DIANA BOT DOCUMENTACIÓN

INDICE

STACK TECNOLOGICO

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.

PLATAFORMAS

El proyecto funciona para las siguientes plataformas con sus respectivos códigos. IMPORTANTE: memorizar bien los códigos de plataforma ya que deben ser proporcionados en cada comando.

- JUGA Y GANA ONLINE: 01
- SIEMPRE GANA: 02
- 24LIVE: 03
- CASINO365 ONLINE: 04

COMANDOS

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:

- Información completa:

INFO

- Configuración del bot:

START

- Carga (un jugador a la vez):

CAR + numero de plataforma + usuario + monto

- Descarga (un jugador a la vez):

DES + numero de plataforma + usuario + monto

- Carga múltiple (más de un jugador a la vez):

MULTIPLE + **CAR**

- Descarga (más de un jugador a la vez):

MULTIPLE + DES

- Crear nuevo jugador:

NJ + número de plataforma + nuevo usuario + nueva contraseña

- Crear nuevo agente:

NA + número de plataforma + nuevo usuario + nueva contraseña + comisión deportes(sin %) + comision casino(sin%)

- Consultar fichas en panel de administrador:

FICHAS + numero de plataforma

- Ejemplos comandos:

EJEMPLOS

- Codigos plataformas:

PLATAFORMAS

IDENTIFICAR ERRORES

Al momento de haber un error deberemos ingresar al servidor y chequear la información proporcionada por consola, ingresar en el código fuente y verificar si el error proviene de un elemento de WhatsApp, ruta xPath o error inesperado. Solucionar y volver a levantar el entorno.

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 Diana toma para ubicar los elementos desde el navegador. Este error lo podemos identificar fácilmente ya que cada vez que Diana 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 Diana 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 Diana 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

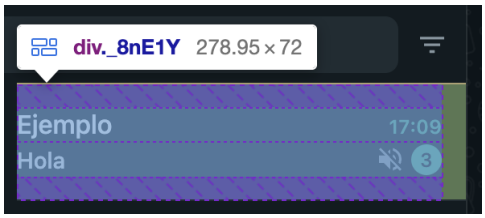
Para facilitar el proceso de actualización de las mismas se deja una referencia de dónde y qué clase corresponde actualmente a cada elemento que se utiliza desde WhatsApp para poder comprobar si ha cambiado alguna de estas (Cabe aclarar que el nombre de las clases abajo mencionadas ya pueden estar desactualizadas).

Estas clases de WhatsApp se encuentran sólo en el archivo **`main.py`**. Se sugiere ver de antemano el código para mayor entendimiento.

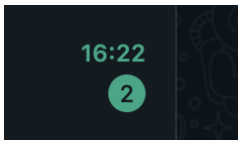
A continuación el nombre de la función y que elemento llama:

Dentro de **`mani.py`**:

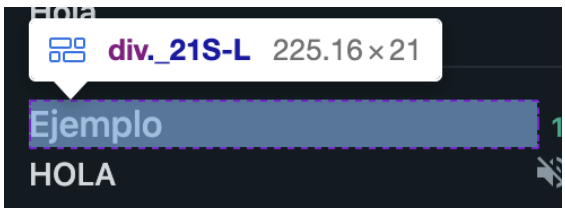
- Función **`buscar_chats()`**, variable **`chats`** tenemos la clase definida en **`(By.CLASS_NAME, "_8nE1Y")`** que equivale al elemento que vemos en la imagen, el cual es la "caja" en donde se encuentra el último chat:



- 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:



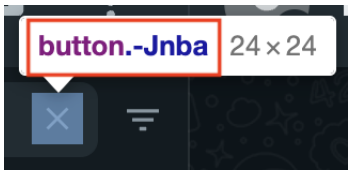
ACTUALIZAR CLASE

Este proceso es el mismo para cada uno de los elementos de WhatsApp, en este ejemplo lo haremos con el elemento **`cancel`** de la funcion **`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.

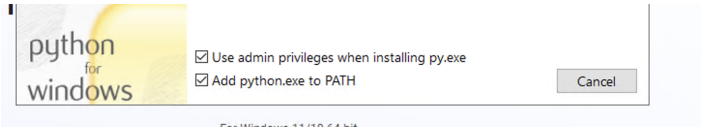
PROCESO LEVANTAR Y CORRER PROYECTO

Requerimientos previos para iniciar:

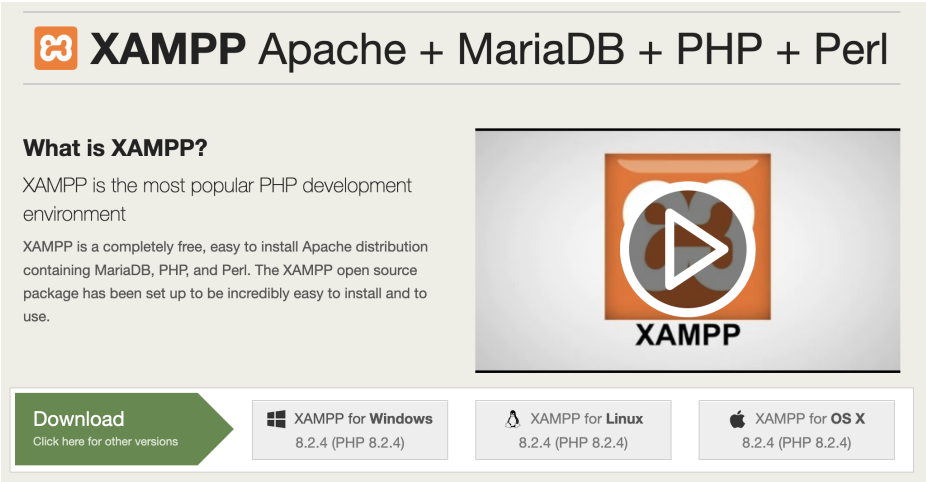
1. Servidor VPS con un mínimo de 8 GB de RAM con sistema operativo Windows Server 2022.
2. Tener el código fuente del proyecto listo para importar o descargar.

Paso a paso:

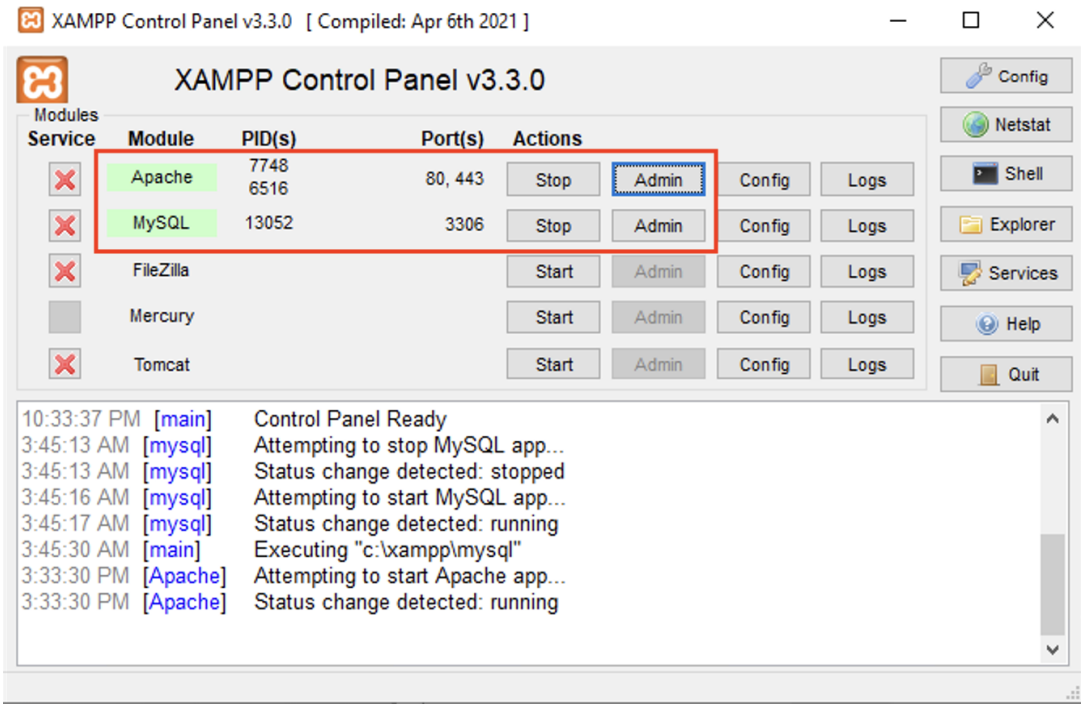
1. Instalar en la máquina la última versión de Python (https://www.python.org/downloads/). Al momento de instalar debemos verificar que tengamos chequeado la opción **`Add python.exe to PATH`** para que esto pueda funcionar de forma correcta.



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 unidecode pyautogui.`**
3. Descomprimimos nuestra carpeta del proyecto dejándola disponible en escritorio para facilitar el acceso.
5. Descargamos de la siguiente dirección Apache Friends la última versión de XAMPP para windows, esto nos permitirá tener nuestra base de datos corriendo.

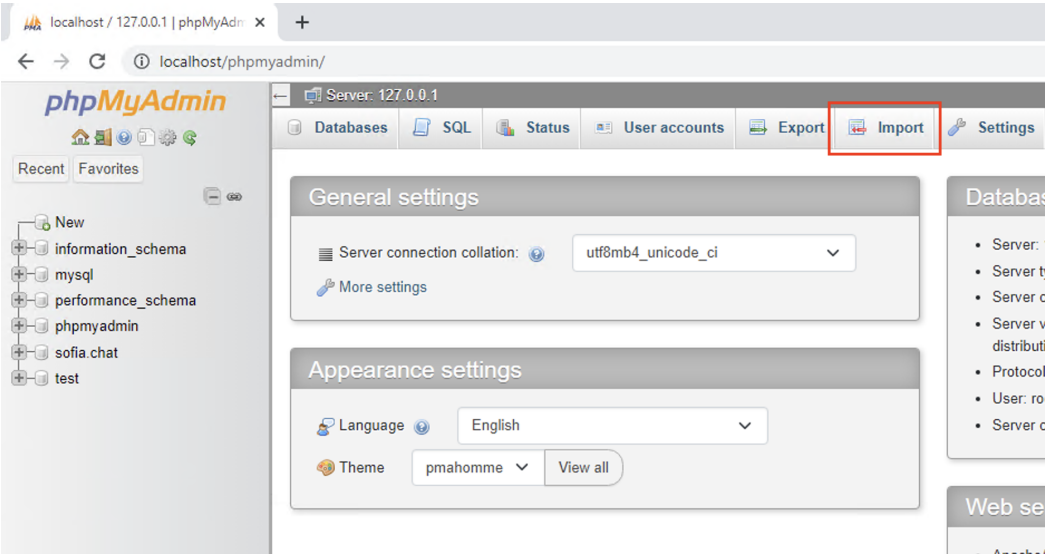


6. Una vez instalado XAMPP lo iniciamos y damos a **`start`** a **`Apache`** y **`MySQL`**.



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 **`diana.chat2`**, 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

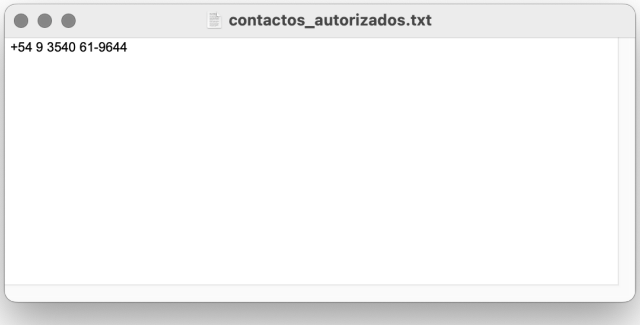
la base de datos sin problema pero no volver a importar la colección ya que si realizamos esto perderemos todos nuestros datos.

8. Como último dato de configuración tener en cuenta:

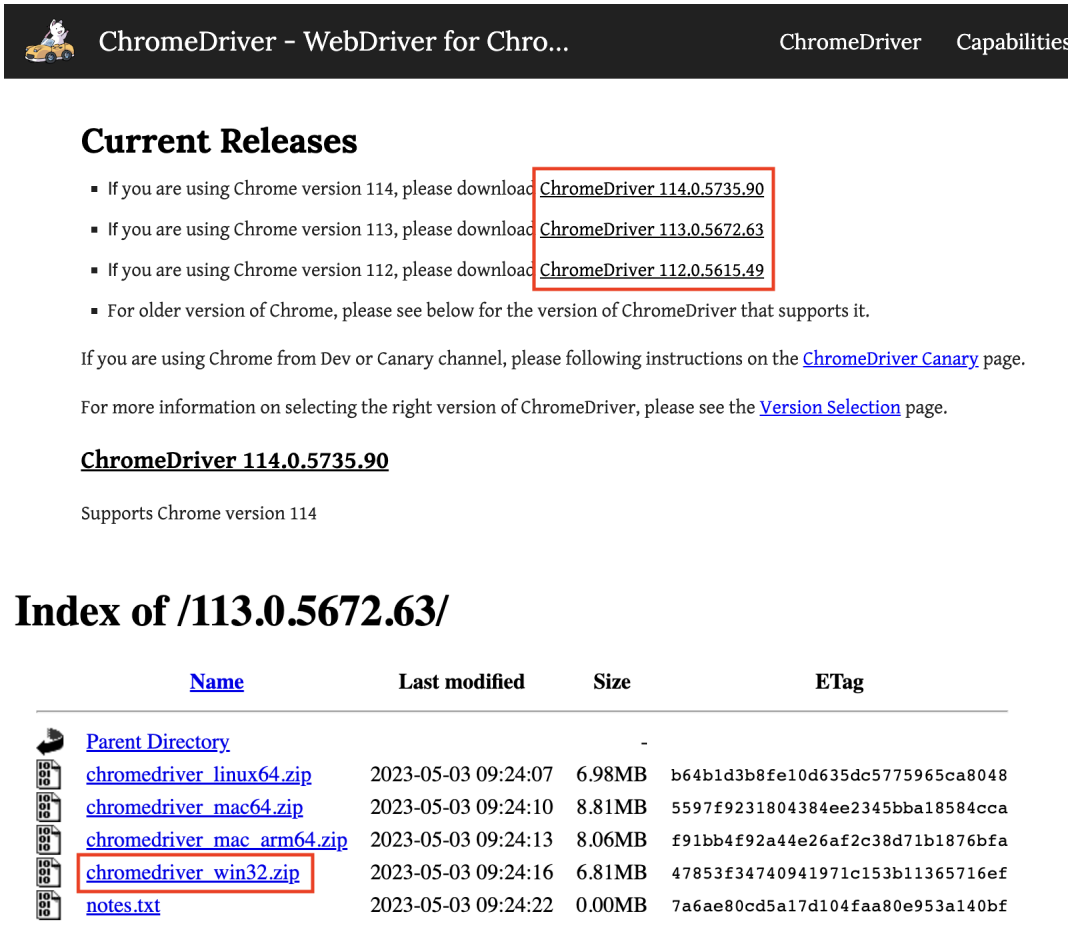
   Cada cajero que vaya a utilizar el servicio de diana **SIEMPRE** debe ser dado de alta **manualmente** en la tabla “cajeros” de la base de datos “diana.chat2”.

   Luego, dentro del archivo **`contactos_autorizados.txt`** dentro de la carpeta **`resource`** del proyecto, agregar el número del cajero correspondiente tal cual como figura en WhatsApp, ejemplo abajo.

   IMPORTANTE: de no cumplirse estos pasos Diana nunca responderá los mensajes del cajero.



9. Descargar e instalar el navegador Google Chrome (Google Chrome - Download the Fast, Secure Browser from Google).
10. 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.



12. Una vez descargado el driver lo ubicamos en la carpeta **`driver`** del proyecto con su nombre **`chromedriver`**.
14. Abrimos una nueva consola (terminal) ubicándonos nuevamente en la raíz del proyecto y corremos el archivo **`keepSession.py`** con el comando **`python keepSession.py`**. Escaneamos el QR que veremos en pantalla y luego corremos **`main.py`** con el comando **`python main.py`**.

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

¡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.