

SOFIA BOT DOCUMENTACIÓN

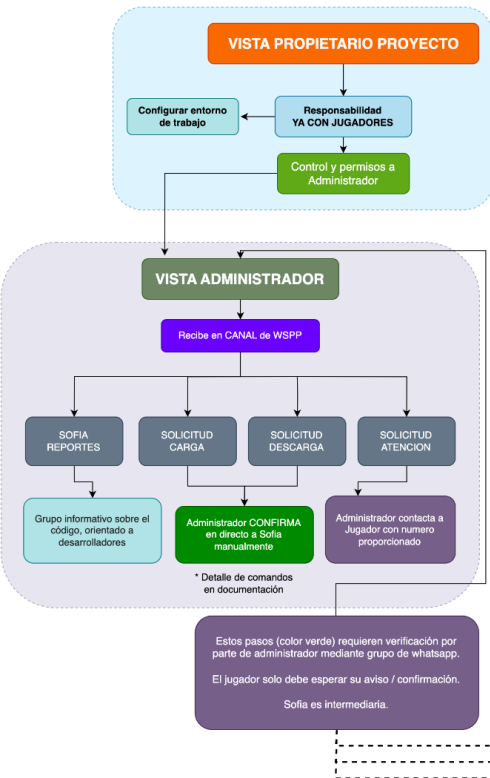
SOLUCION

Automatizar el proceso que se realiza entre el jugador, cajero y plataforma acelerando los tiempos de respuesta y eliminando la interacción por parte del cajero al máximo nivel.

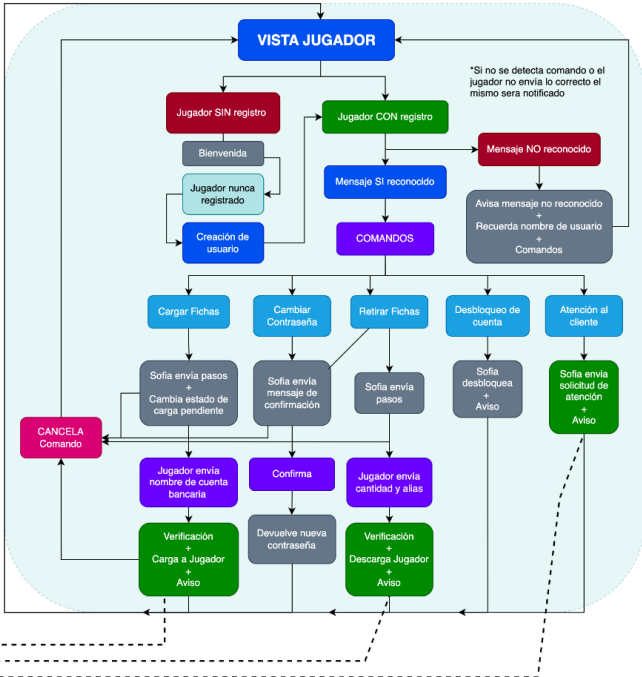
INDICE

- FLUJO COMPLETO
- STACK TECNOLOGICO
- OBSERVACIONES Y RECOMENDACIONES
- ROL SUPERADMINISTRADOR / PROPIETARIO
- COMANDOS SUPERADMINISTRADOR
- ROL ADMINISTRADOR
- COMANDOS ADMINISTRADOR
- **IDENTIFICAR ERRORES Y REPORTE**
- **PROCESO LEVANTAR Y CORRER PROYECTO**
- PERSONALIZACION DE MENSAJES
- SOLUCION ELEMENTO NO ENCONTRADO
- ACTUALIZAR CLASES WHATSAPP
- **LINK VIDEOS EXPLICATIVOS**

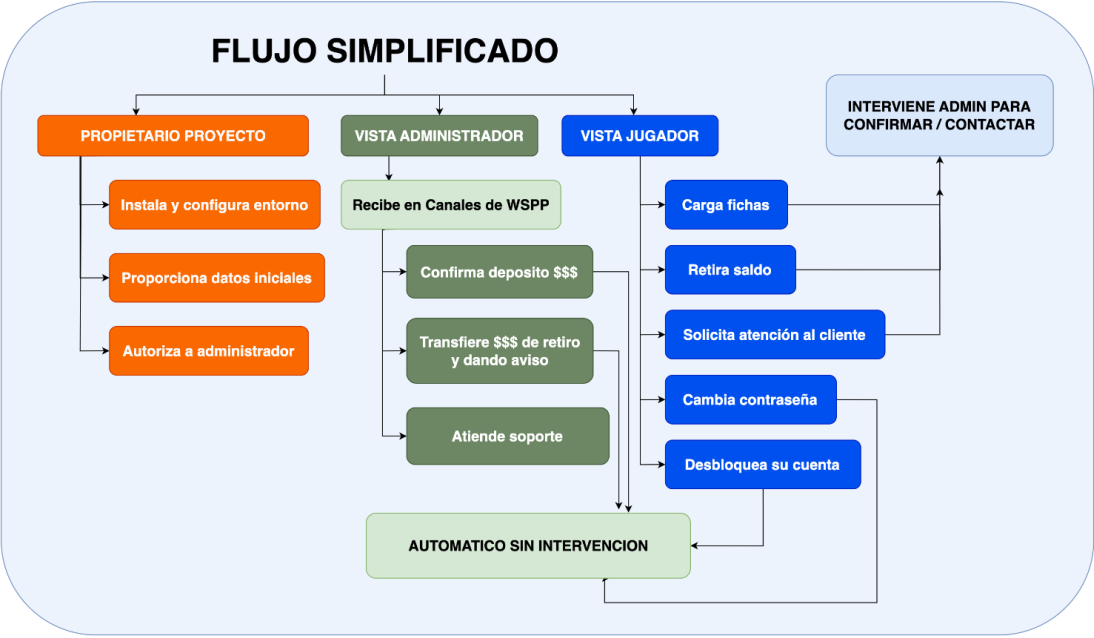
FLUJO COMPLETO



FLUJO COMPLETO SOFIA BOT



FLUJO SIMPLIFICADO



STACK TECNOLÓGICO

El proyecto se construye utilizando las siguientes tecnologías:

- Python
- Selenium
- PyAutoGUI
- XAMPP
- MySQL
- Pandas
- Numpy

Conociendo las bases de cada una de estas podremos desarrollar, corregir o modificar el mismo con mayor entendimiento. Para este entendimiento se debe tener en cuenta el flujo general.

OBSERVACIONES Y RECOMENDACIONES

El bot fue desarrollado desde un principio para que funcione en un UN SOLO PANEL con UNA SOLA BASE DE DATOS y UN SOLO NÚMERO DE WHATSAPP, todo corriendo y funcionando en UN SOLO SERVIDOR. Esto mismo debido a las siguientes razones:

- Seguridad, al tener diversificado cada cliente que use el bot en un solo servidor, con su propia base de datos y su propio número de teléfono, se minimiza al máximo el riesgo extendido. Es decir, si existe algún tipo de problema externo, se da de baja el servidor, número, y base de datos afectado, y listo. Este no afecta a la red de jugadores que se posea ya que funciona sólo como un asistente, y se puede volver a levantar sin problemas en otro servidor, con otra base de datos, otro número de teléfono y el mismo panel de plataforma. Sin mayores inconvenientes, aunque cabe aclarar de que si no se hizo una copia de la base de datos antes de llevarlo o moverlo a otro servidor estos datos se perderán ya que estos solo se encuentran en el mismo server, la solución a esto es de que antes de dar de baja o borrar, se debe hacer una copia de la base de datos de forma manual.
- Su diseño está hecho para que corra UN solo bot (no más de uno) en un mismo servidor, con un número de teléfono y una base de datos. Si se corren más de un bot, puede ser que funcione sin problemas, pero es muy probable que aparezcan errores inesperados y el funcionamiento del mismo se rompa.

A cada cliente que se le venda el servicio del bot, se le debe contratar un servidor propio, que contenga su propia base de datos, y se vincule al número de whatsapp proporcionado por el mismo cliente. De esta forma, se aseguran de que el bot va a correr de forma óptima, y que exista una independencia total entre clientes, lo que hará mucho más fácil el control y administración del servicio. El número de Whatsapp será proporcionado por el cliente, y el servidor, base de datos y panel por el dueño del bot. De esta forma se tendrá un control total

sobre los movimientos realizados en el mismo, y sobre todo el código fuente que deberá estar corriendo en la consola del servidor.

ROL SUPERADMINISTRADOR / PROPIETARIO

El rol del propietario se define por sus tareas y responsabilidades las cuales son:

- Configurar el entorno de trabajo.
- Delegar y controlar los permisos de inicio para Administrador.

El propietario, a su vez, posee comandos de SUPERADMINISTRADOR y ADMINISTRADOR para poder realizar y cumplir con sus responsabilidades sin restricciones por parte del software.

Detalles de como hacer esto en PROCESO LEVANTAR Y CORRER PROYECTO.

COMANDOS SUPERADMINISTRADOR

Recordemos que el SUPERADMINISTRADOR es el propietario del proyecto.

Comandos:

Se incluyen todos los comandos de administrador agregando:

- Cambiar número administrador: **Admin + NUMERO DE WHATSAPP** un ejemplo de uso sería el siguiente **Admin +54 9 3523 41-1111**. El número que coloquemos deberá escribirse TAL CUAL figura el mismo en WhatsApp WEB, incluyendo el signo más (+), los espacios y el guión.

ROL ADMINISTRADOR

El rol del administrador se define por sus tareas y responsabilidades las cuales son:

- Confirmar o cancelar cargas.
- Confirmar o cancelar descargas.
- Atención al cliente.
- Acceso a reportes.

Cada una de estas responsabilidades corresponde a un grupo en donde Sofía enviará mensajes dando aviso, estos grupos son solo de lectura, para confirmar o realizar alguna acción con respecto a estos grupos de avisos, se deberá comunicar POR PRIVADO al número de Sofia utilizando los COMANDOS ADMINISTRADOR.

Las vistas de los grupos serán del tipo:

- Confirmar o cancelar cargas:



SOLICITUD DE CARGA

~BCNbot: OPERACION N° 170. **CARGA**

- Confirmar o cancelar descargas:



SOLICITUD DE DESCARGA

~BCNbot: -----

- Atención al cliente:



SOLICITUD ATENCION AL CLIENTE

~BCNbot: El usuario **zorro35** con el numerc

- Acceso a reportes:



SOFIA REPORTES

~BCNbot: ##### Error inesperado en main.

COMANDOS ADMINISTRADOR

Aclaraciones:

Las palabras en negrita son palabras clave, por ende SIEMPRE se envían igual al momento de realizar el comando. Las palabras subrayadas NO siempre se envían igual, esto depende de la ocasión. Cuando el administrador envíe un comando SIEMPRE espere una respuesta antes de enviar otro.

Comandos:

- Carga:

Cancelar carga: **Carga** + número de operación + **cancelar**

Carga recibida: **Carga** + número de operación + **recibido** + monto

- Descarga:

Cancelar descarga: **Descarga** + número de operación + **cancelar**

Descarga transferida: **Descarga** + número de operación + **transferido**

- Consultar fichas en panel de administrador: **Fichas panel**
- Cambiar datos bancarios para pasos de transferencia: **Banco** + cbu + alias + nombre de cuenta
- Mantenimiento: **Mantenimiento**. Este estado se utiliza cuando las plataformas se encuentran en mantenimiento haciendo que Sofía notifique la situación a los jugadores que le escriben.

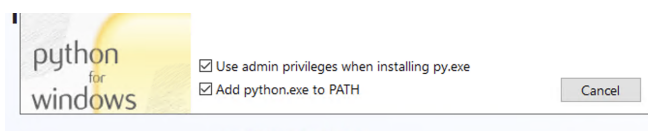
PROCESO LEVANTAR Y CORRER PROYECTO

Requerimientos previos para iniciar:

1. Servidor VPS con un mínimo de 8 GB de RAM con sistema operativo Windows Server 2022.
2. Tener el código fuente del proyecto listo para importar o descargar.

Paso a paso:

1. Instalar en la máquina la última versión de Python (<https://www.python.org/downloads/>). Al momento de instalar debemos verificar que tengamos chequeado la opción **Add python.exe to PATH** para que esto pueda funcionar de forma correcta.



2. Abrimos la terminal de nuestro equipo y en la carpeta raíz correremos el siguiente comando para instalar las dependencias necesarias: primero ejecutamos **install pip** y luego **pip install selenium pandas numpy mysql-connector-python unicode pyautogui**.
3. Descomprimos nuestra carpeta del proyecto dejándola disponible en escritorio para facilitar el acceso.
4. Configuramos datos iniciales:

Para poder iniciar el proyecto correctamente, previamente necesitamos de ciertos datos iniciales los cuales agregaremos en el archivo que se encuentra dentro del proyecto con el nombre **config.py** donde veremos lo siguiente y completamos con nuestros datos aquello que se nos pide en este archivo:

- URL de la plataforma.
- Usuario y contraseña panel plataforma.
- Numero superadministrador.
- Nombre de bot.
- Datos de transferencia.
- User config (solo gama planeta, este nombre de usuario debe comenzar con la letra A, este usuario deberá ser creado si no existe uno que inicie con la letra A).

Ejemplo de **config.py** a continuacion:


```

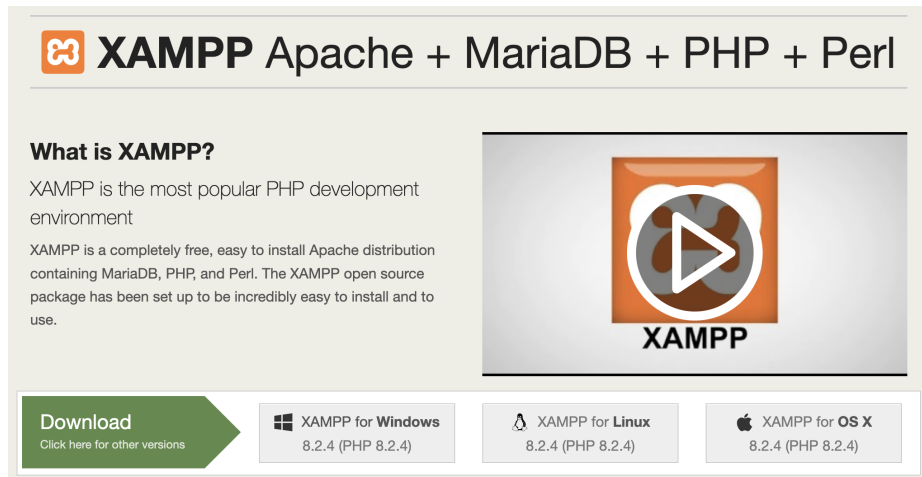
1  # SOLO GAMA PULPO
2  def datos_plataforma():
3      web_admin = "https://admin.pulpobet.com"
4      web_jugador = "https://pulpobet.com"
5      return web_admin, web_jugador
6
7  def cuenta_loggeo():
8      usuario = "botprueba33"
9      contrasena = "Hola123"
10     return usuario,contrasena
11
12  # Valores de inicializacion. SIEMPRE DEBEN CONFIGURARSE DESDE ACA. #
13  def numero_superadministrador():
14     numero = "+54 9 3516 18-5534"
15     return numero
16
17  def nombre_robot():
18     nombre = "BOTINO"
19     return nombre
20
21  # Valores de inicio predeterminados. LUEGO DEBEN CAMBIARSE POR COMANDO DE ADMINISTRADOR!
22  def datos_transferencia():
23     alias = "_alias_"
24     cbu = "_cbu_"
25     nombre_cuenta = "_nombre_"
26     return alias,cbu,nombre_cuenta

```

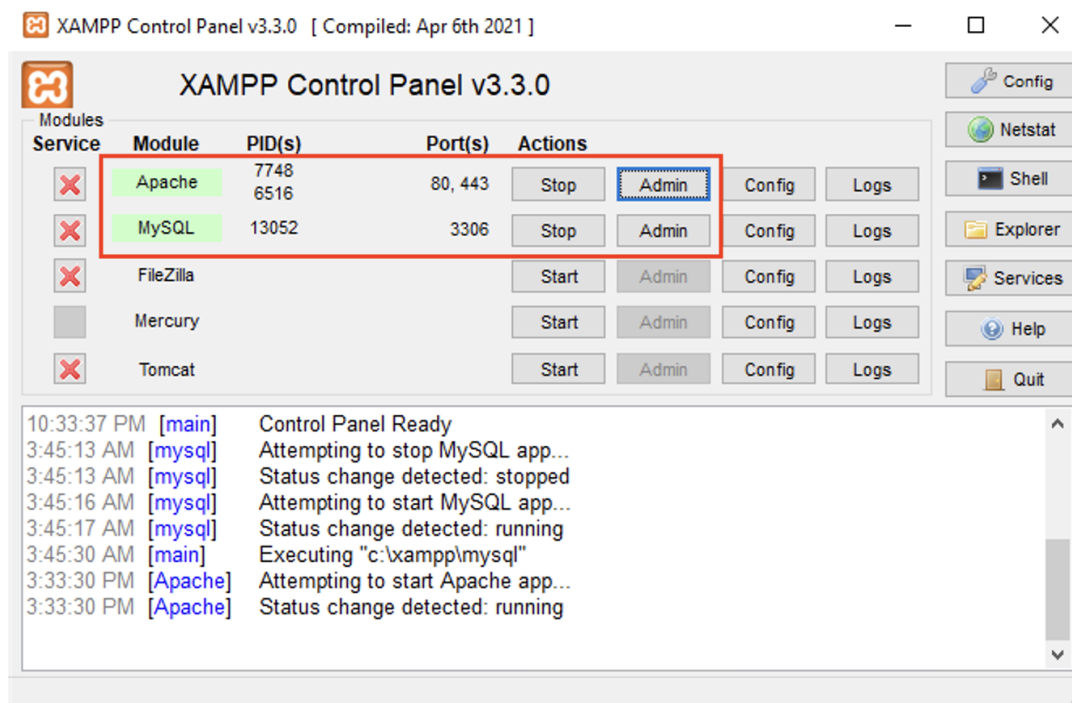
Como último dato de configuración inicial se debe agregar el número de SUPERADMINISTRADOR dentro del archivo **contactos_autorizados.txt** dentro de la carpeta **resource** del proyecto tal cual como figura en WhatsApp, ejemplo abajo.



5. Descargamos de la siguiente dirección [Apache Friends](#) la última versión de XAMPP para windows, esto nos permitirá tener nuestra base de datos corriendo.

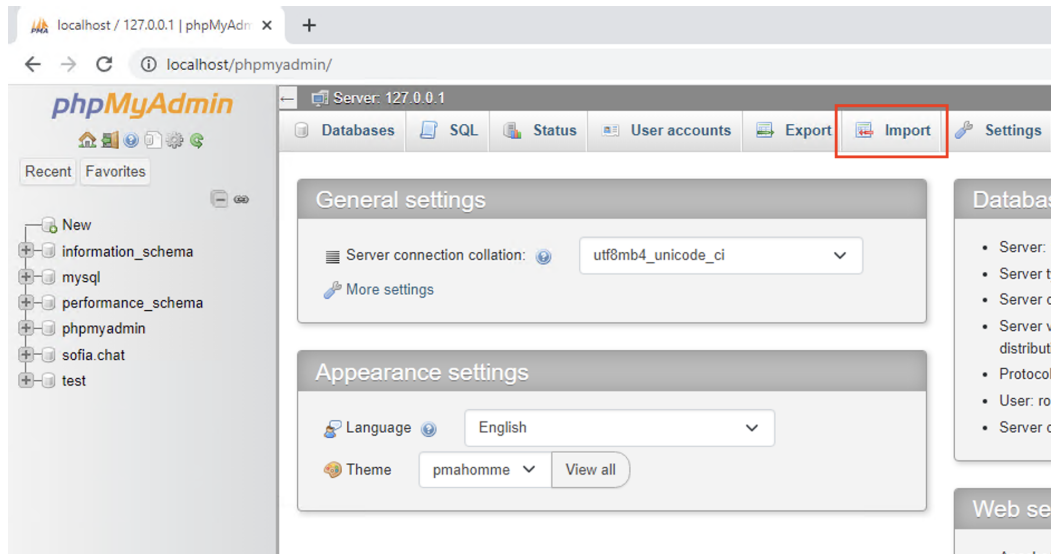


6. Una vez instalado XAMPP lo iniciamos y damos a **start** a **Apache** y **MySQL**.



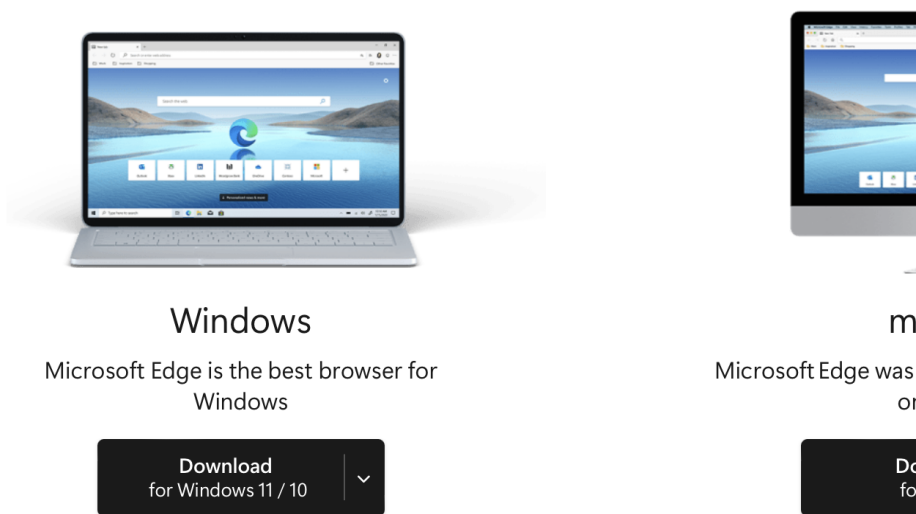
7. Ingresamos dentro del navegador a la ruta **localhost/phpmyadmin** (o ingresando directamente a **localhost** y buscando la sección de **phpmyadmin**) en donde nos llevará al panel de administrador de la base de datos.

Deberemos crear una nueva tabla con el nombre **sofia.chat**, luego de eso importamos el archivo que se encuentra dentro de la carpeta **resource** con la extensión de **.sql**.



IMPORTANTE: En lo que respecta a la importación que realizamos hacia la base de datos, este proceso se realizará una sola vez. Luego de esta importación podremos frenar y volver a iniciar la base de datos sin problema pero no volver a importar la colección ya que si realizamos esto perderemos todos nuestros datos.

8. Descargar e instalar el navegador Microsoft Edge ([Download Microsoft Edge](#)).



9. Descargar e instalar el navegador Google Chrome ([Google Chrome - Download the Fast, Secure Browser from Google](#)).
10. Descargar Web Driver para Edge validando la versión y descargando el correspondiente a la misma, instructivo a detalle en página oficial: [Edge Webdriver](#).

11. Descargar Web Driver para Chrome validando la versión y descargando el correspondiente a la misma en [ChromeDriver](#), descargando la versión de windows disponible. Es importante validar su versión en ambos navegadores, esta información la veremos en detalles del navegador.



Current Releases

- If you are using Chrome version 114, please download [ChromeDriver 114.0.5735.90](#)
- If you are using Chrome version 113, please download [ChromeDriver 113.0.5672.63](#)
- If you are using Chrome version 112, please download [ChromeDriver 112.0.5615.49](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.







If you are using Chrome from Dev or Canary channel, please following instructions on the [ChromeDriver Canary](#) page.

For more information on selecting the right version of ChromeDriver, please see the [Version Selection](#) page.

ChromeDriver 114.0.5735.90

Supports Chrome version 114

Index of /113.0.5672.63/

	Name	Last modified	Size	ETag
	Parent Directory		-	
	chromedriver_linux64.zip	2023-05-03 09:24:07	6.98MB	b64b1d3b8fe10d635dc5775965ca8048
	chromedriver_mac64.zip	2023-05-03 09:24:10	8.81MB	5597f9231804384ee2345bba18584cca
	chromedriver_mac_arm64.zip	2023-05-03 09:24:13	8.06MB	f91bb4f92a44e26af2c38d71b1876bfa
	chromedriver_win32.zip	2023-05-03 09:24:16	6.81MB	47853f34740941971c153b11365716ef
	notes.txt	2023-05-03 09:24:22	0.00MB	7a6ae80cd5a17d104faa80e953a140bf

12. Una vez descargados los drivers los ubicamos en la carpeta **driver** del proyecto, cada uno con su nombre **chromedriver** y **edgedriver**.
13. Para que el bot de avisos a los grupos, deberemos crearlos de forma manual desde el número en el que está corriendo el bot y agregando como miembros al superadministrador (si este lo desea), administrador (obligatorio) y responsable técnico (en grupo de reportes).
- Creamos los grupos desde el número que posee el bot tal cual describimos abajo.
 - Agregamos como miembro al administrador.
 - Configuramos en el grupo que solo puede enviar mensajes el creador del mismo, o sea el bot.
 - Silenciamos el grupo de reportes para que no interrumpa al administrador ya que este solo es de lectura.

Los grupos a crear son los siguientes:

- SOLICITUD DE CARGA
- SOLICITUD DE DESCARGA
- SOLICITUD ATENCION AL CLIENTE
- BOT REPORTES

IMPORTANTE, la creación de estos grupos deberá ser tal cual se describen.

14. Abrimos una nueva consola (terminal) ubicándonos nuevamente en la raíz del proyecto y corremos el archivo **main.py** con el comando **python main.py**.

Al ejecutar este archivo se nos abrirán dos navegadores, uno por vez, en donde nos pedirá escanear el QR de WhatsApp para ingresar con el número que corresponde, cuando escaneamos el mismo, volveremos a la consola a presionar ENTER para continuar con el siguiente navegador, escanear el QR y nuevamente presionar ENTER dando paso a la apertura de la plataforma.

En el caso de que tengamos error al abrirse los navegadores deberemos verificar que la versión del navegador sea la misma que la versión del webdriver.

GAMA PULPO:

Abierta la ventana con la plataforma una vez ejecutado `main.py`, deberemos realizar lo siguiente:

- En la parte superior donde vemos las páginas, seleccionar modo vista de tabla de jugadores con la mayor cantidad disponible, ejemplo 50 (dropdown).
- Con las teclas Ctrl + Menos (-) quitaremos zoom hasta el nivel 33% o cercano.
- Dejamos en primer plano esta ventana en pantalla completa posicionada arriba (inicio o barra de navegación).
- Habilitamos el botón "Incluir Bloqueados".

Quedándonos con una vista de la siguiente forma:

The screenshot shows the 'PulpBet Admin' interface. The top header is dark blue with the text 'PulpBet Admin' on the left and a user profile icon with the name '48.979' on the right. The left sidebar contains a list of navigation items: 'Dashboard', 'Usuarios', 'Finanzas', 'Reporte de Pago', 'Reporte Proveedores', and 'Mi Resumen'. The main content area is titled 'Usuarios' and features a search bar labeled 'Buscar Usuario', a dropdown menu for 'Tipo de Usuario' set to 'Todos', and a toggle switch for 'Incluir Bloqueados'. Below this is a table with the following data:

Usuario	Tipo	Saldo	
adm1	Jugador	0	+ -
adm2	Jugador	0	+ -
bot15min	Jugador	0	
dolar334	Jugador	21	+ -
jose21	Jugador	0	+ -
mat9010	Jugador	450	+ -

At the bottom right of the table, there is a pagination control showing 'Elementos por página: 50' and 'Página 1 de 1'. On the right side of the main content area, there is a search bar labeled 'Agente' and a dropdown menu showing 'OficinaBot1Bet'.

GAMA PLANETA:

- Todos los paneles de la gama de planeta, debe que existir un usuario que empiece con la letra A (para que este al principio) y tiene que cargarse al config.py en la variable user_config.

IMPORTANTE: CREAR UN USUARIO CON LETRA "A".

- Con las teclas Ctrl + Menos (-) quitaremos zoom hasta el nivel 33% o cercano.
- IMPORTANTE, en esta gama NO debe haber agentes en el panel, solo debe haber jugadores, si este tiene agentes puede no funcionar.

GAMA JUGALO:

- Con las teclas Ctrl + Menos (-) quitaremos zoom hasta el nivel 33% o cercano.
- Seleccionar si no lo está, el botón "Todos" en el área de la tabla de usuarios.

¡Listo para trabajar! Ahora solo queda hacer unas breves pruebas para comprobar que todo esté correctamente funcionando y ya podemos dejarlo corriendo. De lo contrario deberemos volver a los pasos o identificar el error para hacerlo que funcione.

IDENTIFICAR ERRORES Y REPORTE

Este apartado ha sido simplificado de manera que se puedan identificar de manera rápida y efectiva los errores y ver el estado del software accediendo al grupo de WhatsApp en el cual Sofía enviará estas alertas.

Funcionalidad y uso:

Sofía a medida que va realizando sus acciones, desde leer el chat, hasta cumplir con lo que el jugador o administrador solicitó, cada que ingresa a una nueva funcionalidad, internamente va registrando en que proceso / funcionalidad se encuentra. Si en el mismo proceso ha sucedido algún tipo de error, se registra este y se envía una alerta en el grupo de WhatsApp dando aviso y el detalle de cuál fue el último proceso que se realizó para poder ubicarlo dentro del código mediante vía grupo "SOFIA REPORTES" (para mayor organización) y realizar correcciones de ser necesarias.

Si deseamos corregir algún error o simplemente acceder al código para verificar su funcionamiento o causa del error, debemos abrir nuestro proyecto en el editor de preferencia y buscar en el proyecto el nombre de la función o línea que ha generado ese error, el cual nos llevara a donde se encuentra toda la lógica de ese proceso y verificar o corregir lo que debamos.

PERSONALIZACIÓN DE MENSAJES

Si deseamos personalizar uno o varios mensajes que realiza el software deberemos modificar estos directamente desde su código fuente. A continuación los pasos:

1. En el caso de que tengamos el proyecto sin funcionar ni corriendo en su entorno, omitiremos este punto. De lo contrario si tenemos el proyecto funcionando deberemos finalizar el **main.py**, menos la base de datos.
2. Una vez finalizado el **main.py**, deberemos ingresar en nuestra carpeta donde se aloja todo el código del proyecto y abrir el archivo **msjs.py**.
3. En este archivo se encontrarán todos los mensajes que envía el software, para ubicar el que queremos modificar simplemente lo buscaremos en base al texto que tiene este.
4. A la hora de editar solo lo haremos dentro de los strings (") y no modificaremos nada aquello que se encuentre entre corchetes ({}), ejemplo **{fichas_a_retirar}** en donde en el se colocaran los datos que necesitamos mostrar, en el caso ejemplo, se mostrará dependiendo del caso, la cantidad de fichas por retirar.

Recordemos que podemos agregar negritas en nuestros mensajes con el doble asterisco entre la palabra u oración que deseemos hacerlo, ejemplo " **** texto en negrita ****".

NO está permitido usar emojis en estos mensajes, los mismos no funcionarán cuando se envíen estos.

SOLO se pueden reemplazar palabra por otra palabra y NO agregar más oraciones.

SOLUCION ELEMENTO NO ENCONTRADO

Aclaración Estas causas de error siempre provienen por el cambio de nombre de un elemento o modificación del mismo, ya sea desde la plataforma o del mismo WhatsApp web.

Este tipo de error puede llevar a dos causas:

1 - Que WhatsApp haya cambiado sus clases en las cuales Sofía toma para ubicar los elementos desde el navegador. Este error lo podemos identificar fácilmente ya que cada vez que Sofía quiera acceder o leer los nuevos mensajes tendremos este error de forma recurrente. Si este es el caso, aplicaremos lo del apartado ACTUALIZAR CLASES WHATSAPP.

2 - En base a lo anterior, teniendo en cuenta que NO es recurrente el error de elemento no encontrado, debemos verificar en los mensajes que Sofía envía para chequear en qué proceso hemos tenido error, ya que este no es porque WhatsApp haya cambiado sus clases de los elementos, sino porque la plataforma ha cambiado de lugar un elemento que utiliza Sofía o ha renombrado uno de ellos. Si es este el caso, debemos verificar la clase o el xPath que tiene el elemento que utilizamos en ese proceso para poder actualizar en donde sea necesario.

ACTUALIZAR CLASES WHATSAPP

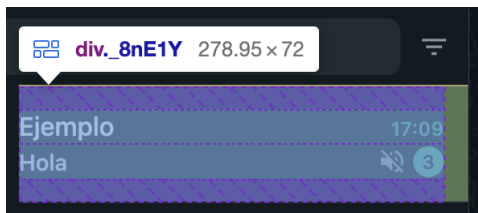
Para facilitar el proceso de actualización de las mismas se deja una referencia de dónde y qué clase corresponde actualmente a cada elemento que se utiliza desde WhatsApp para poder comprobar si ha cambiado alguna de estas (Cabe aclarar que el nombre de las clases abajo mencionadas ya pueden estar desactualizadas).

Estas clases de WhatsApp se encuentran sólo en los archivos **wp.py**, **wpEdge.py** y **main.py**. Se sugiere ver de antemano el código para mayor entendimiento.

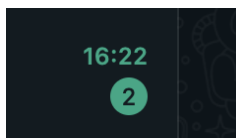
A continuación el nombre de la función y que elemento llama:

Dentro de **mani.py**:

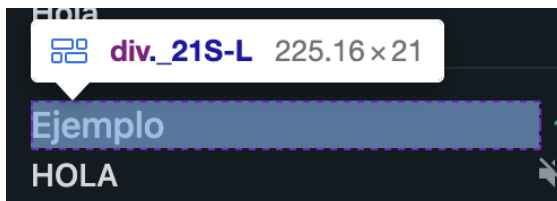
- Función **buscar_chats()**, variable **chats** tenemos la clase definida en (**By.CLASS_NAME**, "**_8nE1Y**") que equivale al elemento que vemos en la imagen, el cual es la "caja" en donde se encuentra el último chat:



- Función **buscar_chats()**, variable **chats_nuevos** tenemos la clase definida en (**By.CLASS_NAME**, "**_21S-L**") que equivale al elemento que vemos en la imagen, el cual hace referencia a si existe un chat nuevo:

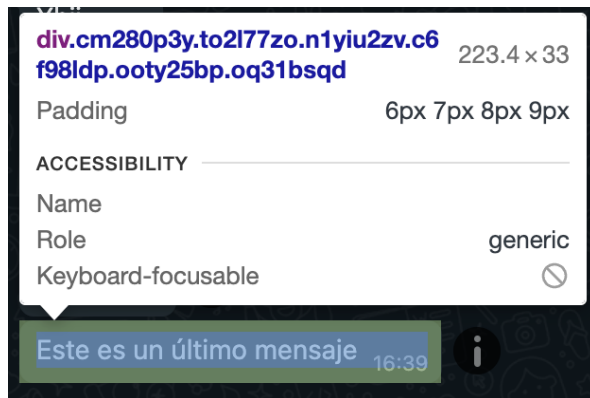


- Función **buscar_chats()**, variable **chat_numero** tenemos la clase definida en (**By.CLASS_NAME**, "**_21S-L**") que equivale al elemento que vemos en la imagen, el cual es el numero del cual recibimos el mensaje:

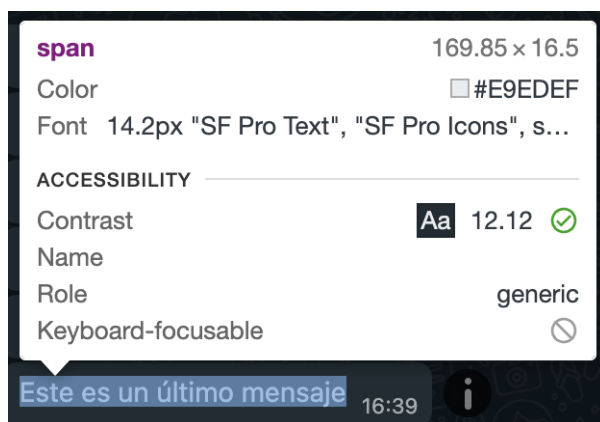


Dentro de **wp.py**:

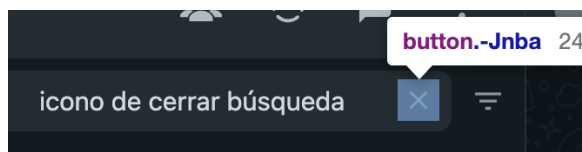
- Función **leer_ultimo_mensaje()**, variable **element_box_message** tenemos la clase definida en (**By.CLASS_NAME**, "**ItfyB**") que equivale al elemento que vemos en la imagen, el cual es la "caja" en donde se encuentra el último mensaje:



- Función **leer_ultimo_mensaje()**, variable **element_message** tenemos la clase definida en (**By.CLASS_NAME**, "**_21Ahp**") que equivale al elemento que vemos en la imagen, el cual corresponde al contenido del mensaje:



- Función **buscar_chat()**, variable **cancel** tenemos la clase definida en (**By.CLASS_NAME**, "**-Jnba**") que equivale al elemento que vemos en la imagen:



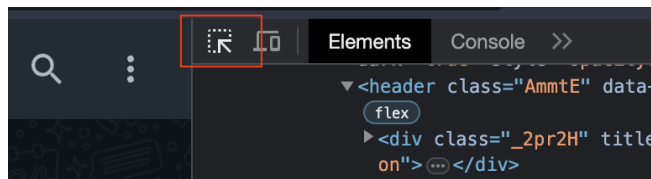
Dentro de **wpEdge.py** se hacen referencia a los mismos elementos pero NO poseen las mismas clases, el proceso de verificar es el mismo que **wp.py** solo que el nombre de las clases será distinto.

ACTUALIZAR CLASE

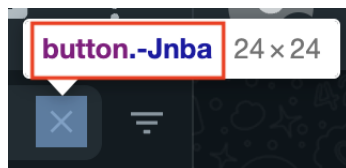
Este proceso es el mismo para cada uno de los elementos de WhatsApp, en este ejemplo lo haremos con el elemento `cancel` de la función `buscar_chat()`:

Veamos los pasos:

1. Abierta la ventana de WhatsApp web, ingresamos / abrimos DevTools. En el caso de Chrome las DevTools [tutorial aquí](#) y en el caso de Edge [tutorial aquí](#).
2. Hacemos click en el icono de flecha que vemos en la imagen:



4. Buscamos el elemento que queremos actualizar teniendo en cuenta la guía anterior.



5. Encontrado el elemento debemos ver que clase posee el mismo.



6. Con la clase que tiene este elemento ahora, debemos copiar ese valor de clase y pegarlo en nuestro código actualizado.
7. Verificar que esta actualización haya sido correcta corriendo el software nuevamente. De ser correcta la actualización de estas clases, no habrán problemas, de lo contrario seguirá fallando por motivo de que no actualizamos correctamente la clase.

LINK VIDEOS EXPLICATIVOS

DIANA

- [INICIAR](#)
- [ERROR VERSION WEBDRIVER](#)
- [DIANA ERROR EXTENSION DRIVE](#)

SOFIA

- [INICIAR ENTORNO](#)
- [ERROR VERSION WEBDRIVER](#)
- [ERROR EXTENSION DRIVER](#)
- [ERROR CAMBIO DE XPATH](#)