Uso de Analizador.exe

# Composición de los archivos dentro de la carpeta *src*

## Archivo index.py

Cuenta con los procedimientos necesarios para unir todas las funciones, el flujo es el siguiente:

1. Espera la opción de bandera
2. Descarga el archivo de la plataforma y verifica que se haya realizado la descarga
3. Procede a hacer los análisis
4. Subida a Base de Datos
5. Obtención de gráficos

## Carpeta download

Es una carpeta asociada a los módulos utilizados para automatizar las descargas de las páginas web, de tal manera que cumple una función similar a las consultas por archivos en base a una API.

### Archivo call.py

Utiliza Selenium para acceder a la plataforma Genesys Cloud y realizar la descarga de llamadas. Cuenta con la clase **Descargar\_Llamada** con los siguientes métodos.

* descargar(self, llamada\_id): Accede a la página y procede a descargar una llamada, es necesario proveer el id.
* validar(self, llamada\_id): Verifica que un archivo haya sido descargado, para cerrar la ventana del explorador dado el id de la llamada.

### Archivo chat.py

Utiliza Selenium para acceder a la plataforma S1 Gateway y descargar los chats. Cuenta con la clase **Descargar\_Reporte**.

* descargar(self, patron): Accede a la página y procede a descargar una llamada, puede descargar mensajes o reportes de conversaciones a partir de un patron especificado.
* validar(self: Verifica que el archivo excel haya sido descargado, para cerrar la ventana del explorador dado el id de la llamada.

## Carpeta load

Carpeta utilizado para la creación y carga de modelos de IA.

### Archivo maquina.py

Utiliza un modelo de IA de clasificación basado en el procedimiento Maquina de Soporte Vectorial (SVM). Esta clase cuenta con los métodos:

* clasificar(self, df): Recibe un dataframe de excel de conversaciones de chats para obtener un vector donde cada fila se le asocia un producto que fue entrenado anteriormente.
* clasificar\_texto(self, texto): Recibe una transcripción y puede obtener una predicción de producto.

### Archivo sentimientos.py

Utiliza un modelo de IA para analizar textos y retornar los valores entre 0 a 1 para el análisis de sentimientos. Se retornan diccionarios de valores para su resultado positivo, negativo y neutral.

### Archivo transcripcion.py

Realiza una carga de un modelo de IA de transcripción que hace una carga del modelo **Whisper V3 Large.** Se le provee una ruta de un mp3 en el método *transcribir(self, ruta\_audio)* para obtener una lista de transcripciones con sus respectivas marcas de tiempo.

### Archivo vectorizadores.py

Utiliza modelos de procesamiento de texto para crear modelos basados en palabras clave en cada documento que representa una conversación ya sea llamada o chat.

* Counter: Esta clase corresponde a la carga de un procedimiento de vectorización. Esta clase puede obtener una lista de conversaciones y retorna la misma cantidad de vectores para cada palabra signficativa. Cada valor del vector corresponde a su frecuencia.
* TfIdf: Esta clase corresponde a la carga de un procedimiento de vectorización. Esta clase puede obtener una lista de conversaciones y retorna la misma cantidad de vectores para cada palabra signficativa. Cada valor del vector corresponde a su frecuencia invertida por la cantidad de documentos que contienen aquella palabra.

## Carpeta utils

Carpeta con funciones utiles para utilizar en el archivo ```index.py```.

### Archivo database.py

Funciones para obtener datos esenciales sobre alguna conversación postventa y subir resultados de los análisis.

* Análisis de conversación de Llamada y Chat: *obtener\_producto\_asesor\_grupo\_IVR* y *obtener\_producto\_asesor\_grupo\_S1*.
* Subida de resultados de Llamada y Chat: *subir\_datos\_IVR* y *subir\_datos\_S1*.

### Archivo fragmentar.py

Función *Fragmentar(ruta\_audio, carpeta\_salida, tiempo\_fragmento)* que recibe una ruta de archivo y una ruta de carpeta que recibirá los audios particionados.

### Achivo graficos.py

Métodos para crear los siguientes gráficos:

* Nubes de palabras: *graficos\_palabras(diccionario, carpeta, ruta\_salida)*
* Gráfico de barras de tiempos de espera/ muertos: *grafico\_barras\_espera(espera\_min, espera\_max, espera\_prom, caso\_id, carpeta, ruta\_salida)* | *grafico\_barras\_activo(tiempo\_activo, tiempo\_total, carpeta, ruta\_salida)*
* Gráfico de barras de análisis de sentimientos:

### Archivo texto\_clave.py

Utilizar funciones para obtener tokens de texto ya sea utilizando los vectorizadores **Counter** y **TfIdf**.

### Archivo wait.py

* obtener\_tiempos(df): Obtener tiempos de diferencia entre mensajes
* tiempo\_muerto(dfs): Obtener tiempos dado las marcas de tiempo de transcripción donde el cliente espera a la respuesta del asesor

# Requisitos (NO MOVER LA CARPETA *res* POR NINGÚN MOTIVO, MOVER TODO EL DIRECTORIO)

Es necesario definir bien los archivos res/descarga\_chat.json, res/descarga\_llamada.json y res/rutas.json.

**Nota**: Para algunos valores es necesario guiarse de <edge://version/> dentro de edge.

## Archivo res/descarga\_chat.json

### Decripción de parámetros

* user-data-dir: Ruta encontrada en  <edge://version/>
* profile-directory: Ruta encontrada en <edge://version/>
* msedgedriver: Ruta del webdriver de Edge, este debe ser compatible, puede descargarse de: [https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/?form=MA13LH#downloads](https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/?form=MA13LH%23downloads)
* downloads: Carpeta de decargas de Edge
* range: Cantidad de dias a tomar en cuenta (ej: "yesterday" y "last2weeks")

### Ejemplo de parámetros

{

    "user-data-dir": "C:/Users/AppData/Local/Microsoft/Edge/User Data",

    "profile-directory": "Profile 3",

    "msedgedriver": "C:\\Users\\Documents\\Speech-Analytics\\Descargar\\msedgedriver.exe",

    "downloads": "C:/Users/Downloads/Speech-Analytics-Downloads",

    "range": "yesterday"

}

## Archivo res/descarga\_llamada.json

### Descripción de parámetros

* user\_data\_dir: Ruta encontrada en <edge://version/>
* profile\_directory: Ruta encontrada en <edge://version/>
* edge\_driver: Ruta del webdriver de Edge, este debe ser compatible, puede descargarse de: [https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/?form=MA13LH#downloads](https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/?form=MA13LH%23downloads)
* admin\_tab\_id: Parámetro tabId dentro del enlace de IVR (https://apps.mypurecloud.com/directory/#/analytics/interactions/29af59bb-657c-4bea-b2dc-51f884f14c7a/admin?tabId=346200bf-61d3-4c17-a54f-23370c6cf11d)
* download\_directory: Directorio de descargas

### Ejemplo de parámetros

{

    "user\_data\_dir": "C:/Users/bp3285/AppData/Local/Microsoft/Edge/User Data",

    "profile\_directory": "Profile 3",

    "edge\_driver": "C:\\Users\\bp3285\\Documents\\Speech-Analytics\\Descargar-Llamadas\\msedgedriver.exe",

    "admin\_tab\_id": "346200bf-61d3-4c17-a54f-23370c6cf11d",

    "download\_directory": "C:/Users/bp3285/Downloads/Speech-Analytics-Downloads"

}

## Archivo res/rutas.json

### Descripción de parámetros

* export: Directorio donde se encuentran los modelos y ffmpeg
* fragmentos: Directorio que fragmenta los audios para su procesamiento
* salida: Direcotrio en el que se obtendrán los gráficos, transcripciones, mensajes, etc.

### Ejemplo de parámetros

{

    "export": "C:/Users/bp3285/Documents/ejecutable/export",

    "fragmentos": "./fragmentos",

    "salida": "./salida"

}

# Funciones

## Descarga y análisis

### Conversaciones de WhatsApp

.\analizador.exe --excel

### Conversaciones de Teléfono

.\analizador.exe --mp3={uuid-de-la-llamada}

## Análisis de archivos

### Conversaciones de WhatsApp

.\analizador.exe --ruta\_excel={ruta-a-archivo-xlsx}

### Conversaciones de Teléfono

.\analizador.exe --ruta\_llamada={ruta-a-llamada-mp3}