexiones principalmente, la local (fichero con los códigos almacenados) y la remota (sitios que se encuentran en internet que cumplan con las características establecidas).