Documentación RPA

SAT - Harmony - CamiAPP

Índice

Información General

Resumen

El proyecto tiene como objetivo la automatización de procesos repetitivos, en este caso, la SAT, la plataforma Harmony y Cami APP. El enfoque está centrado en simplificar la interacción con elementos dinámicos de la página, y garantizar que las acciones en la interfaz se realicen correctamente sin intervención manual para agilizar los procesos. La automatización incluye procesos como el inicio de sesión, retención de facturas, la creación y modificación de documentos, y la verificación de datos de las interacciones con los elementos de la plataforma.

Cada parte de este flujo se maneja de manera modular, lo que permite su fácil adaptación y reutilización en otros procesos de automatización relacionados con las plataformas SAT, Harmony o Cami. La lógica subyacente está desarrollada en Python utilizando Selenium para manejar el navegador de manera eficiente, permitiendo realizar las interacciones como si fueran realizadas por un usuario humano.

Para la ejecución, se utiliza el navegador Brave, pero el código está diseñado de forma flexible para que pueda adaptarse a otros navegadores basados en Chromium si es necesario, ajustando mínimamente la configuración de Selenium. Las pruebas se realizan con la integración de controles de flujo robustos, manejando excepciones, errores y condiciones especiales, como tiempos de carga largos de la página o la falta de elementos esperados.

La estructura del código está organizada en módulos independientes, donde cada uno maneja una tarea o conjunto de tareas específicas. Además, se integran prácticas de logging para tener un registro detallado de cada paso del proceso, lo que facilita la identificación de errores y el monitoreo de la ejecución.

Notas importantes

1. Los archivos llevan el nombre de la plataforma que se está automatizando, seguido de la sección o módulo de la misma. Los archivos que pertenecen a la misma plataforma se encuentran en la carpeta **modulos_(plataforma)**.

2. Para modificar el flujo general del bot, solo es necesario cambiar el orden o comentar cada flujo en el archivo main.py.

3. El archivo constantes.py incluye todos los datos genéricos utilizados en los campos de entrada de los flujos de cada módulo, que pueden modificarse si es necesario, estos solamente están en caso de prueba o cuando no necesitan ser cambiados frecuentemente (p. e. la URL de la Agencia Virtual SAT) ya que los valores reales se ingresan llamando a la función desde el API.

4. Es necesario crear un archivo .env con las credenciales de inicio de sesión de la página, utilizando la estructura del archivo .env.example.

5. Todo el flujo se registra tanto en la terminal como en un archivo llamado log_RPA_(fecha)_(hora).log en la carpeta logs (esta carpeta se crea automáticamente en caso de que no exista). Existen varios niveles de registro (log) dependiendo de lo que esté ocurriendo:

- **INFO:** Muestra los pasos que está siguiendo el bot, los datos encontrados o ingresados.

- **WARNING:** Indica que el bot no siguió el flujo exactamente como estaba definido, pero puede continuar. Este mensaje puede aparecer, por ejemplo, cuando se busca un elemento que no está visible (como un botón de "Guardar" que solo aparece al realizar cambios, y estos no se hicieron). No necesariamente implica un error en el código o en la página. Cada *warning* describe la razón por la que fue generado. Es importante revisarlos, aunque en la mayoría de los casos no afectan al flujo ni indican problemas graves en la página.

- **ERROR:** Este mensaje se muestra cuando el bot no puede seguir el flujo normal debido a algo imprevisto o que no se consideraba posible. En algunos casos, puede impedir que el bot continúe con el flujo. Estos mensajes suelen aparecer si ha habido cambios en la página, por lo que siempre deben revisarse para identificar posibles ajustes necesarios.

- **CRITICAL:** Este mensaje aparece únicamente cuando el bot queda "atorado" en alguna página. Por ejemplo, si al iniciar sesión las credenciales

son inválidas y no se puede acceder al menú de la página se genera el mensaje CRITICAL. Es raro que ocurra, pero cuando sucede, la automatización se detiene automáticamente y libera los recursos. Siempre es necesario revisar estos mensajes para resolver el problema y permitir que el flujo continúe correctamente.

5. En caso de que se muestre un error de este tipo (stacktrace) en el log o terminal:

Este error significa que el bot no pudo encontrar o acceder a algún elemento del HTML de la página y no se está manejando bien esa excepción, en el código actual se intentó cambiar este tipo de errores por solamente los errores del log que se vieron anteriormente, por lo que suelen aparecer en casos no previstos durante la creación del código. Este tipo de errores son buenos indicadores de que algo en la página ha cambiado desde el momento de creación del código o no se tuvo en cuenta un caso específico con ciertos datos ingresados.

Requisitos de Instalación

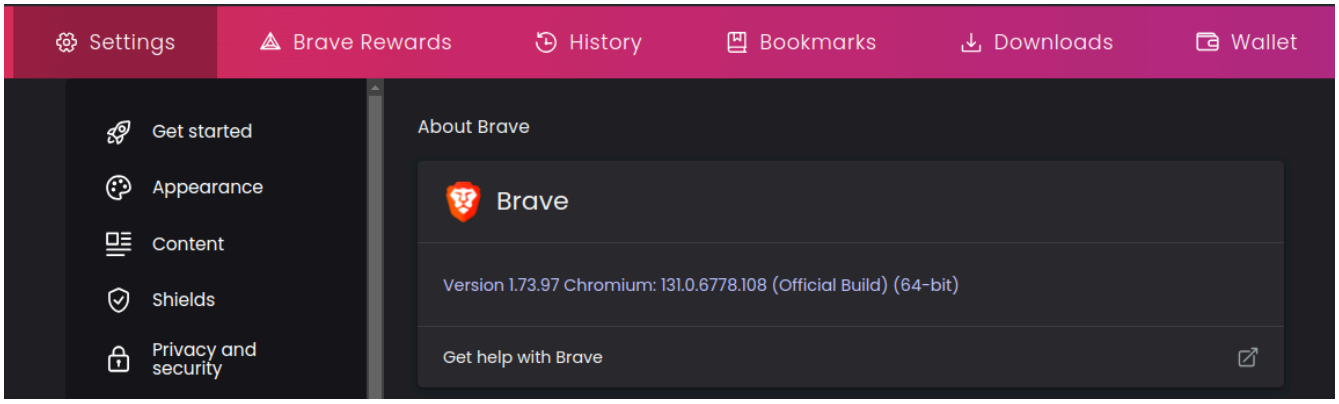
Para ejecutar este script se deben cumplir los siguientes requisitos:

- **Sistema Operativo**: Cualquier distribución de Linux o Windows 10. Para la creación se utilizó Ubuntu 24.04 (pero con algunos cambios se puede ejecutar en Windows).

- **Python**: Versión 3.12.3 o superior.

- **Selenium**: 4.6.0 (Se encuentra en el archivo requirements.txt)

- **WebDriver**: Descarga el WebDriver correspondiente a la versión de Chromium de tu navegador Brave: https://googlechromelabs.github.io/chrome-for-testing/



Descarga un zip, lo único importante es el archivo chromedriver.exe; su ruta, al igual que la del navegador instalado, debe definirse en la primera sección del archivo constantes.py, el chromedriver puede ir dentro de la carpeta drivers del programa. Se recomienda verificar las rutas específicas de la computadora y modificar estos datos si es necesario. Actualmente el código cuenta con el chromedriver para Chromium 131.0. Si el navegador tiene esa versión no es necesario hacer cambios.

Archivos Principales

constantes.py

Este es el archivo donde se deben modificar los datos. Está dividido en secciones con los nombres de los archivos que utilizan esas constantes:

```python
# -x- modulo_emision_constancias_de_retencion.py -x-

S_EMISION_CONSTANCIAS_EMISION_DEL = "04/12/2024" # Opcional
S_EMISION_CONSTANCIAS_EMISION_AL = "20/12/2024" # Opcional
S_EMISION_CONSTANCIAS_RETENCIONES_QUE_DECLARA_IVA = 1
S_EMISION_CONSTANCIAS_REGIMEN_GEN = 1
S_EMISION_CONSTANCIAS_TIPO_DOCUMENTO = 1
S_EMISION_CONSTANCIAS_NIT_RETENIDO = "12345678"
S_EMISION_CONSTANCIAS_NO_AUTORIZACION_FEL = "" # Opcional
S_EMISION_CONSTANCIAS_SERIE_DE_FACTURA = "" # Opcional
S_EMISION_CONSTANCIAS_NO_DE_FACTURA = "1234567890"
S_EMISION_CONSTANCIAS_DIRECTORIO_DESCARGAS = r"C:\Users\ads_edgar.menendez\Downloads"
S_EMISION_CONSTANCIAS_DIRECTORIO_FACTURAS_IVA = r"C:\Users\ads_edgar.menendez\Documents"
S_EMISION_CONSTANCIAS_NOMBRE_PROVEEDOR = "Kevin"
S_EMISION_CONSTANCIAS_FECHA_FACTURA = "20/12/2024"

S_EMISION_CONSTANCIAS_RETENCIONES_QUE_DECLARA_ISR = 2
S_EMISION_CONSTANCIAS_DIRECTORIO_FACTURAS_ISR = "/home/diego/Music"

# Datos de prueba que cambian luego para el caso 3 (pequeño contribuyente).
S_EMISION_CONSTANCIAS_REGIMEN_PEQ = 2

# Variables de impresión (función pendiente):
S_EMISION_CONSTANCIAS_NOMBRE_IMPRESORA = "RICOH_MP_C307_002673EBD502"
S_EMISION_CONSTANCIAS_CANTIDAD_COPIAS = 3
```

Dado que el archivo constantes.py contiene muchos datos, se recomienda ejecutar el flujo correspondiente para identificar en qué momento se utilizan y modificarlos según sea necesario.

Notas:

1. Dentro del archivo, cada dato importante es comentado con su propósito específico.
2. Los datos suelen procesarse de arriba hacia abajo, según el flujo y su nombre es bastante específico para que sección de la plataforma se está utilizando.

3. No es posible ordenarlas por pasos o flujos debido a que algunas se utilizan en repetidas ocasiones.
4. Los datos que tienen como valor un número entero representan índices de las listas desplegables de las páginas web (select), empiezan por 1.
5. Los datos que tienen como valor una fecha, tienen el siguiente formato: dd/mm/yy
6. Es necesario cambiar las constantes de directorios dependiendo del SO.
7. No todos las variables son necesarias, por ejemplo, algunas son para filtros de búsqueda, las variables de este tipo llevan el comentario indicándolo.

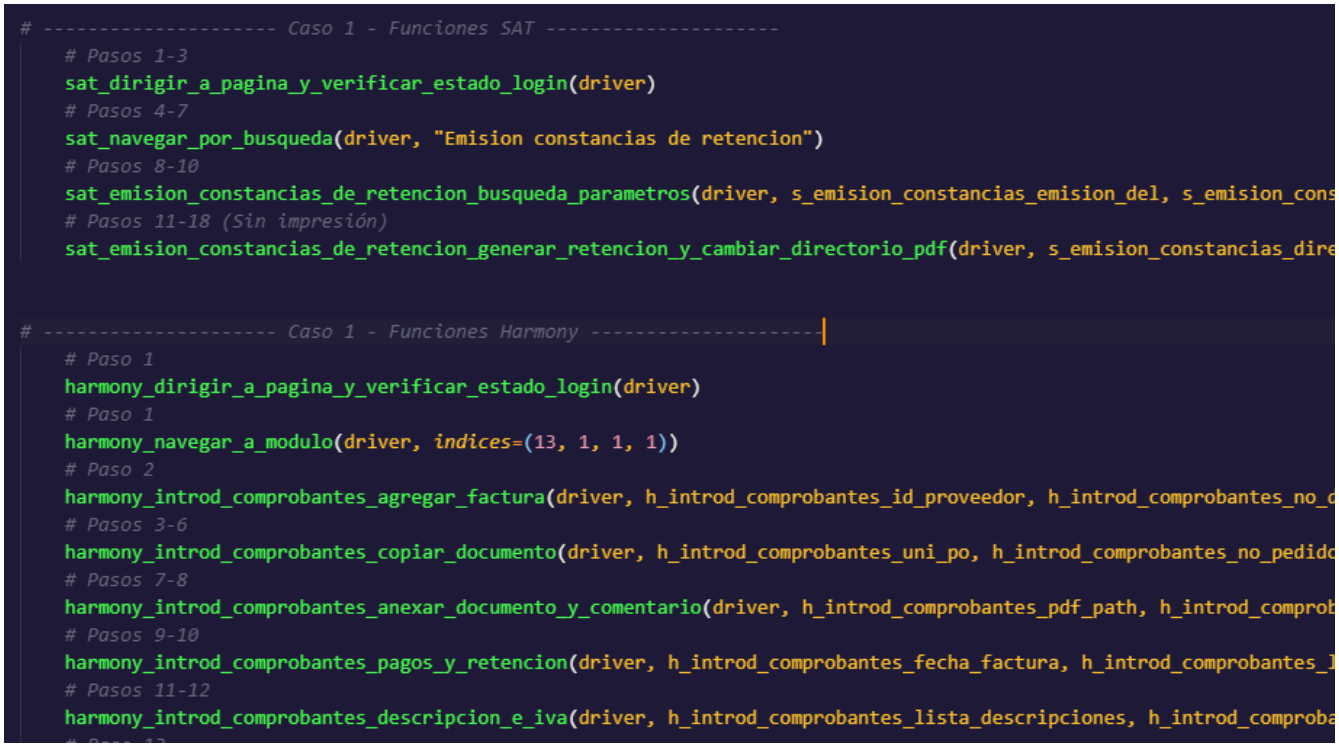
configuracion_bot_y_flujos.py

1. Flujos

Este archivo centraliza las herramientas y flujos necesarios para ejecutar la automatización. A continuación, se describen las secciones principales:

Se importan las bibliotecas necesarias y todos los módulos específicos del proyecto para la automatización.

Incluye funciones generales, por ejemplo, **caso_1_reten_IVA_GEN**, que contienen las funciones para cada flujo (caso):



```
# ---------------------- Caso 1 - Funciones SAT ----------------------
    # Pasos 1-3
    sat_dirigir_a_pagina_y_verificar_estado_login(driver)
    # Pasos 4-7
    sat_navegar_por_busqueda(driver, "Emision constancias de retencion")
    # Pasos 8-10
    sat_emision_constancias_de_retencion_busqueda_parametros(driver, s_emision_constancias_emision_del, s_emision_cons
    # Pasos 11-18 (Sin impresión)
    sat_emision_constancias_de_retencion_generar_retencion_y_cambiar_directorio_pdf(driver, s_emision_constancias_dire


# ---------------------- Caso 1 - Funciones Harmony ----------------------
    # Paso 1
    harmony_dirigir_a_pagina_y_verificar_estado_login(driver)
    # Paso 1
    harmony_navegar_a_modulo(driver, indices=(13, 1, 1, 1))
    # Paso 2
    harmony_introd_comprobantes_agregar_factura(driver, h_introd_comprobantes_id_proveedor, h_introd_comprobantes_no_c
    # Pasos 3-6
    harmony_introd_comprobantes_copiar_documento(driver, h_introd_comprobantes_uni_po, h_introd_comprobantes_no_pedido
    # Pasos 7-8
    harmony_introd_comprobantes_anexar_documento_y_comentario(driver, h_introd_comprobantes_pdf_path, h_introd_comprob
    # Pasos 9-10
    harmony_introd_comprobantes_pagos_y_retencion(driver, h_introd_comprobantes_fecha_factura, h_introd_comprobantes_l
    # Pasos 11-12
    harmony_introd_comprobantes_descripcion_e_iva(driver, h_introd_comprobantes_lista_descripciones, h_introd_comproba
```

Estas son las funciones de cada flujo (las mismas funciones se encuentran en el archivo con el mismo nombre). Desde aquí se define el orden de ellas. Si no se quiere ejecutar una función, se puede comentar en esta sección; sin embargo, **NO** es recomendable si no se entiende por completo que hace cada función. El comentario indica que serie de pasos ejecuta esa función del archivo "PDD_Recepción de factura_V01".

2. Clase CriticalHandler

Sirve por si se registra un error crítico (CRITICAL) en el log, detiene la ejecución del programa. Intenta cerrar el navegador en caso de un fallo grave, garantizando que no queden procesos abiertos. Esto fue necesario implementarlo ya que naturalmente Selenium no se detiene hasta terminar de ejecutar todo el script, incluso si se encuentra atorado en alguna página, por ejemplo, si no se logró iniciar sesión se muestra un error CRITICAL terminando con la automatización ya que no hay forma de que pueda continuar.

3. Configuración del Logging

configurar_logging(driver=None): Configura el registro de logs para monitorear la ejecución:
Guarda los logs en un archivo (log_RPA_(fecha)_(hora).log) y los imprime en la terminal.
Incluye un manejador personalizado (CriticalHandler) para detener la ejecución si ocurre un error crítico.
Formato de los logs: timestamp - nivel - mensaje.

4. Configuración del WebDriver

configurar_driver(): Configura el navegador Brave para la automatización:
Usa el CHROMEDRIVER_PATH y BRAVE_BINARY_PATH definidos en las constantes.
Maximiza la ventana del navegador al iniciar (Es importante que se muestre en pantalla completa, ya que en ocasiones el bot no logra acceder a componentes de la página que no están cargados).
Inicia el navegador sin ninguna página web, ya que el bot ingresa a la página desde las funciones con el nombre (modulo)_dirigir_a_pagina… de cada sección del flujo.

Main.py

En este archivo comienza la automatización. Primero llama a las funciones del archivo de configuración (logging y driver). Luego inicia el flujo que no esté comentado llamando a la función general (del archivo de configuración) que incluye todas las funciones que sigue el bot.

Finalmente, se detiene el bot, cerrando el navegador y liberando los recursos utilizados, concluyendo así la automatización.

Archivos SAT

Notas:

1. Para continuar con la lectura se recomienda comparar cada uno con los pasos de cada flujo del archivo “PDD_Recepción de factura_V01” e ir navegando por las diferentes páginas web de cada módulo.
2. Esta sección se divide en archivos y cada una de sus funciones, las que dicen “función auxiliar” no se encuentran en la configuración del flujo ya que no siempre se utilizan, se utilizan varias veces o son para manejar algunos componentes complejos (p. e. el calendario desplegable) y no saturar la función original.
3. Abajo de cada función se indica que datos del usuario utiliza, algunos deben ser ingresados o no (dependiendo de lo que se indique en el archivo constantes.py), por ejemplo, la URL de la página (no se debería cambiar con frecuencia), el número de factura (cambia cada ejecución). No indica las otras variables que se utilizan solo dentro de la función.

sat_login.py

sat_dirigir_a_pagina_y_verificar_estado_login(driver)

Variables: SAT_URL

Esta función verifica si ya hay una sesión activa en la página de la SAT. Si la sesión no está activa, carga la página de la SAT y realiza el inicio de sesión. Los pasos son:

1. Carga la página de la SAT y espera un máximo de 5 segundos para la carga.

2. Si la página no carga dentro de ese tiempo, se genera un error crítico, ya que aunque cargue después, la conexión será mala y el bot se puede “atorar” en alguna página, el error indica que se vuelva a intentar.

3. Si la URL contiene "login", significa que no hay sesión activa, por lo que llama a la función sat_login para realizar el inicio de sesión.

4. Si la URL no contiene "login", indica que la sesión ya está activa y no es necesario iniciar sesión.

sat_login(driver, username, password)

(Función auxiliar)

Variables: SAT_USER_EMAIL, SAT_USER_PASSWORD

Esta función se encarga de realizar el inicio de sesión en la página de la SAT con las credenciales proporcionadas. Los pasos son:

1. Encuentra los campos de usuario, contraseña y el botón de inicio de sesión.
2. Ingresa las credenciales en los campos correspondientes y hace clic en el botón de login.
3. Espera 7 segundos para verificar si el inicio de sesión fue exitoso.
4. Si la URL sigue siendo de "login", genera un error crítico indicando que el inicio de sesión ha fallado.
5. Si todo es exitoso, se registra que el inicio de sesión fue exitoso.

sat_navegar_a_modulo.py

sat_navegar_por_busqueda

Variables: texto_busqueda

1. Ubica la barra de búsqueda del navbar de la página de la Agencia Virtual SAT.
2. Si no logra interactuar con ella recarga la página para regresar al “home”.
3. Una vez localizado, limpia el contenido del input e ingresa el nombre de la sección de la página que se quiere acceder.
4. Se despliega una lista con un elemento al que da click para dirigirse a la página.

sat_modulo_emision_constancias_de_retencion.py

sat_emision_constancias_de_retencion_busqueda_parametros

Variables: EMISION_CONSTANCIAS_EMISION_DEL, EMISION_CONSTANCIAS_EMISION_AL,

EMISION_CONSTANCIAS_RETENCIONES_QUE_DECLARA,
EMISION_CONSTANCIAS_REGIMEN,
EMISION_CONSTANCIAS_TIPO_DOCUMENTO,
EMISION_CONSTANCIAS_NIT_RETENIDO,
EMISION_CONSTANCIAS_NO_AUTORIZACION_FEL,
EMISION_CONSTANCIAS_SERIE_DE_FACTURA,
EMISION_CONSTANCIAS_NO_DE_FACTURA

Esta función se encarga de realizar la búsqueda de constancias de retención en el sistema SAT, interactuando con varios campos del formulario para ingresar parámetros de búsqueda. Los pasos principales incluyen:

1. Selecciona una fecha de emisión desde un calendario. Usa un campo de entrada o un botón de calendario para seleccionar la fecha.

2. Ingresa una fecha límite en el campo correspondiente, adaptando el formato.

3. Selecciona una opción de un menú desplegable que especifica qué retenciones se declaran (IVA o ISR)

4. Selecciona una opción en un menú desplegable para especificar el régimen fiscal.

5. Selecciona un tipo de documento fiscal desde un menú desplegable.

6. Ingresa un número de NIT en el campo correspondiente.

7. Ingresa un número de autorización FEL (Opcional).

8. Ingresa el número de serie de la factura (Opcional).

9. Ingresa el número de la factura.

10. Hace clic en el botón de "Buscar" para enviar el formulario.

11. Marca un checkbox de confirmación de la fecha de emisión, utilizando JavaScript para garantizar la selección.

sat_emision_constancias_de_retencion_generar_retencion_y_cambiar_directorio_pdf

Variables: EMISION_CONSTANCIAS_DIRECTORIO_DESCARGAS, EMISION_CONSTANCIAS_DIRECTORIO_FACTURAS, EMISION_CONSTANCIAS_NOMBRE_PROVEEDOR, EMISION_CONSTANCIAS_NO_DE_FACTURA, EMISION_CONSTANCIAS_FECHA_FACTURA

1. La función intenta localizar el primer checkbox para seleccionar la factura utilizando una ruta XPath. Si no se encuentra, intenta con un segundo XPath. (Cada checkbox aparece dependiendo de los datos ingresados en la búsqueda por lo que intenta con ambos).

2. Una vez localizado el checkbox, la función hace scroll hasta él para asegurarse de que sea visible y luego lo selecciona.

3. Se obtiene el número de factura proporcionado como argumento y se asigna a la variable numero_de_factura_retenida.

4. La función intenta localizar el botón "Generar retención" en la página. Si no se puede hacer clic de manera estándar, intenta hacer clic utilizando JavaScript.

5. Después de generar la retención, la función espera brevemente a que aparezca una ventana emergente con el enlace a la constancia de retención.

6. Se obtiene el texto del enlace que contiene el número de la constancia de retención y el nombre del archivo PDF.

7. La función hace clic en el enlace para descargar el PDF que contiene la constancia de retención.

8. La función espera unos segundos para asegurarse de que el archivo se haya descargado correctamente.

9. Después de la descarga, la función hace clic en el botón "Cerrar" para cerrar la ventana emergente.

10. Finalmente, la función llama a cambiar_nombre_y_directorio_pdf para renombrar y mover el archivo PDF descargado a la ubicación adecuada.

cambiar_nombre_y_directorio_pdf

(Función auxiliar)
Variables: EMISION_CONSTANCIAS_DIRECTORIO_DESCARGAS, EMISION_CONSTANCIAS_DIRECTORIO_FACTURAS, EMISION_CONSTANCIAS_NOMBRE_PROVEEDOR, EMISION_CONSTANCIAS_FECHA_FACTURA

1. Se construye la ruta completa del archivo PDF descargado utilizando el nombre de la constancia de retención y la extensión `.pdf`. La ruta se forma uniendo el directorio de descargas y el nombre del archivo.

2. Se verifica si el archivo PDF descargado existe en el directorio de descargas utilizando `os.path.exists`. Si no se encuentra, se genera un error y se termina la ejecución de la función.

3. Se construye el nuevo nombre para el archivo PDF en función del número de factura retenida y el nombre del proveedor. Luego, se genera la nueva ruta en el directorio de facturas utilizando el nuevo nombre del archivo.

4. Se utiliza `shutil.move` para mover y renombrar el archivo PDF desde el directorio de descargas al directorio de facturas con el nuevo nombre. Se registra un mensaje de éxito indicando la nueva ruta del archivo.

sat_modulo_categoria_de_rentas.py

sat_categoria_de_rentas_busqueda_parametros

Variables: CATEGORIA_DE_RENTAS_NIT_RETENIDO, CATEGORIA_DE_RENTAS_PERIODO_DEL, CATEGORIA_DE_RENTAS_PERIODO_AL, CATEGORIA_DE_RENTAS_ESTADO_DE_ASIGNACION

1. Se intenta cambiar al iframe donde se encuentran los elementos del formulario, utilizando driver.switch_to.frame(). Si no se encuentra el iframe adecuado, se vuelve al contenido principal utilizando driver.switch_to.default_content() y se intenta nuevamente, esto es necesario porque en ocasiones los elementos dentro de diferentes contextos pueden no ser accesibles.

2. Se llama a la función seleccionar_fecha para seleccionar la fecha en el campo "PERIODO_DEL", pasando el driver y el XPath del campo de fecha, así como el valor de la fecha a seleccionar.

3. Similar al paso anterior, se utiliza la función seleccionar_fecha para seleccionar la fecha en el campo "PERIODO_AL".

4. Se localiza el campo de texto correspondiente al NIT retenido utilizando su ID. Si el campo es encontrado, se limpia y se ingresa el valor del NIT retenido.

5. Se hace clic en el label del campo "ESTADO_DE_ASIGNACION" para abrir la lista desplegable. Luego, se selecciona la opción correspondiente basándose en el valor de CATEGORIA_DE_RENTAS_ESTADO_DE_ASIGNACION, que se ajusta restando 1 a su valor para que coincida con el índice de la lista.

6. Se localiza el botón "BUSCAR" utilizando su ID y se hace clic en él. Luego, se espera un breve periodo para permitir que la tabla de resultados se genere.

seleccionar_fecha

(Función auxiliar)

1. Se localiza el campo de entrada de fecha y se hace clic en él para abrir el calendario.

2. La fecha proporcionada se divide en día, mes y año, y se utilizan estos valores para seleccionar la fecha en el calendario.

3. Se localiza y selecciona el año correspondiente en el calendario.

4. Se localiza y selecciona el mes correspondiente en el calendario.

5. Se recorren los elementos del calendario hasta encontrar el día correcto, que debe estar habilitado, y se hace clic sobre él.

sat_categoria_de_rentas_buscar_en_tabla

Variables: numero factura

1. Se localiza la tabla de facturas.

2. Una vez localizada la tabla, se obtienen todas las filas <tr> dentro de ella mediante el método `find_elements(By.TAG_NAME, "tr")` y se localiza donde está.

4. Se localizan las celdas (`<td>`) dentro de la fila, y se verifica si la fila tiene al menos 9 celdas. Si es así, se procede a extraer el número de factura, que está en la última celda (índice 8).

5. El número de factura extraído se compara con el número de factura recibido como parámetro (`numero_factura`). Si coinciden, se procede con la interacción.

6. Se localiza el checkbox en la primera columna de la fila usando el XPath.

7. Utilizando JavaScript, se hace clic en el checkbox.

8. Posteriormente, se localiza el icono del lápiz en la última columna de la fila y se hace clic en él usando el mismo método de JavaScript.

9. Si el número de factura no se encuentra, se registra un mensaje indicando que no se encontró la factura.

sat_categoria_de_rentas_asignar_categoria_y_regimen

Variables: CATEGORIA_DE_RENTAS_OPCION_CATEGORIA_DE_RENTA, CATEGORIA_DE_RENTAS_OPCION_REGIMEN

1. Primero, se selecciona la opción de categoría de renta en la lista de categorías.

3. Se verifica que el valor de la constante CATEGORIA_DE_RENTAS_OPCION_CATEGORIA_DE_RENTA esté dentro del rango de opciones disponibles. Si es así, se selecciona la opción correspondiente y se hace clic en ella.

4. Después de seleccionar la categoría, se presiona el botón OK para confirmar la selección.

5. A continuación, se selecciona el régimen, que corresponde al segundo desplegable de régimen.

6. Se verifica que el valor de la constante CATEGORIA_DE_RENTAS_OPCION_REGIMEN esté dentro del rango de opciones disponibles. Si es así, se selecciona la opción.

7. Se presiona el botón para asignar el régimen.

8. Después de asignar el régimen, se maneja la ventana emergente de confirmación. Si se muestra un botón de confirmación, se hace clic en él para confirmar la asignación. Si no, se verifica si aparece una advertencia de cambio de categoría, en cuyo caso se presiona el botón correspondiente para confirmar el cambio.

9. Finalmente, se espera un corto periodo de tiempo para que el proceso se complete.

Archivos Harmony

harmony_login.py

harmony_dirigir_a_pagina_y_verificar_estado_login

Variables: HARMONY_URL, HARMONY_USER_EMAIL, HARMONY_USER_PASSWORD

Esta función verifica si ya hay una sesión activa en la página de Harmony y, de no ser así, carga la página e inicia sesión. Los pasos son:

1. Establece un límite de tiempo para la carga de la página y entra en un bucle de reintentos para cargar la URL de Harmony.
2. Si la página no carga dentro del tiempo especificado, incrementa el contador de reintentos y vuelve a intentar cargar la página.
3. Después de alcanzar el número máximo de reintentos sin éxito, registra un error crítico y aborta el proceso.
4. Una vez cargada la página con éxito, restaura el tiempo de espera para cargas posteriores.
5. Verifica si la URL actual contiene la palabra "login". Si es así, significa que no hay sesión activa, por lo que llama a la función harmony_login para iniciar sesión.
6. Si la URL no contiene "login", se asume que ya hay una sesión activa y continúa sin realizar el login.

harmony_login

Variables: HARMONY_USER_EMAIL HARMONY_USER_PASSWORD

Esta función realiza el inicio de sesión en la página de Harmony con las credenciales proporcionadas. Los pasos son:

1. Espera a que aparezcan los campos de usuario y contraseña, y localiza el botón de inicio de sesión.
2. Ingresa el nombre de usuario y la contraseña en los campos correspondientes y hace clic en el botón de login.
3. Espera a que la URL cambie y ya no contenga "login", lo que indica que el inicio de sesión fue exitoso.
4. Si la URL sigue conteniendo "login" después de un tiempo razonable, registra un error crítico indicando un posible problema con la conexión o las credenciales.
5. Si se produce cualquier otra excepción inesperada durante el inicio de sesión, registra un error crítico con detalles del error.

harmony_navegar_a_modulo.py

harmony_navegar_a_modulo

Variables: indices

Esta función sirve para navegar en las diferentes secciones de la página utilizando la serie de listas desplegables del lado izquierdo de la página. “indices” es una tupla de números que indica el índice (contando de arriba para abajo) de cada lista desplegable. Y el último número sirve para dar click a esa opción y navegar a la sección. Se hizo de esta forma porque la barra de búsqueda no encuentra las secciones. Por ejemplo, si indices = (2, 3, 1), la función accede al segundo elemento en el primer nivel, luego al tercer elemento del submenú correspondiente, y finalmente al primer elemento del siguiente submenú.

1. Se verifica si existe el botón "Volver a la Página Home" (PT_WORK_PT_BUTTON_BACK):
    - Si se encuentra, se hace clic para regresar a la página de inicio y se espera 3 segundos.
    - Si no se encuentra, se asume que ya se está en la página de inicio.
2. Se localiza el árbol de navegación principal utilizando el XPath //*[@id="ptnav2tree"] y se guarda en current_ul.
3. Se itera sobre cada nivel de navegación definido en indices:
    - Para cada nivel i y su correspondiente índice index en indices:
        1. Se accede al elemento <li> en la posición index dentro del <ul> actual.
        2. Se hace scroll para llevar el elemento a la vista y se espera 1 segundo.
        3. Si no se encuentra en el último nivel de navegación:
            - Se localiza y se hace clic en la flecha de expansión (div[@class="ptnav2toggle"]) para desplegar la sublista.
            - Se espera 1 segundo para que se cargue la sublista.
            - Se actualiza current_ul con la nueva sublista <ul> encontrada dentro del elemento <li> actual.
        4. Si se encuentra en el último nivel:
            - Se busca el enlace <a> dentro del <li>.
            - Se hace scroll al enlace, se espera 1 segundo, y se intenta hacer clic en él.
            - Se espera hasta que se cargue un nuevo elemento en la página (usando un XPath de ejemplo para verificar la carga).

harmony_modulo_recepciones.py (OC)

harmony_recepciones_agregar_valor_y_busqueda_oc

Variables: h_recepciones_uni_po, h_recepciones_id_oc

Esta función corresponde al flujo del recepción de órdenes de compra, se encarga de agregar la unidad de recepción (UNI PO), buscar una orden de compra (OC) y finalmente confirmar la recepción en la interfaz de Harmony.

Pasos:

1. Cambia al iframe principal con ID ptifrmtgtframe.
2. Ingresa el UNI PO en el campo RECV_PO_ADD_BUSINESS_UNIT.
3. Hace clic en el botón "Añadir" (#ICSearch) y espera 2–3 segundos para que la siguiente pantalla se cargue.
4. Ingresa el ID de la OC en el campo PO_PICK_ORD_WRK_ORDER_ID.
5. Clic en el botón "Buscar" (PO_PICK_ORD_WRK_PB_FETCH_PO) y espera 2 segundos mientras se cargan las líneas de la orden.
6. Realiza scroll hasta el enlace "Selec Todo" (PO_PICK_ORD_WRK_SELECT_ALL_BTN) y hace clic para seleccionar todos los artículos.
7. Hace clic en el botón "Aceptar" (#ICSave) para confirmar las recepciones y concluir la operación en la página.

harmony_recepciones_guardar

Esta función presiona el botón "Guardar" en la sección de recepciones dentro de la interfaz de Harmony.

Pasos:

1. Localiza el botón "Guardar" (#ICSave) en la página utilizando un WebDriverWait.
2. Hace clic en el botón para confirmar y guardar los cambios en las recepciones.

harmony_modulo_introd_comprobantes.py (Facturas)

harmony_introd_comprobantes_agregar_factura

Variables: proveedor_id, no_factura, fecha_factura

Esta función corresponde al paso 2 del flujo de Harmony y se encarga de agregar una factura en la interfaz de Harmony. Para ello, navega al iframe principal, ingresa los datos del proveedor, número y fecha de la factura, y finalmente presiona el botón 'Añadir' para completar el registro.

1. Se cambia al iframe principal con ID ptifrmtgtframe para poder interactuar con los elementos de la página.
2. Se ingresa el ID del proveedor en el campo correspondiente (VCHR_ADDSRCH_VW_VENDOR_ID). Se realiza una pausa para permitir que PeopleSoft procese cualquier recarga automática tras ingresar el proveedor.
3. Se realiza una búsqueda preliminar del proveedor interactuando con el campo de número de factura (VCHR_ADDSRCH_VW_INVOICE_ID) y se ingresa un espacio en este campo. Esto inicia la búsqueda del proveedor.
4. Se ingresa la fecha de la factura en el campo VCHR_ADDSRCH_VW_INVOICE_DT. Se limpia el campo, se ingresa la fecha proporcionada y se hace una pausa para que el sistema procese el dato.
5. Se ingresa el número de factura en el campo VCHR_ADDSRCH_VW_INVOICE_ID. Se limpia el campo, se espera un momento y luego se ingresa el número de factura, seguido de una breve espera.
6. Se localiza y se hace clic en el botón "Añadir" utilizando su XPath //*[@id="#ICSearch"]. Una vez hecho el clic, se registra que la factura se ha añadido correctamente.

harmony_introd_comprobantes_copiar_documento

Variables: proveedor_id, no_factura, fecha_factura, iva_percent, factura_num, factura_serie

Esta función corresponde a los pasos 3 a 6 del flujo, copia comprobantes y genera una retención en Harmony. Sigue una serie de pasos para seleccionar factura, calcular impuestos y modificar datos de la factura, retornando al final una lista cuyo primer elemento es la suma del total de la factura más el IVA calculado.

1. Espera a que aparezca el componente con ID win0divVOUCHER_BUSINESS_UNIT para asegurar que la página siguiente ha cargado.
2. Selecciona la opción "N Recepción Ped" en el select VCHR_PANELS_WRK_VCHRWS_CPY_OPT.
3. Hace clic en el link "Ir".

4. Espera 2 segundos para que la nueva pantalla cargue.
5. Ingresa el UNI PO en el campo VCHR_PANELS_WRK_BUSINESS_UNIT_PO.
6. Ingresa el número de pedido en el campo VCHR_PANELS_WRK_PO_ID.
7. Hace clic en el botón "Buscar".
8. Espera a que se cargue la tabla de resultados.
9. Realiza scroll hasta el botón "Selec Todo" y hace clic en él.
10. Hace clic en el botón "Copiar Líneas Seleccionadas".
11. Espera 2 segundos a que cargue la página anterior.
12. Copia el total de la factura desde el campo VOUCHER_GROSS_AMT.
13. Parsea el total obtenido y calcula el IVA correspondiente usando iva_percent.
14. Ingresa el IVA calculado en el campo VOUCHER_VAT_ENTRD_AMT.
15. Suma el total de la factura con el IVA calculado y reingresa esta suma en el mismo campo VOUCHER_GROSS_AMT.
16. Ingresa el número de factura en el campo VCHR_UUID_SBF_UUID_AC_SBF.
17. Ingresa el número de serie en el campo VCHR_UUID_SBF_SERIES_AC_SBF.
18. La función finaliza retornando una lista cuyo primer elemento es la suma del total de la factura más el IVA calculado.

harmony_introd_comprobantes_anexar_documento_y_comentario

Variables: pdf_dir, pdf_filename, numero_factura, nombre_proveedor, comentario

Esta función corresponde a los pasos 7 y 8. Gestiona la carga de un documento PDF, genera una constancia de retención, y luego agrega un comentario correspondiente en la interfaz de Harmony. Los pasos seguidos son:

1. Hace clic en el enlace "Anexos (0)" desde la página principal para abrir la primera ventana emergente.
2. Cambia al iframe de la primera ventana emergente para interactuar con sus elementos, de lo contrario no puede ubicarlos.
3. En la primera ventana emergente, hace clic en el botón "Añadir Anexo" para abrir la segunda ventana emergente.
4. Cambia al iframe de la segunda ventana emergente para subir el archivo PDF.
5. Localiza el campo de carga de archivos, ingresa la ruta del PDF y hace clic en el botón "Cargar". Ya que el input es del tipo archivo no es necesario interactuar con ventanas del sistema operativo.
6. Espera 3 segundos para que el archivo se suba completamente y la ventana emergente se cierre.

7. Vuelve al iframe de la primera ventana emergente para continuar el proceso.
8. Intenta ingresar un texto de descripción en el campo "PV_ATTACH_WRK_ATTACH_DESCR$0" que incluye el número de factura y el nombre del proveedor.
9. Hace clic en el botón "Aceptar" para confirmar la acción.
10. Espera 3 segundos para que la acción se procese y la ventana se cierre, regresando a la página principal.
11. Cambia al iframe principal "ptifrmtgtframe" para continuar con la navegación.
12. Hace clic en la sección de comentarios "Coment(0)" para abrir la ventana emergente de comentarios.
13. Cambia al iframe de la ventana emergente de comentarios.
14. Ingresa el comentario proporcionado en el textarea "VOUCHER_DESCR254_MIXED".
15. Hace clic en el botón "Acep" para cerrar la ventana de comentarios y finalizar la operación.

harmony_introd_comprobantes_pagos_y_retencion

Variables: fecha_factura, total_retencion_por_articulo (lista con totales por artículo), porcentaje_retencion (lista con porcentajes de retención)

Esta función corresponde a los pasos 9 a 10 del flujo y gestiona el proceso de pagos y retenciones en Harmony. Navega a la sección de pagos, calcula la fecha de pago, ingresa valores de retención en múltiples páginas y finalmente regresa a la pantalla principal de la factura.

En el caso 1 no recibe datos en la lista total_retencion_por_articulo, en el caso 2 debe recibir en la lista los resultados que aparecen en la hoja de excel del documento del flujo, solo en el caso 3 se utiliza el return de la función harmony_introd_comprobantes_copiar_documento ya que regresa el total de la factura que se ingresa como primer elemento de la lista total_retencion_por_articulo y se ingresa en el mismo lado, se recomienda revisar el documento del flujo.

1. Cambia al iframe principal "ptifrmtgtframe".
2. Localiza la pestaña "Pagos" (id=ICTAB_1) de la parte superior de la página, hace scroll y realiza clic para entrar a la sección.
3. Espera para que la pestaña de pagos cargue completamente.
4. Calcula la fecha de pago sumando 30 días a fecha_factura y ajusta al siguiente viernes.
5. Ingresa la fecha de pago calculada en el campo PYMNT_VCHR_XREF_SCHEDULED_PAY_DT$0.

6. Regresa a la pestaña "Información sobre Factura" (id=ICTAB_0) y espera 3 segundos para que se cargue.
7. Hace clic en "Retención" (id=VCHR_HDR_WRK_XFR_WTHD_PB) para cargar la vista de retención.
8. Lee el texto "X de Y" en la página de retención para determinar el número total de páginas.
9. Para cada página de retención (de 1 a Y):
    - Llena dos inputs para porcentajes de retención (VCHR_LINE_WTHD_WTHD_RULE$0 y $1) con los valores correspondientes de la lista porcentaje_retencion.
    - Llena dos inputs para totales por artículo (VCHR_LINE_WTHD_WTHD_BASIS_AMT$0 y $1) con los valores correspondientes de la lista total_retencion_por_articulo.
    - Si no es la última página, hace clic en el botón "Siguiente" para avanzar a la siguiente página y espera 3 segundos.
10. Después de completar todas las páginas, hace clic en "Volver a Factura" (id=VCHR_PANELS_WRK_GOTO_VCHR_HDR) y espera 3 segundos para que la pantalla anterior se cargue completamente.

harmony_introd_comprobantes_descripcion_e_iva

Variables: descripciones, ivas, porcentaje_retencion, total_retencion_por_articulo

Esta función corresponde a los pasos 11 a 12 del flujo e ingresa descripciones e IVA para cada artículo de la factura. Navega por páginas de artículos, abre ventanas emergentes para ingresar valores de IVA, contrae secciones cuando es necesario y llena campos correspondientes con datos de listas proporcionadas.

1. Localiza el botón "Primero" utilizando su XPath. Luego realiza scroll hasta él y hace clic para regresar a la primera página. Espera 2 segundos para que la página se actualice.
2. Lee el texto "X de Y" para determinar cuántos artículos hay en total (Y). Calcula max_articulos como el mínimo entre Y, la longitud de descripciones y la longitud de ivas.
3. Iteración sobre cada artículo (desde 1 hasta max_articulos):
   Para cada artículo:
    1. Ingresa la descripción correspondiente en el campo VOUCHER_LINE_DESCR$0.
    2. Hace clic en "IVA Línea Factura" para abrir la ventana emergente de IVA.
    3. Espera para que la ventana emergente cargue.
    4. Sale al contenido principal y cambia al iframe emergente.

5. Hace clic en "Contraer Todas Secciones" para estandarizar el estado de la página.
6. Hace clic en la flecha "VAT_LABEL_WRK_VAT_DETAILS" para desplegar la sección de detalles de IVA.
7. Espera 2 segundos para que la sección se despliegue.
8. Ingresa el valor de IVA correspondiente en el campo VAT_FIELDS_WRK_TAX_CD_VAT.
9. Hace clic en "Ir a Línea Factura" para cerrar la ventana emergente y regresar al iframe principal.

11. Vuelve al contenido principal y cambia nuevamente al iframe principal ptifrmtgtframe.
12. Si no se ha procesado el último artículo, hace clic en el botón "Siguiente" para avanzar al siguiente artículo y espera 3 segundos.

Este flujo se repite para cada artículo hasta llenar todos los campos de descripción e IVA proporcionados en las listas. La función termina una vez procesados todos los artículos o alcanzado el número máximo calculado, en caso de que alguna de las listas para el ingreso de datos tenga menos elementos los inputs restantes se quedan vacíos.

harmony_introd_comprobantes_guardar

Variables: No requiere variables externas.

Esta función se ejecuta en el último paso de los tres casos, se encarga de guardar los comprobantes en la interfaz de Harmony. Realiza dos clics secuenciales en botones de guardar y espera tiempos específicos para asegurar que las acciones se procesen correctamente.

1. Localiza el primer botón "Guardar" utilizando el XPath proporcionado (//*[@id="win0divVCHR_PANELS_WRK_VCHR_SAVE_PB"]/a), realiza scroll hasta él y hace clic.
2. Espera 3 segundos para que el sistema procese el primer clic y se preparen los elementos necesarios para el siguiente paso.
3. Localiza el botón final "Guardar" utilizando el ID #ICSave y hace clic en él para confirmar y finalizar el guardado de los comprobantes.

Al finalizar estos pasos, la función registra que el proceso de guardado ha concluido correctamente.

Archivos Cami APP

cami_login.py

cami_dirigir_a_pagina_y_verificar_estado_login

Variables: CAMI_URL, CAMI_USER_EMAIL, CAMI_USER_PASSWORD

Esta función hace que el bot se dirija a la página de Cami (se encuentra en constantes.py) luego verifica en qué estado se encuentra (sin sesión, sin empresa, con empresa), esto existe porque, si el usuario da clic en "guardar contraseña" o si no se cerró correctamente el navegador, la sesión puede quedar abierta. Esto puede provocar que el bot intente iniciar sesión cuando ya está en el *home* o en la página de "elegir empresa". No es necesario hacer ninguna modificación en esta parte del código. Si no se puede iniciar sesión correctamente, verifique el archivo .env con las credenciales o las otras funciones.

Dependiendo del resultado se dirige a la función correspondiente:

cami_login

(Función auxiliar)
Variables: CAMI_USER_EMAIL, CAMI_USER_PASSWORD

Como su nombre lo indica inicia sesión en la página de Cami con las credenciales correspondientes del archivo .env. Ingresando los datos en los inputs correspondientes y presionando el botón.

cami_seleccionar_empresa

(Función auxiliar)
Variables: nombre empresa

Elige la empresa a la que va a ingresar el bot, en la pantalla que aparece justo después de iniciar sesión.

cami_navegar_a_modulo.py

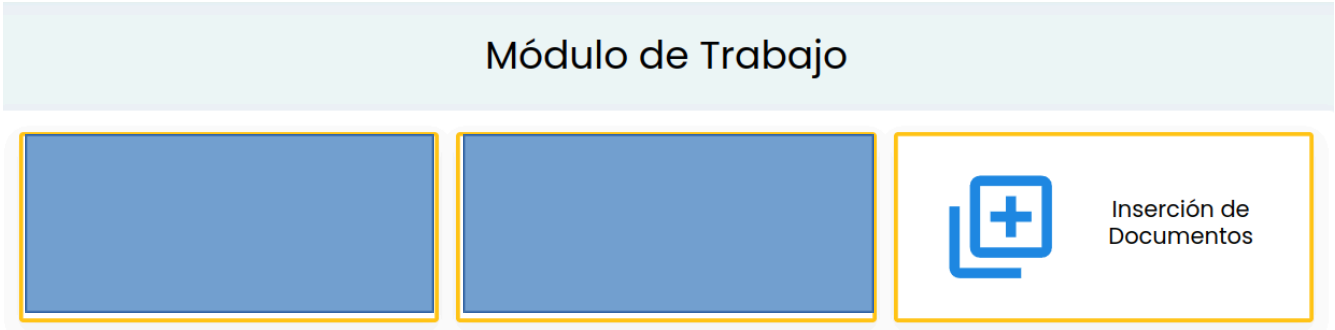
cami_navegar_a_modulo

Variables: nombre_seccion

Su única función es ir a la sección de “trabajo” de Cami (la que aparece en el navegador general) e ingresar al módulo que se desee, utilizando el texto visible en la página para que sea fácil de usar. Esta función es genérica y se emplea en cualquier caso. Por ejemplo:

cami_navegar_a_modulo(driver, "Inserción de Documentos").

Debe utilizarse exactamente el mismo texto que aparece en el módulo de Cami para que el bot lo encuentre correctamente.



Modificar Flujos

Para crear o modificar los flujos en el programa sigue los siguientes pasos:

1. Crea el archivo con el nombre de la página o programa a automatizar, seguido de la sección, por ejemplo:

sat_modulo_categoria_de_rentas.py

2. Define las funciones que quieras ejecutar dentro de este archivo, siguen la misma estructura: página, sección y luego una pequeña descripción, por ejemplo:

sat_categoria_de_rentas_busqueda_parametros()

Estas funciones deben empezar a funcionar desde que el bot se encuentra dentro del módulo, ya que existe la función navegar _a_ módulo para cada página.

Se pueden agregar otras funciones que se utilicen en repetidas ocasiones al final del código pero que son necesarias para el funcionamiento de alguna de las funciones principales (las que se acaban de definir), se puede indicar como “Funciones auxiliares”, estas funciones no van en la configuración del flujo.

3. En el archivo configuration bot_y_flujos.py, importar las funciones correspondientes y luego indicar en el comentario a qué pasos pertenece. Por último, revisar que la función general del flujo esté siendo llamada desde el main.py.

Notas:

Es importante siempre pasar como atributo de la función el driver, este incluye la configuración del bot de selenium.

Al agregar el flujo nuevo, utilizar al inicio la función navegar _a_ módulo correspondiente a la página web.

Al hacerlo de esta forma cada flujo es independiente y el código se puede expandir sin ningún problema, siempre que en las nuevas funciones no interactúen con otras de otros archivos. Se recomienda ver el código de los flujos ya existentes para seguir la misma lógica e incluir los logs con la misma estructura.

Uso de Selenium

Selenium proporciona varios métodos para interactuar con elementos en una página web, siendo la eficiencia y claridad clave para elegir el método adecuado. El más eficiente es el uso de ID find_element(By.ID), ya que los navegadores están optimizados para localizar elementos con atributos `id`. Este método es ideal cuando el `id` del elemento es único y fijo. Si el atributo `id` no está disponible, el segundo método recomendado es buscar por nombre find_element(By.NAME), que utiliza el atributo `name` del elemento HTML y también ofrece un rendimiento rápido.

Cuando los identificadores únicos no están presentes al inspeccionar el elemento, se puede recurrir a los selectores CSS find_element(By.CSS_SELECTOR), que permiten buscar elementos por atributos, clases, o relaciones jerárquicas. Este método es flexible y eficiente, ideal para estructuras HTML complejas. Alternativamente, XPath find_element(By.XPATH) ofrece un alto grado de flexibilidad para buscar elementos utilizando condiciones complejas o rutas relativas, aunque generalmente es menos eficiente que los selectores CSS aunque

mucho más fácil de implementar, ya que solo se necesita copiar el XPATH del componente. Si se trata de buscar elementos con clases específicas o genéricas, se puede usar el método class find_element(By.CLASS_NAME), aunque debe hacerse con precaución, ya que las clases suelen repetirse en el HTML y Selenium solo utilizará el primer elemento encontrado que pertenece a esa clase.

Otras formas de interacción incluyen buscar elementos por etiquetas find_element(By.TAG_NAME), que puede ser útil para localizar grupos de elementos como tablas o listas, y por texto visible de enlaces find_element(By.LINK_TEXT o find_element(By.PARTIAL_LINK_TEXT), ideal para encontrar enlaces en el DOM.

Además de localizar elementos, Selenium permite realizar diversas acciones sobre ellos. Puedes usar el método click() para hacer clic en un elemento, como un botón o enlace. Si necesitas ingresar texto en un campo, send_keys() es el método adecuado. Antes se puede usar clear() para vaciar el contenido previo del campo. En casos donde el elemento no está visible en la pantalla, el método execute_script() con scrollIntoView() permite desplazar la página hasta que el elemento esté en vista, este último se recomienda utilizar siempre ya que en algunas ocasiones las pantallas son de diferentes tamaños y aunque originalmente el código funciona bien, al ejecutarlo en una pantalla más pequeña puede que no pueda acceder al elemento, por esta razón, se recomienda hacer scroll hasta que el elemento seleccionado esté visible.

Cuando los elementos tardan en cargarse o dependen de eventos dinámicos, se utiliza WebDriverWait. Este método permite esperar explícitamente hasta que un elemento sea visible, clickeable, o cumpla con una condición específica, asegurando que las interacciones sean estables. Por ejemplo:

```
from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

```
element = WebDriverWait(driver,
10).until(EC.visibility_of_element_located((By.ID, "element-id")))
```

Otra recomendación es utilizar time.sleep(), esta función nativa de Python sirve para detener la ejecución del código por cierto tiempo, ya que normalmente Selenium realiza las acciones muy rápido, en caso de que el elemento no sea accesible o no se esté haciendo click intente agregar un time.sleep(), en especial en casos en los que se utiliza scrollIntoView, para que le de tiempo al navegador de ubicar el elemento en pantalla antes de hacer click, ingresar texto, etc.

Finalmente, Selenium también permite ejecutar código JavaScript directamente con execute_script(). Esto es útil para realizar acciones que no están disponibles nativamente en Selenium o para manipular el DOM directamente. Por ejemplo, puedes simular clics complejos, ajustar valores de estilo o ejecutar funciones específicas de la página.