🚀 Ejecución de Automatizaciones desde Línea de Comandos y API

Este documento proporciona una guía rápida para ejecutar procesos y bots en las plataformas de automatización más populares: **UiPath**, **Automation Anywhere** y **Power Automate** desde **Orquestador de Rocketbot.**

Debes crear un paquete para ejecutar desde el orquestador como se indica aqui Subir archivos con extensión .zip al robot puedes utilizar la siguiente guia para agregar comandos que llamen a estas herramientas de RPA

🤖 UiPath

🖥️ Ejecución desde Línea de Comandos

UiPath proporciona una herramienta llamada UiRobot.exe para ejecutar procesos sin necesidad de Orchestrator.

Comandos básicos:

```
UiRobot.exe execute --process <NombreDelProceso>
UiRobot.exe execute --file "C:\Ruta\Al\Archivo.xaml"
UiRobot.exe execute --process <NombreDelProceso> --input
"{'param1':'valor1','param2':2}"
UiRobot.exe execute --process <NombreDelProceso> --folder CarpetaOrchestrator
```

- --file: Ejecuta directamente un .xaml local.
- --input: Parámetros en formato JSON.
- --folder: Carpeta desde Orchestrator.

📘 Documentación CLI UiPath

🌐 Ejecución mediante API (UiPath Orchestrator)

UiPath Orchestrator permite controlar procesos de forma remota mediante API REST.

1. Autenticación:

```
POST /api/account/authenticate
{
  "tenancyName": "default",
  "usernameOrEmailAddress": "user",
```

```
  "password": "pass"
}
```

Respuesta: {"result": "token"}

2. Obtener Release Key:

```
GET /odata/Releases
Authorization: Bearer <token>
```

3. Ejecutar el proceso:

```
POST /odata/Jobs/UiPath.Server.Configuration.OData.StartJobs
Authorization: Bearer <token>
{
  "startInfo": {
    "ReleaseKey": "<releaseKey>",
    "Strategy": "Specific",
    "RobotIds": [1234],
    "InputArguments": "{"param":"value"}"
  }
}
```

 Documentación API UiPath

Automation Anywhere

Ejecución desde Línea de Comandos

Utiliza la herramienta aascli:

```
aascli run --bot "Bots/MyBotFolder/MyBot" --params '{"param1":"valor1"}'
```

- El bot debe estar publicado en Control Room.
- Necesitas estar autenticado con Bot Agent activo.

 CLI de Automation Anywhere

Ejecución mediante API

1. Autenticación:

```
POST /v1/authentication
{
  "username": "user",
  "password": "pass"
```

```
}
```

Respuesta: {"token": "JWT_TOKEN"}

2. Obtener IDs necesarios:

```
GET /v3/automations
GET /v3/users
```

3. Ejecutar el bot:

```
POST /v3/automations/deploy
Authorization: Bearer <token>
{
  "fileId": "<bot_id>",
  "runAsUserIds": ["<user_id>"],
  "botInput": {
    "param1": "value1"
  }
}
```

 API Automation Anywhere

Power Automate

Flujos de Escritorio desde Línea de Comandos

Usa PAD.Console.Host.exe:

```
"C:\Program Files (x86)\Power Automate Desktop\PAD.Console.Host.exe" "ms-powerautomate:/console/flow/run?workflowName=NombreDelFlujo"
```

También puedes ejecutar por ID:

```
PAD.Console.Host.exe "ms-powerautomate:/console/flow/run?workflowId=<id>&inputArguments={\"param1\":\"value1\"}&autologin=true"
```

 Documentación CLI PAD

Ejecución mediante API

1. Autenticación con Azure AD:

Utiliza Microsoft Identity Platform para obtener un token con permisos para acceder a Dataverse.

2. Obtener Workflow ID:

```
GET https://<org>.crm.dynamics.com/api/data/v9.2/workflows?$filter=category eq 6
```

3. Ejecutar flujo:

```
POST https://<org>.crm.dynamics.com/api/data/v9.2/workflows(<workflowid>)/Microsoft.Dynamics.CRM.RunDesktopFlow
Authorization: Bearer <token>
{
  "runMode": "attended",
  "runPriority": "normal",
  "connectionName": "Default Windows",
  "timeout": 7200,
  "inputs": "{\"param1\":\"value1\"}"
}
```

 API Power Automate

Recursos Recomendados

- UiPath sin Orchestrator (YouTube)
- Automation Anywhere desde API (YouTube)
- Power Automate ejecución directa (YouTube)

Crear contraseña de aplicación para Gmail

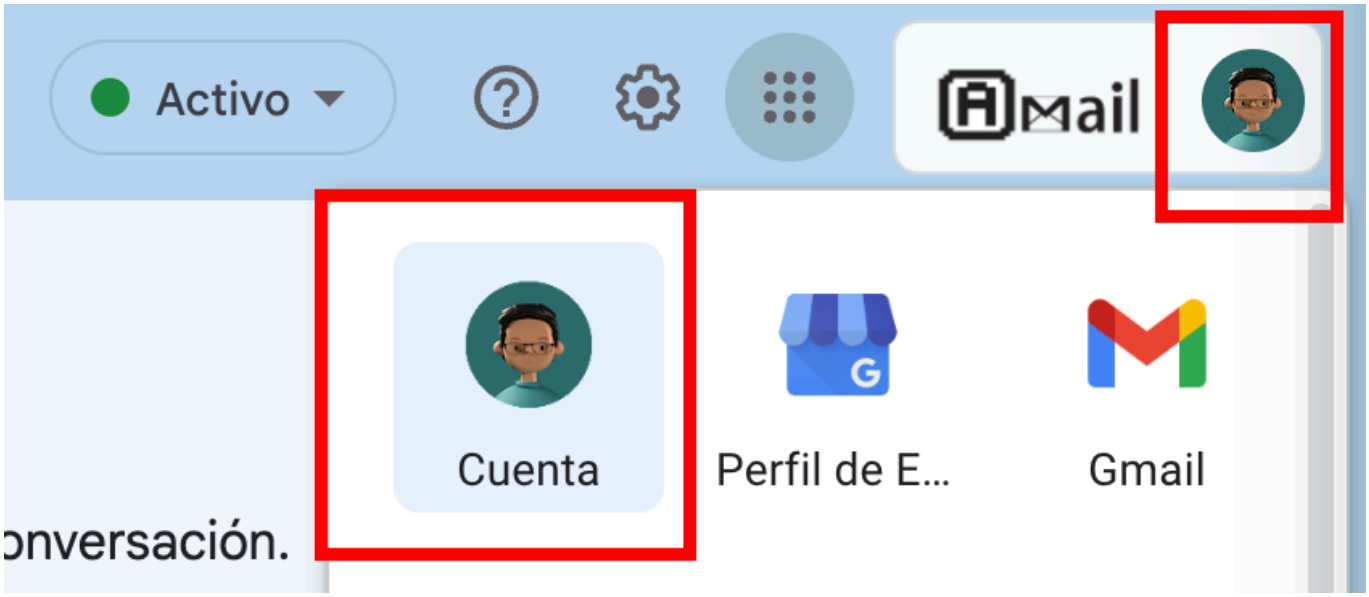
Una contraseña para una aplicación es un password de 16 letras que Google almacena en su cuenta y que da acceso al sistema sin necesidad de que se produzca una verificación en dos pasos.

Para su configuración es necesario cumplir tener activo la verificación de segundo factor 2FA. En el caso de no tenerlo activado, no podrá crear una contraseña de aplicación. Para ello debemos de cumplir los siguientes pasos:

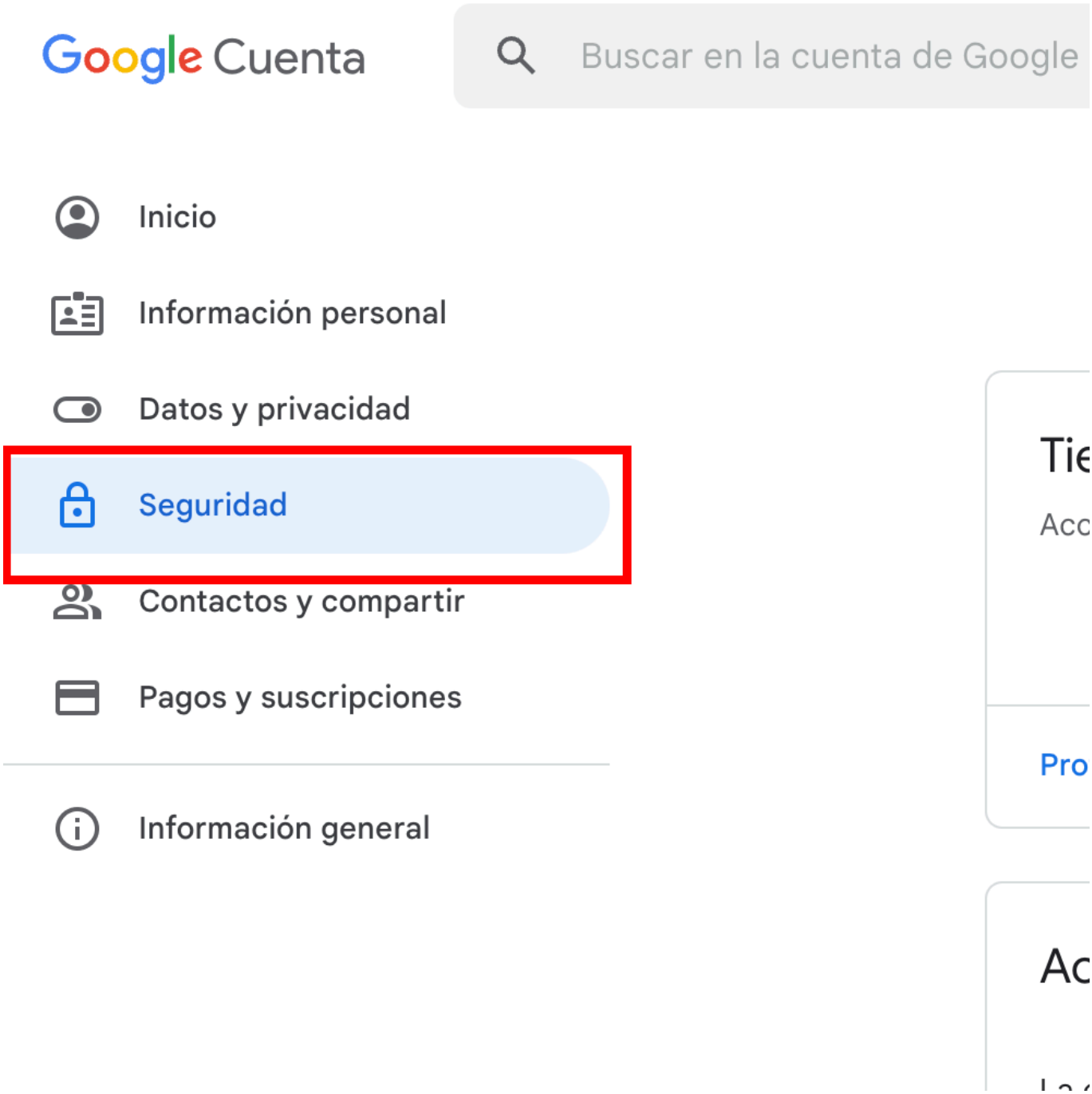
Primer Paso .- Activar verificación de segundo

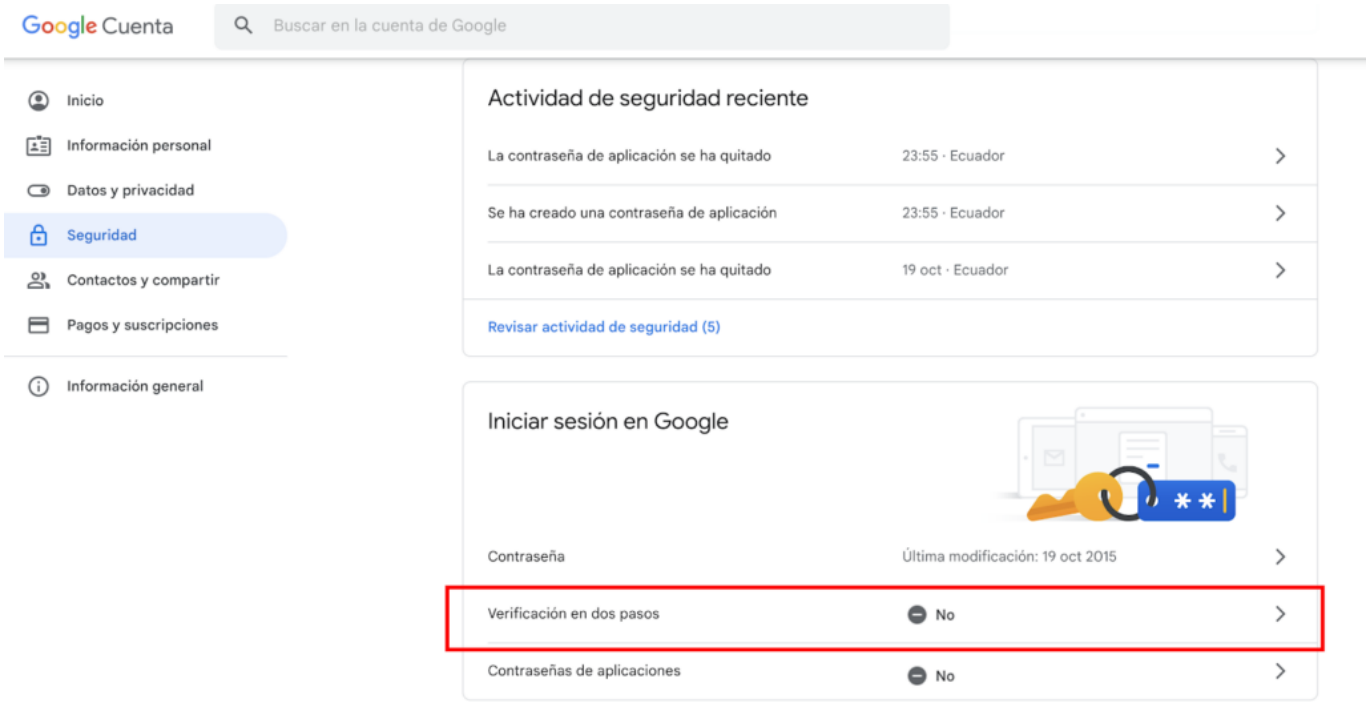
factor.

Lo primero que debes hacer es iniciar sesión en tu cuenta de Gmail, una vez conectado a tu cuenta, accede al menú de configuraciones. Lo encuentras en la parte superior derecha del navegador, en el círculo con tu avatar, luego debes elegir la opción «Cuenta de Google«

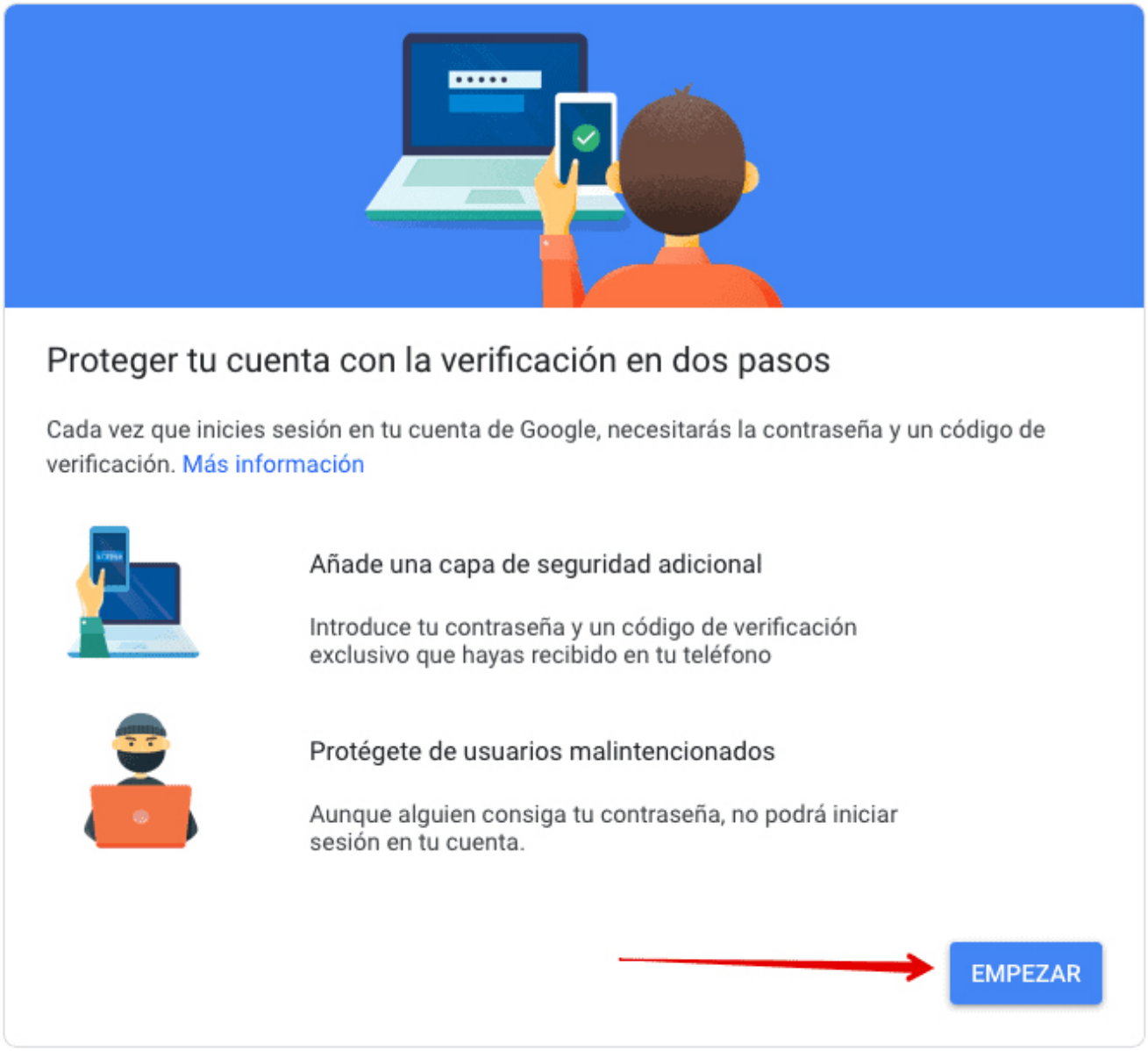


Te llevará a una pantalla de configuración de tu cuenta, donde tendrás que acceder al menú lateral»Seguridad»:


Google Cuenta
Buscar en la cuenta de Google
Inicio
Información personal
Datos y privacidad
Seguridad
Contactos y compartir
Pagos y suscripciones
Información general
Tie
Acc
Pro
Ac

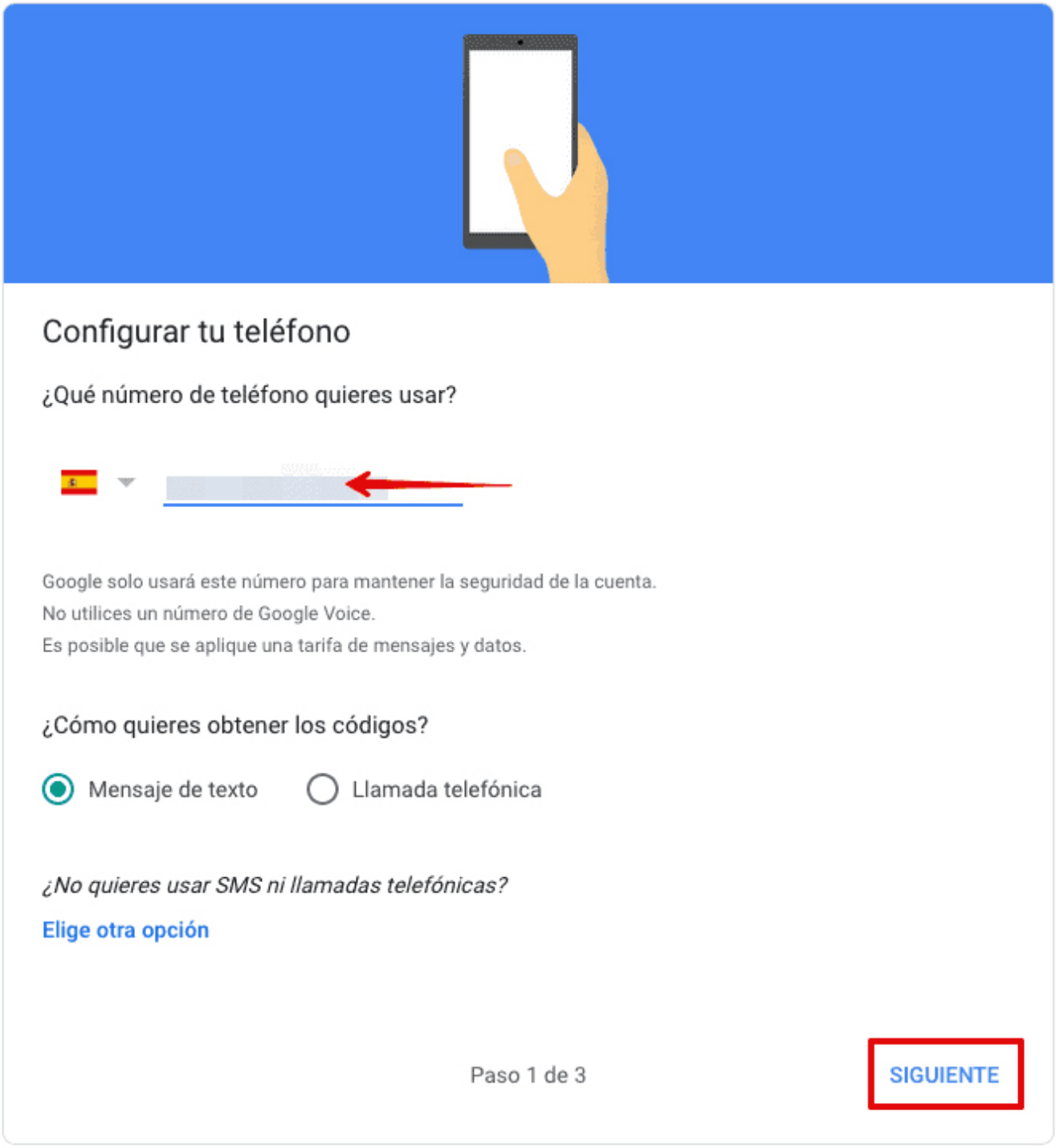


Verás una breve explicación sobre el doble factor de autenticación. Para iniciar este proceso haz clic en el botón «Empezar».

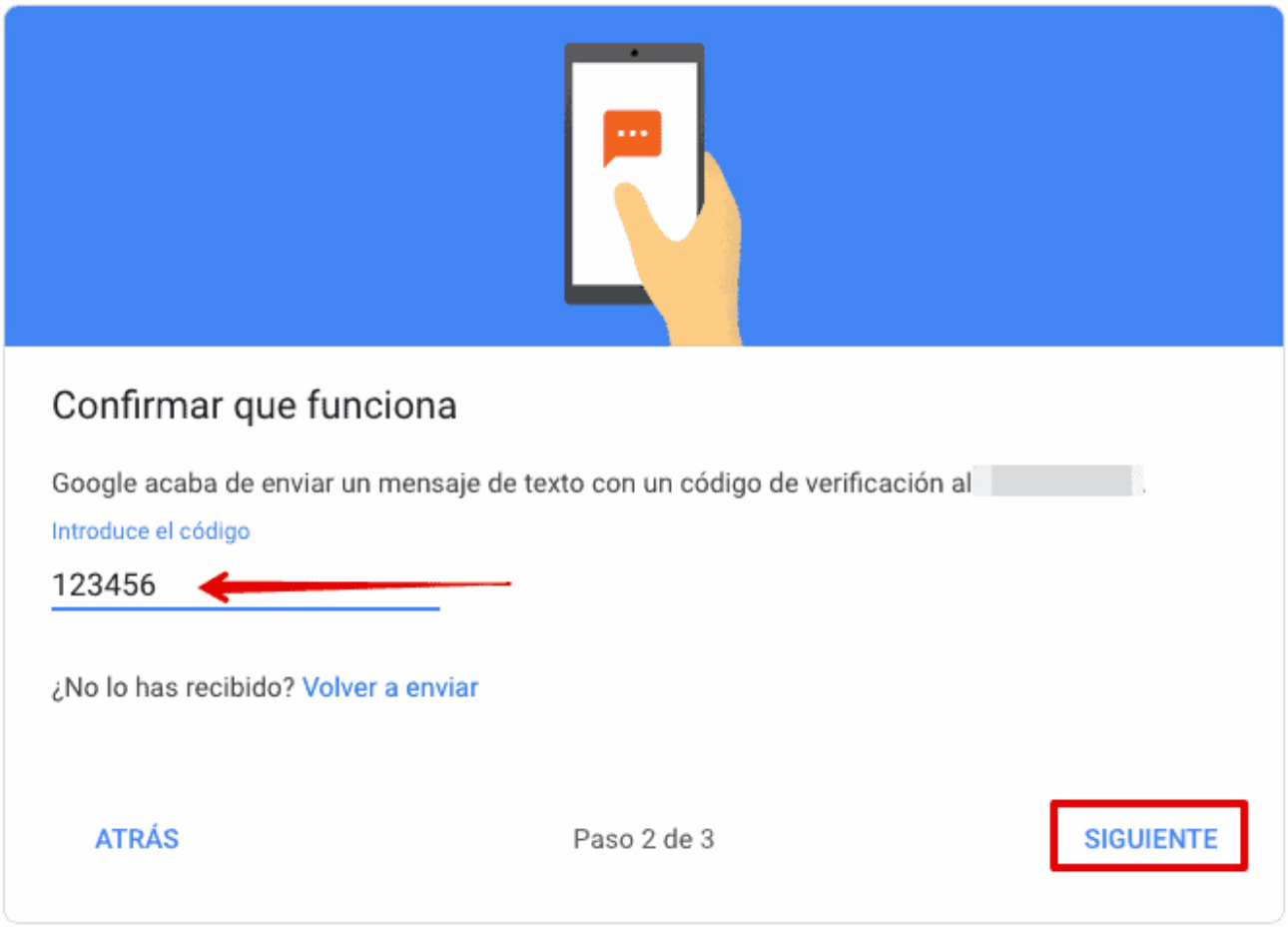


Vuelve a indicar la contraseña de tu cuenta en Google y haz clic en «Siguiente«:

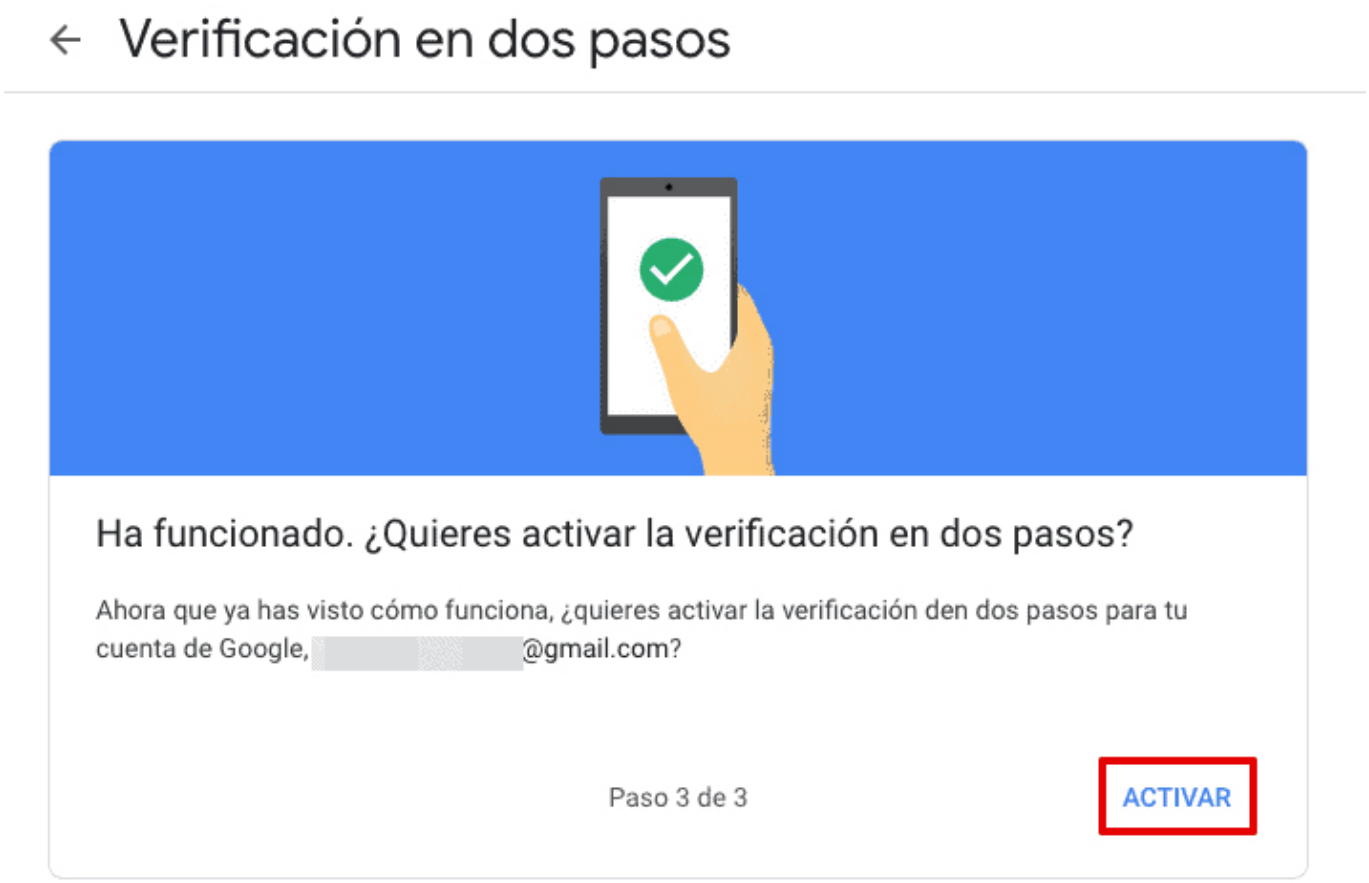
Introduce tu número de teléfono y la forma que deseas recibir los códigos, a continuación pulsa en «Siguiente«:



Google te enviará los códigos de confirmación. Introduce tu código, a continuación pulsa en «Siguiente«:



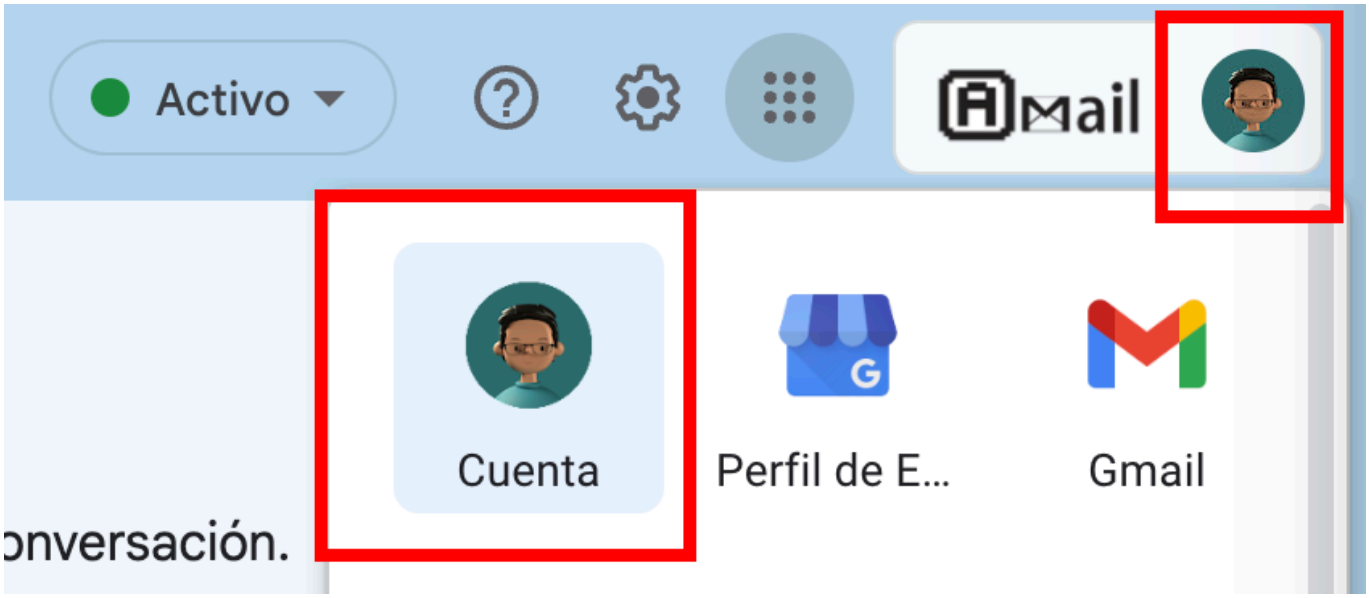
Una vez confirmado tu número de teléfono, puedes empezar a configurar el 2FA, para ello haz clic en «Activar«:



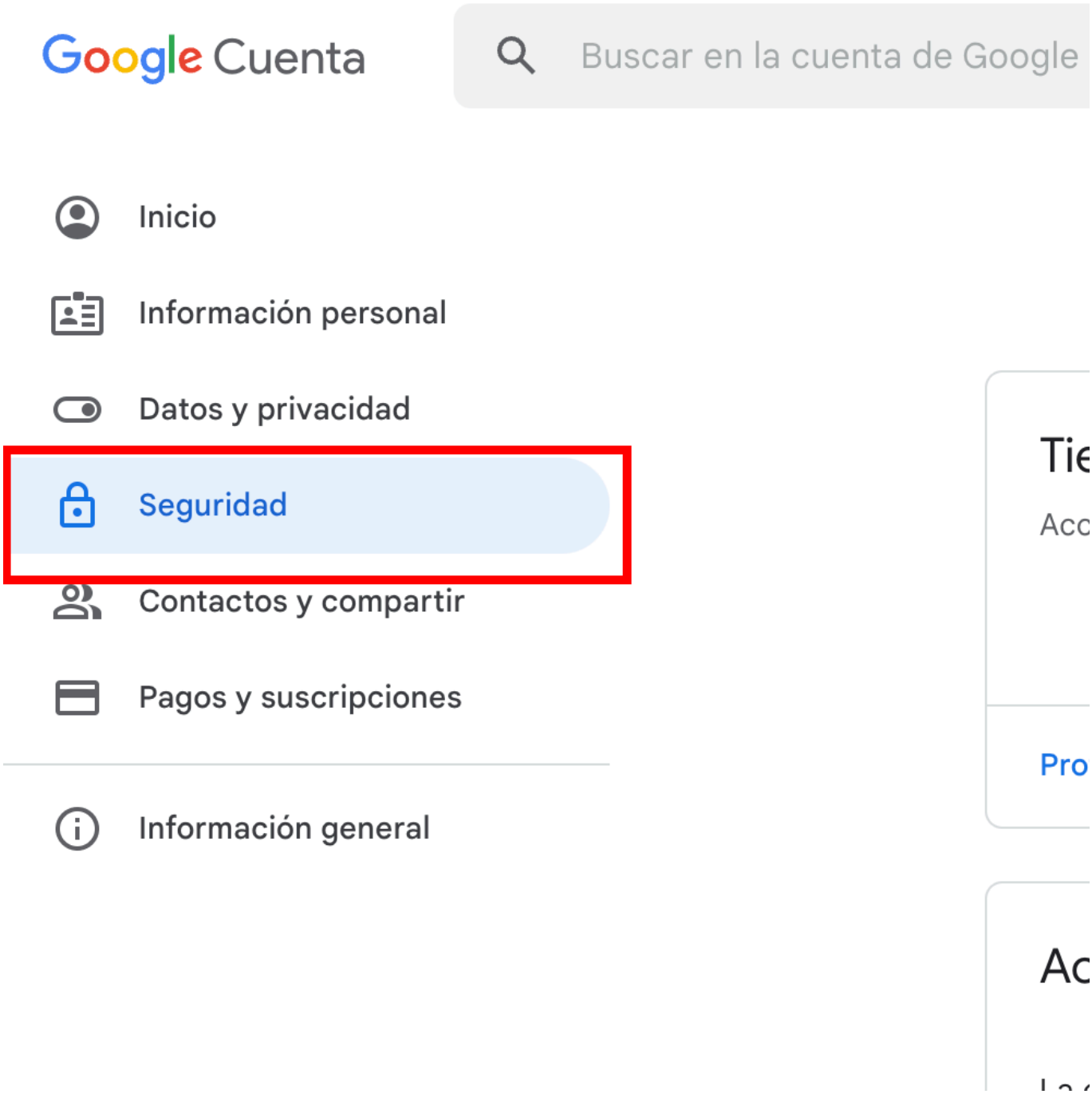
Listo, su cuenta de gmail.com es segura y cuenta con una segunda capa de seguridad para acceder a su correo.

Segundo Paso .- Configurar una contraseña de Aplicación

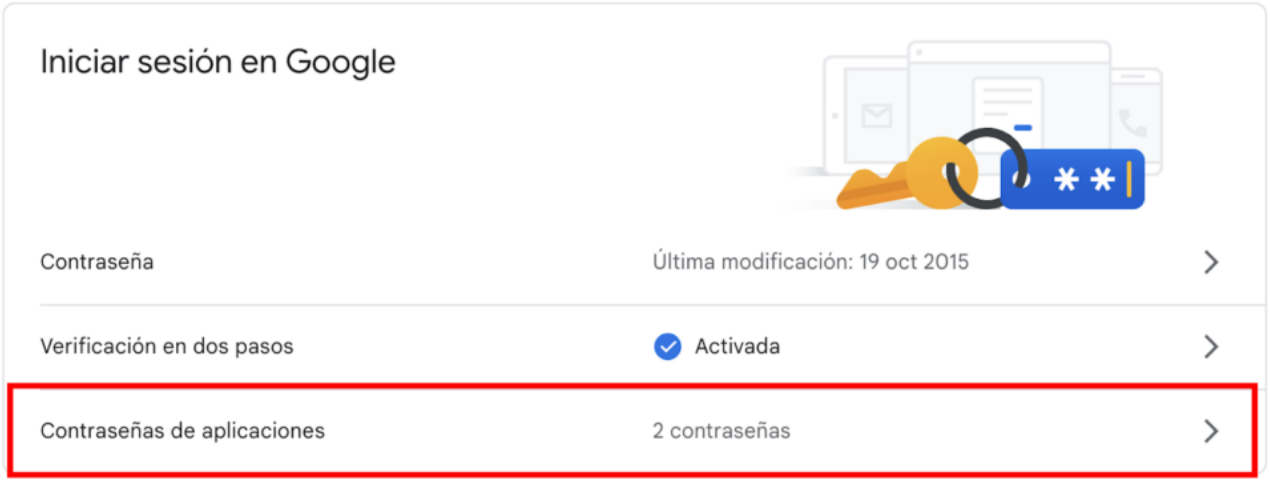
La contraseña de aplicación se encuentra en la misma sección de Gestionar tu cuenta.



En la parte media de la página, en la sección iniciar sesión en google, encontrarás una parte donde pone contraseñas de aplicación: ninguna. Haz clic en ninguna o en la flecha hacia la derecha

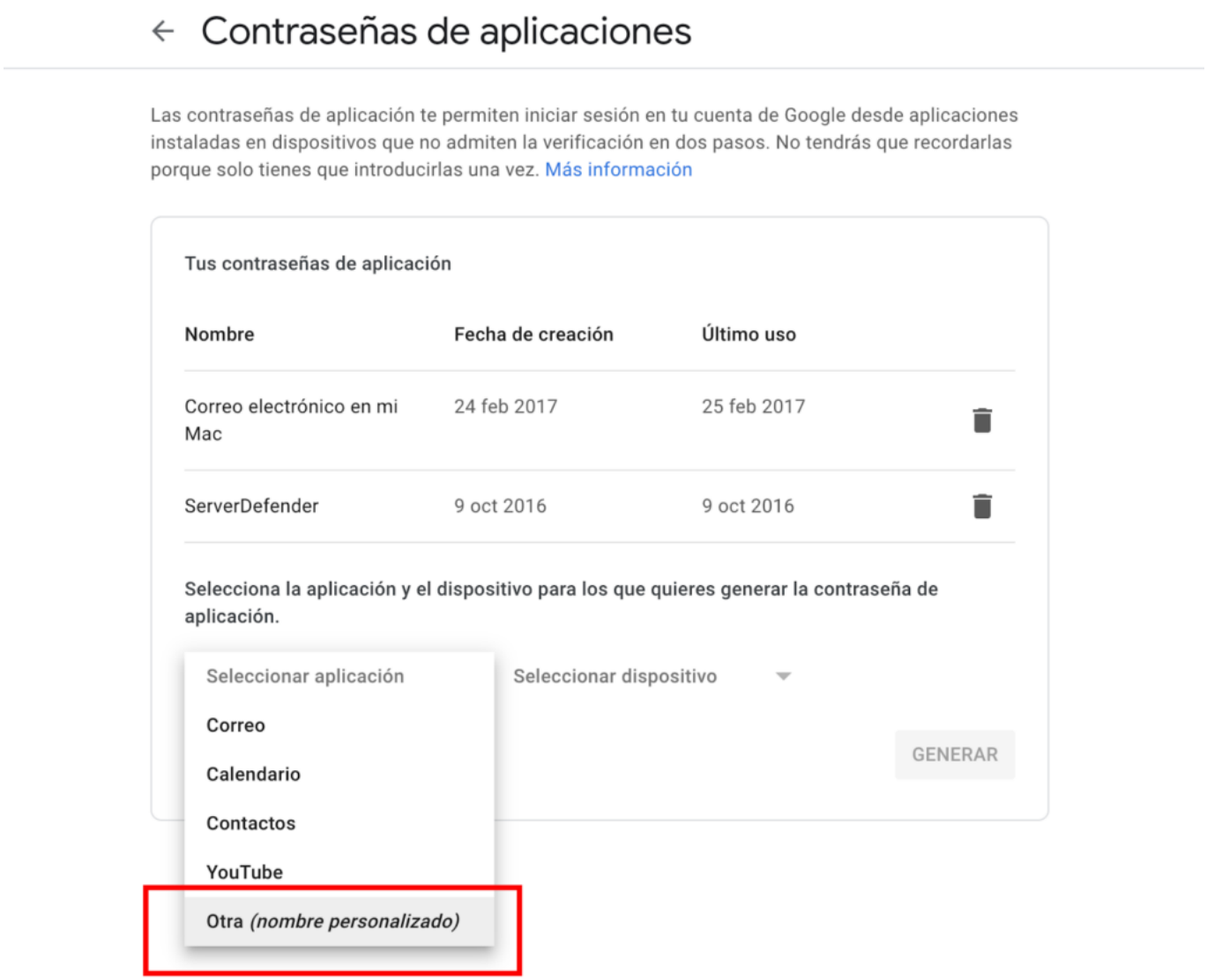


En esta imágen de referencia notamos que ya existen 2 contraseñas de aplicación creadas, pero en el caso de no tener ninguna contraseña creada se mostrará la palabra NInguna.

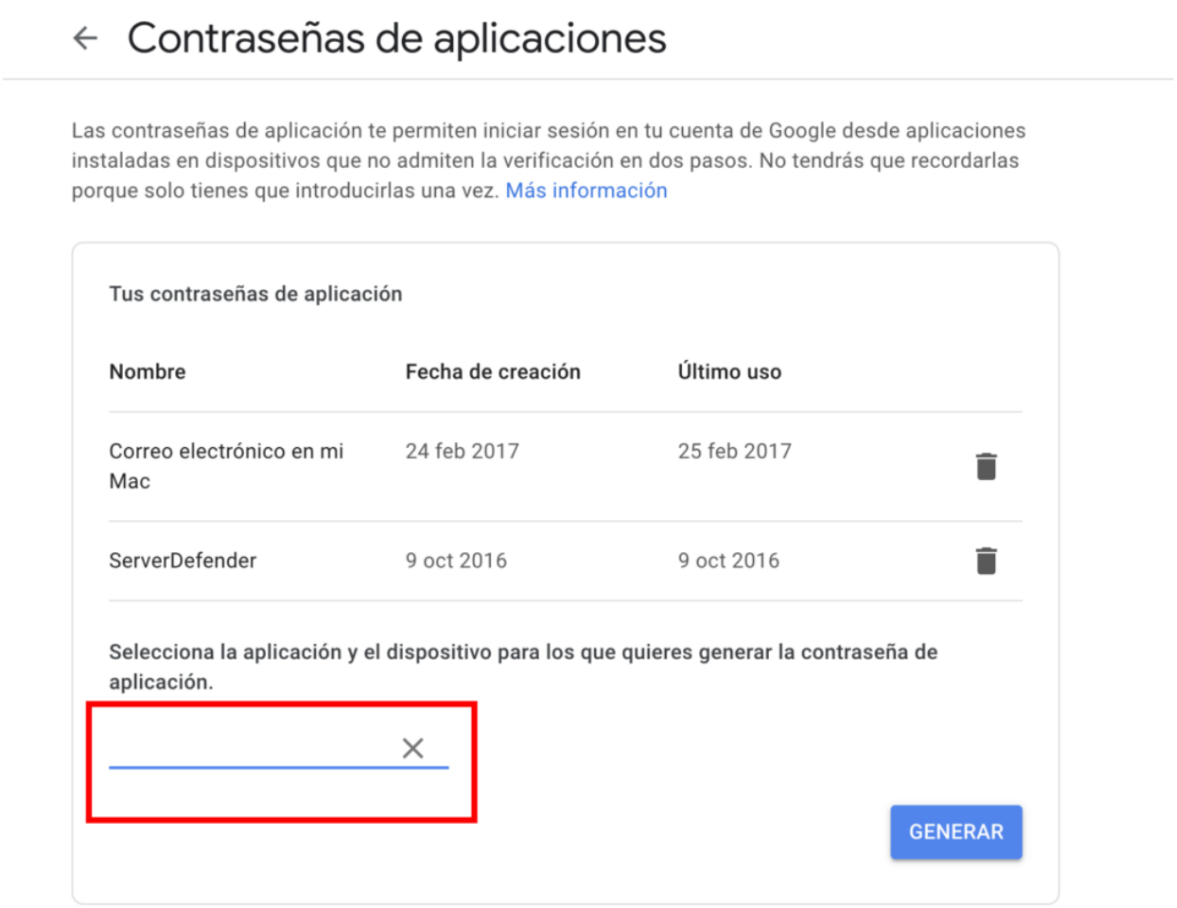


En la siguiente pantalla pedirá que vuelvas a poner tu contraseña, por seguridad. Debes poner tu contraseña actual. No se va a cambiar, vamos a crear una especial. Una vez pongas tu contraseña, la siguiente pantalla es la que permite crear la nueva contraseña de aplicación

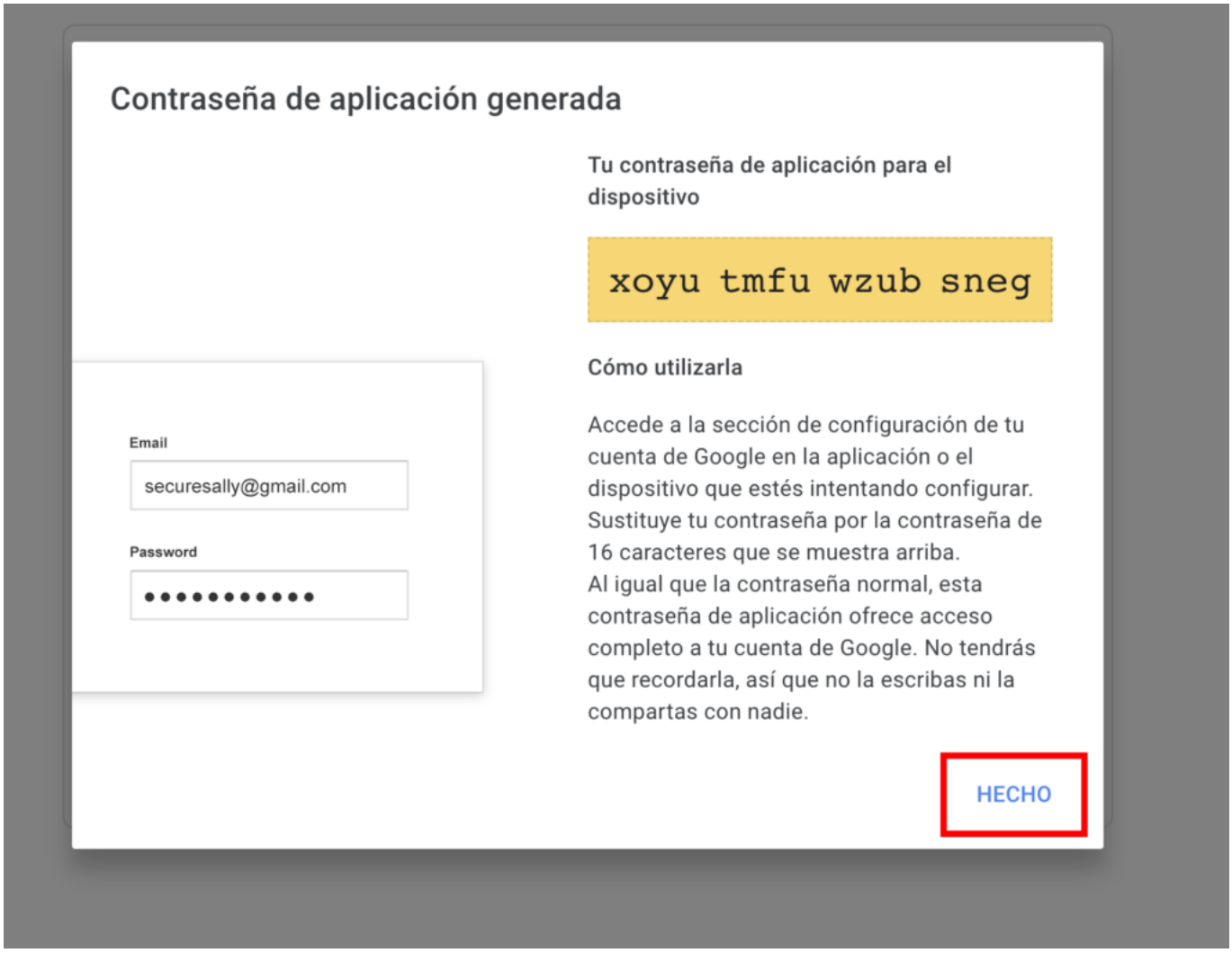
Puedes seleccionar en Otros (nombre personalizado) con la finalidad de asignarle un nombre.



Al dar clic , debemos colocar el nombre y pulsa el botón generar



Ahora por fin has creado una nueva contraseña de aplicación exclusiva para enviar emails desde cualquier App . Solamente, no sustituye a tu contraseña de gmail.



Datos para Configuración

Luego de crear la contraseña de aplicación, podrá configurar su cuenta en el portal. Los datos de envio son los siguiente:

1. Correo/Usuario: sucorreo@gmail.com (en el caso de google apps sería @sudominio.com)
2. Contraseña: La generada en el segundo paso.
3. Servidor SMTP : smtp.gmail.com
4. Puerto: 587
5. TLS : SI
6. Autenticación SMTP : SI

Aqui estan los links de los modulos donde se pueden ejecutar estas acciones :

Email Advaced : https://market.rocketbot.com/module/emailAdvanced

Gmail : https://market.rocketbot.com/module/gmail_

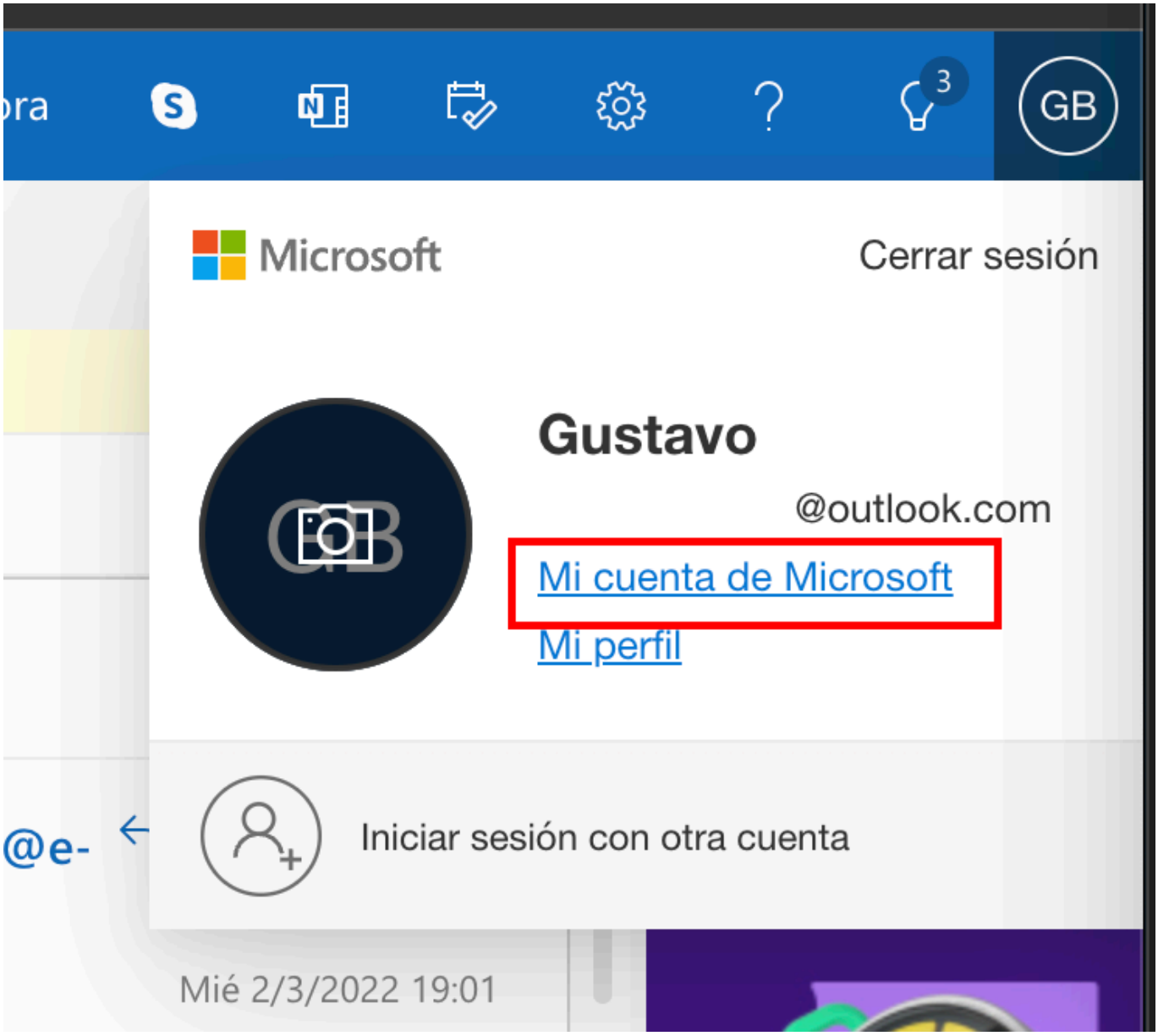
Crear contraseña de aplicación para Outlook

Una contraseña para una aplicación es un password que Outlook almacena en su cuenta y que da acceso al sistema sin necesidad de que se produzca una verificación en dos pasos.

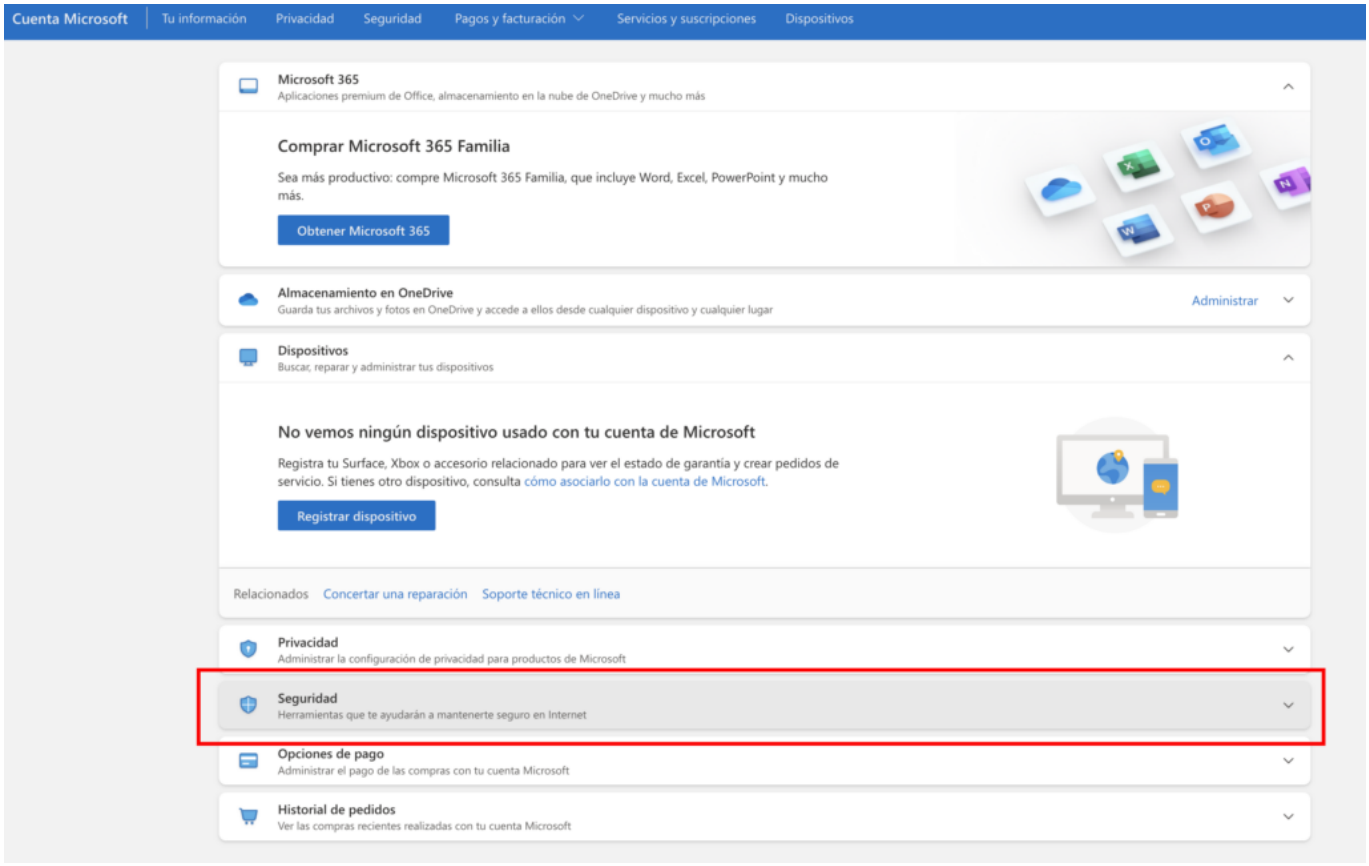
Para su configuración es necesario cumplir tener activo la verificación de segundo factor 2FA. En el caso de no tenerlo activado, no podrá crear una contraseña de aplicación. Para ello debemos de cumplir los siguientes pasos:

Primer Paso .- Activar verificación de segundo factor.

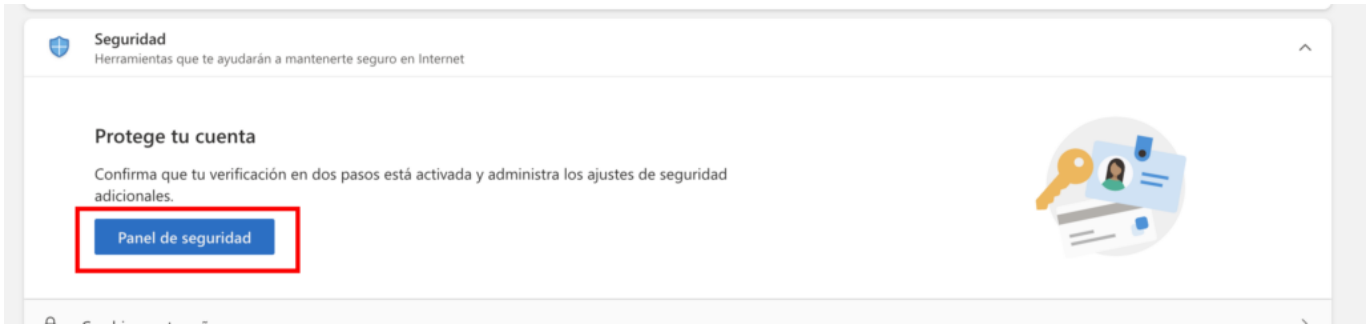
Lo primero que debes hacer es iniciar sesión en tu cuenta de Outlook, una vez conectado a tu cuenta, accede al menú de tu cuenta. Lo encuentras en la parte superior derecha del navegador, en el cícurlo con tu avatar, luego debes elegir la opción «Mi cuenta de Microsoft«



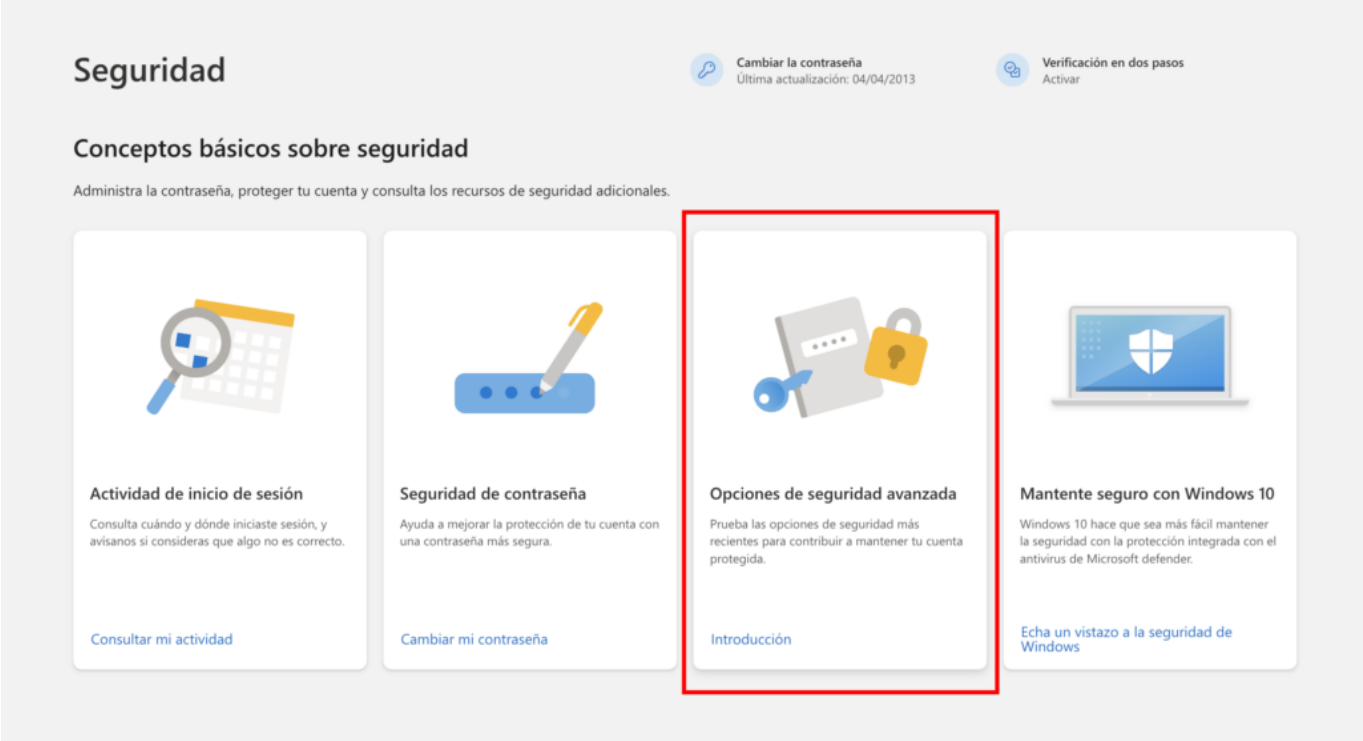
Te llevará a una pantalla, donde tendrás que acceder a la opción de »Seguridad»



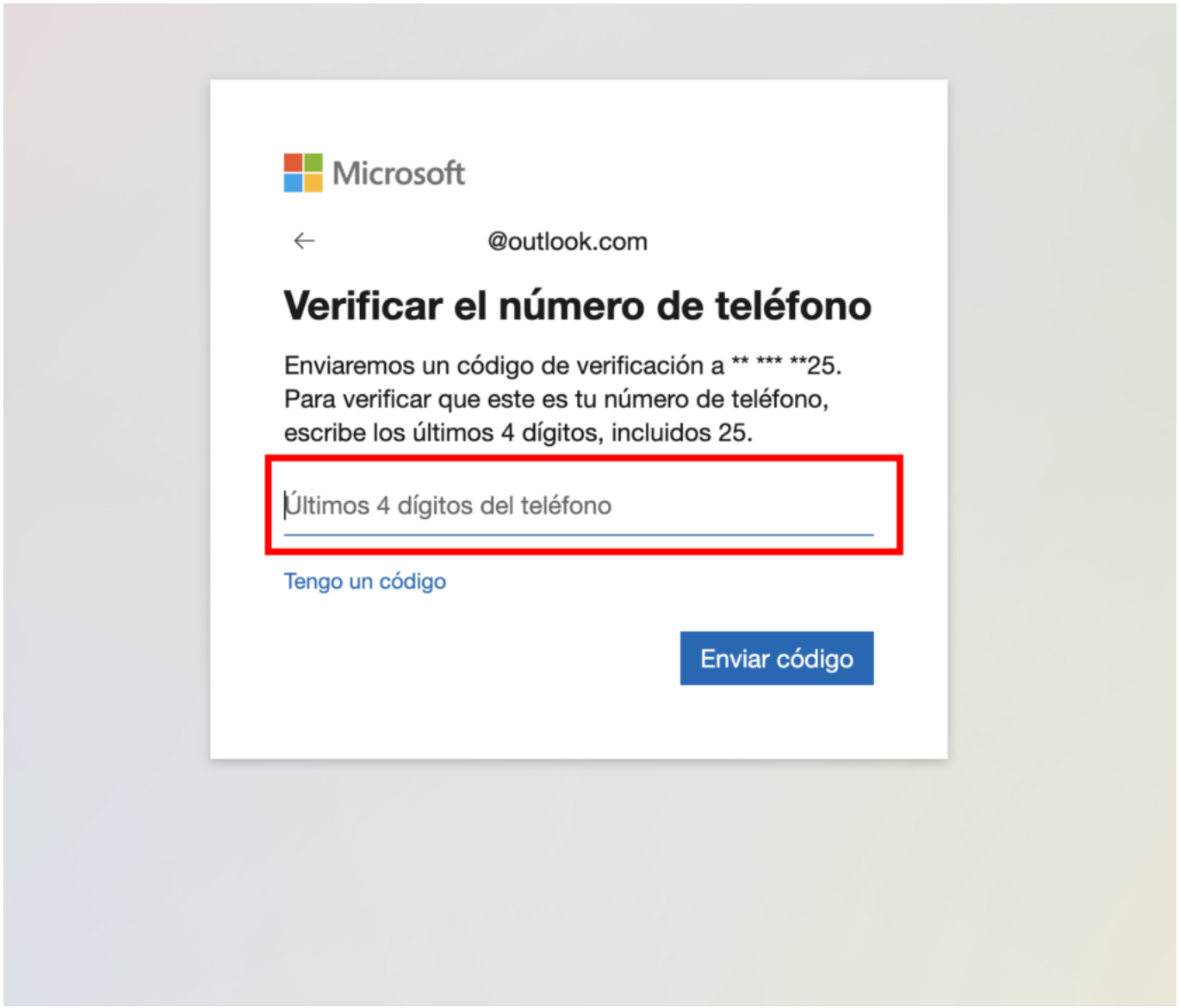
Al presionar te mostrará una opción para acceder presionando en el botón de »Panel de seguridad»:



Serás redireccionado al panel de Seguridad, donde podrás ver las opciones de conceptos básicos. Para iniciar este proceso haz clic en la opción **«Opciones de seguridad avanzada».**



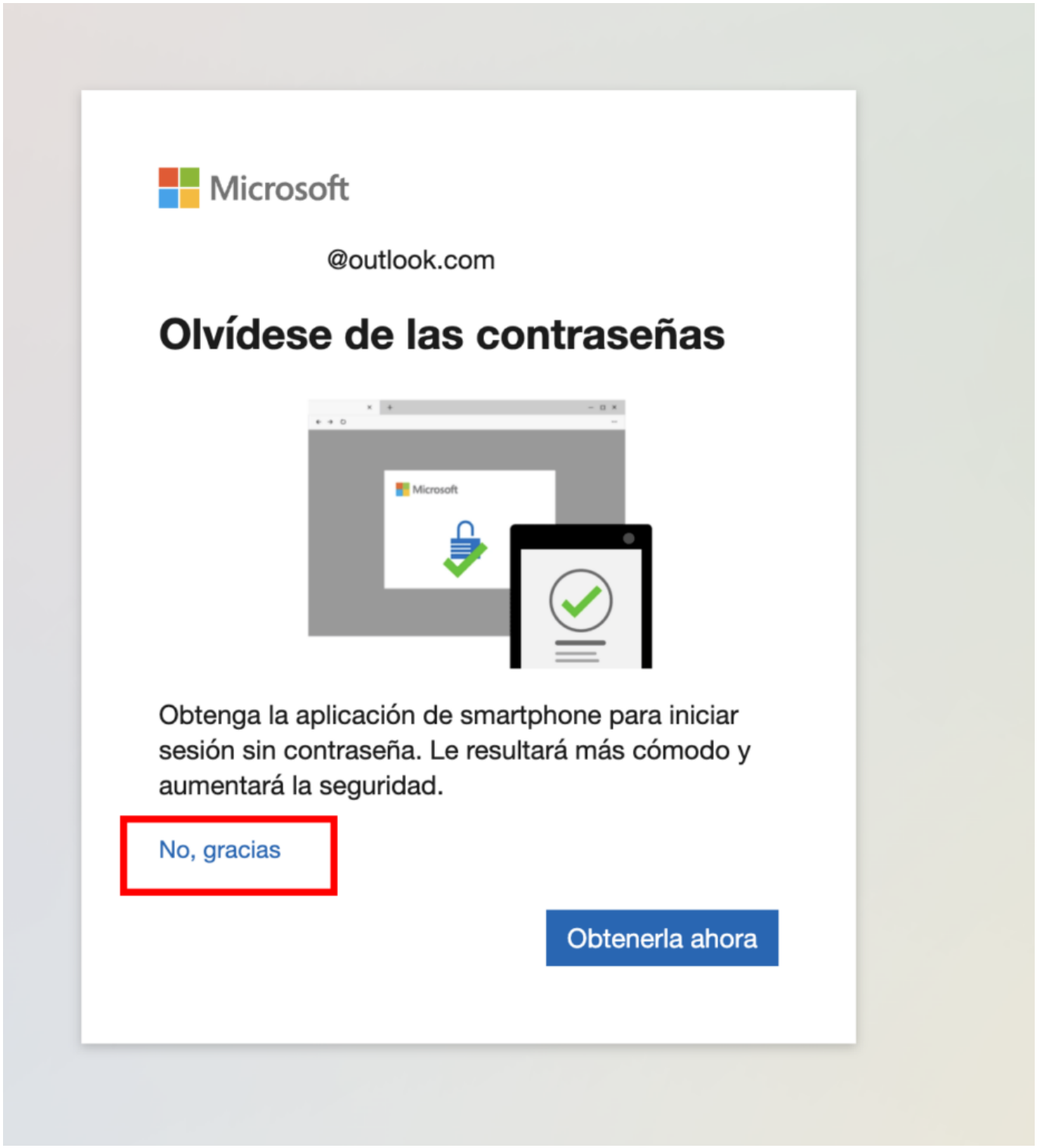
Luego de eso se le pedirá que confirme el número de telefono asociado a su cuenta, por lo cual debe ingresar los 4 últimos digitos para continuar.



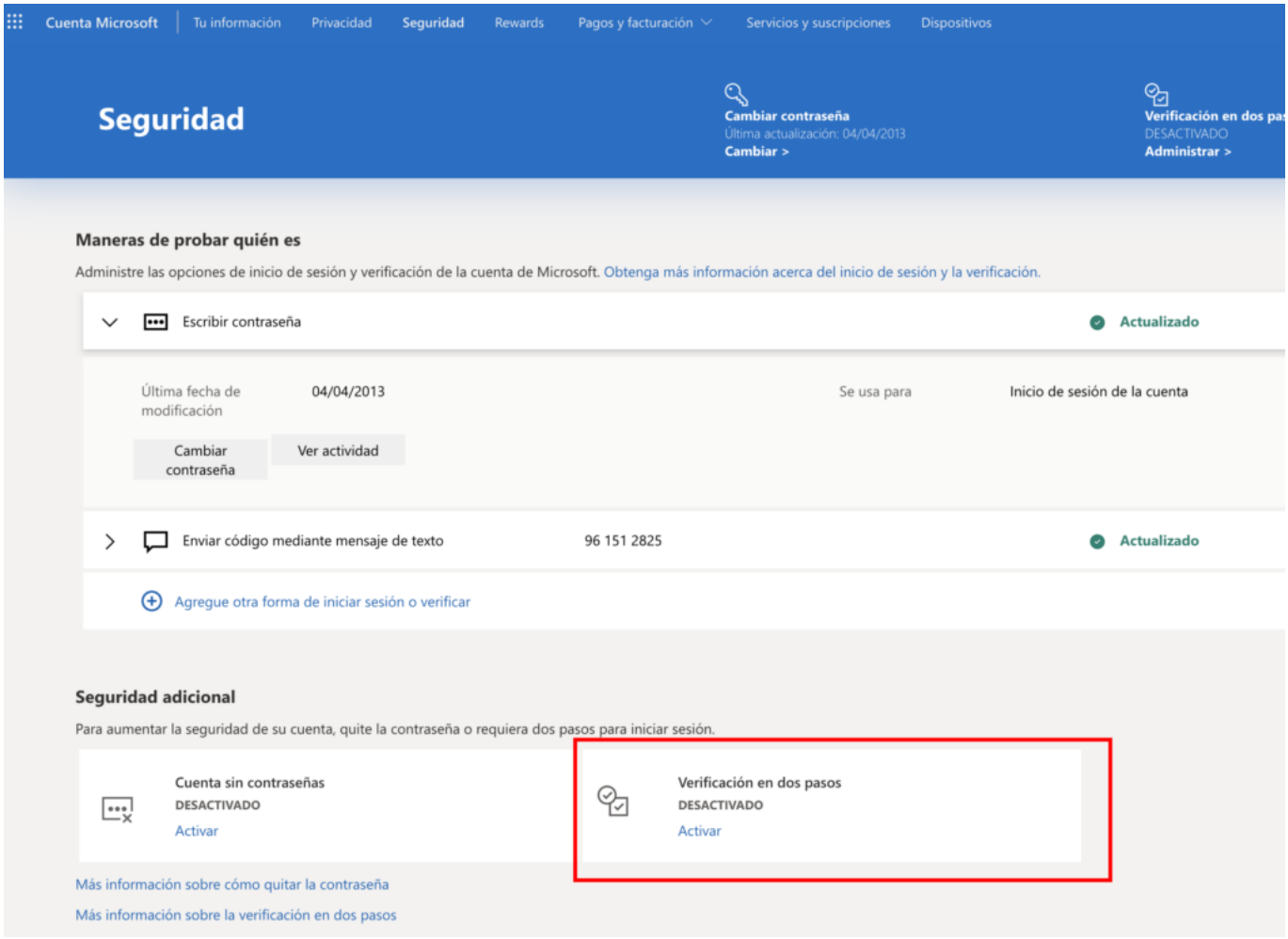
Con la finalidad de poder explicar el método más siemple para la verificación de segundo factor, escogeremos la opción de «**No, gracias**» del panel de opciones.

Importante:

Cómo alternativa Outlook le ofrecerá utilizar su aplicación móvil para generar las contraseñas. Puede optar por ese procedimiento en el caso de que desee.



Al finalizar, se nos mostrará la siguiente pantalla de Seguridad, en donde podrá notar que la **Verificación en dos pasos** se encuentra **DESACTIVADA** , por lo que procedemos a dar clic en **«Activar»**

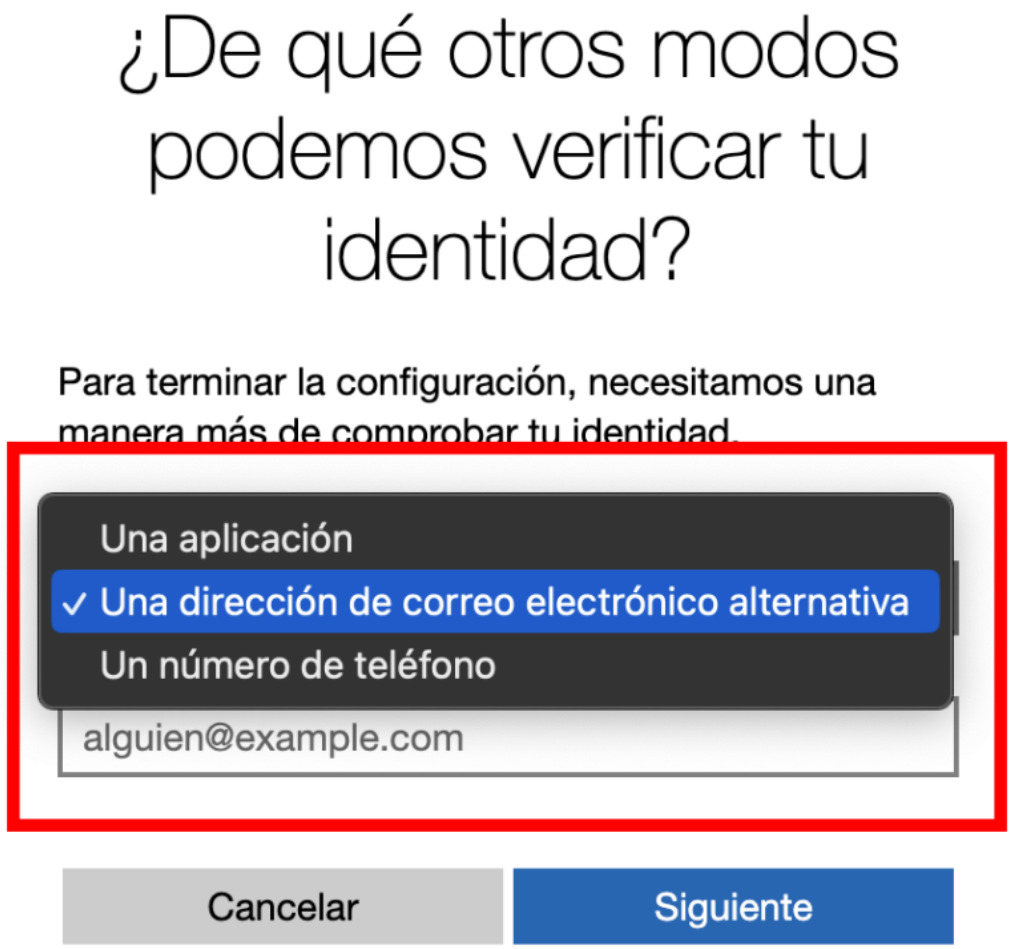


Al dar clic en la opción de Activar, se nos mostrará un nuevo cuadro de dialogo con diferentes opciones para verificar el acceso a la cuenta.

Usted puede escoger cualquiera de las 3 opciones que muestran a continuación. A continuación escogeremos la forma más sencilla de verificación de correo el cual es **Una dirección de correo electrónico alternativa.**

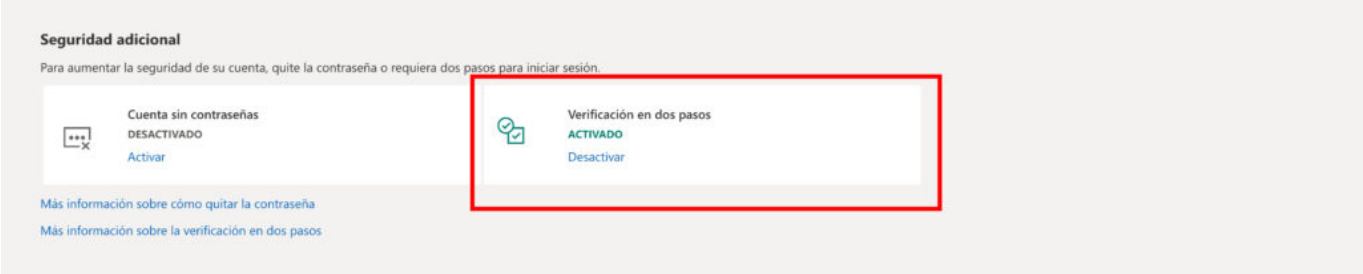
Importante:

Puede también escoger la opción de aplicación el cual le pedirá tener instalado una aplicación móvil. Ejemplo: Microsoft Authenticator o Google Authenticator.



Luego de colocar la cuenta de correo solo debe dar clic en siguiente y recibirá un email indicando la clave que debe agregar para validar su cuenta.

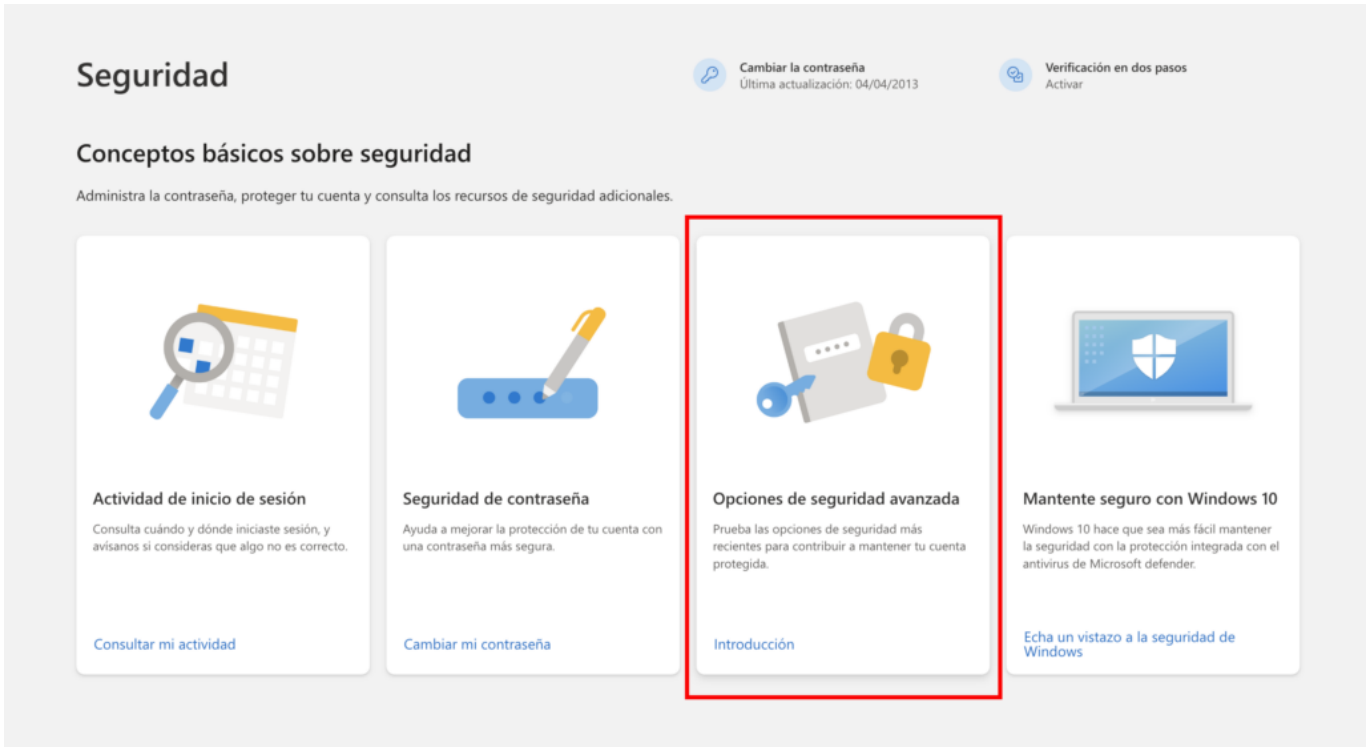
Luego de confirmar el código, su cuenta tendrá activada la **Verificación de dos pasos.**



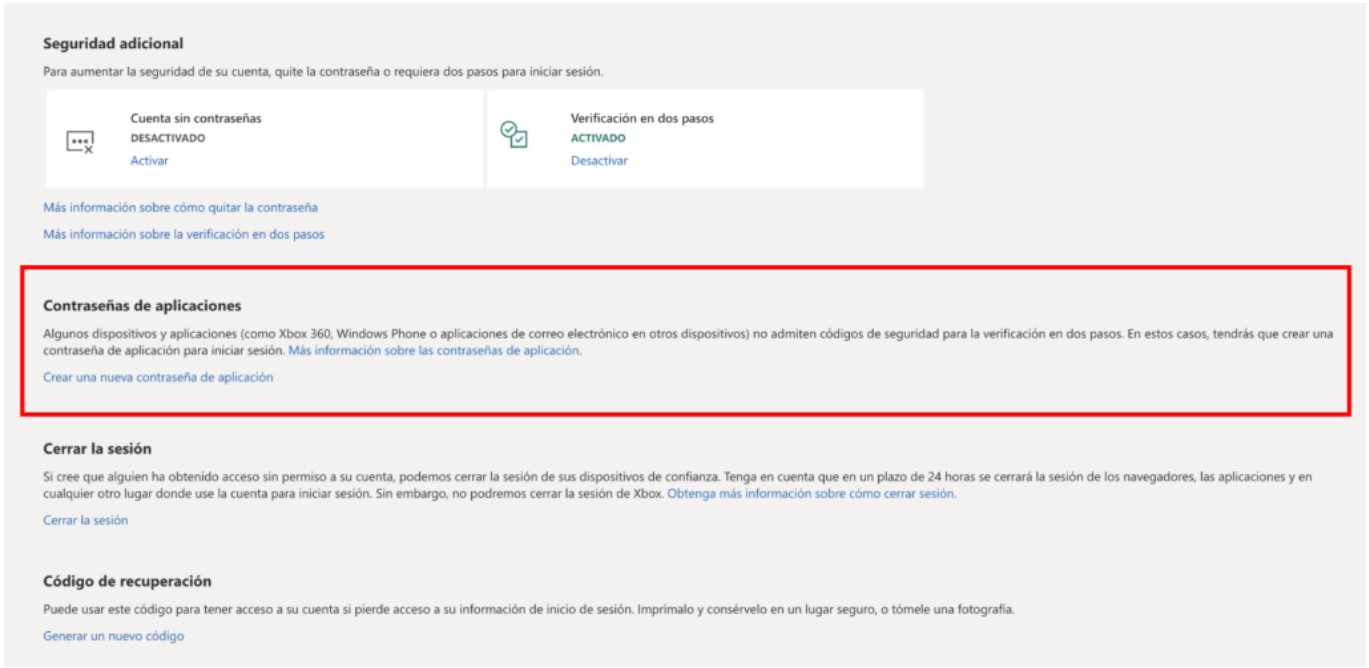
Segundo Paso.- Configurar una contraseña de Aplicación

Para crear una nueva contraseña de aplicación para una aplicación o dispositivo, sigue los pasos que se indican a continuación. Puedes repetir estos pasos de creación de una contraseña de aplicación para todas las aplicaciones o dispositivos que necesites.

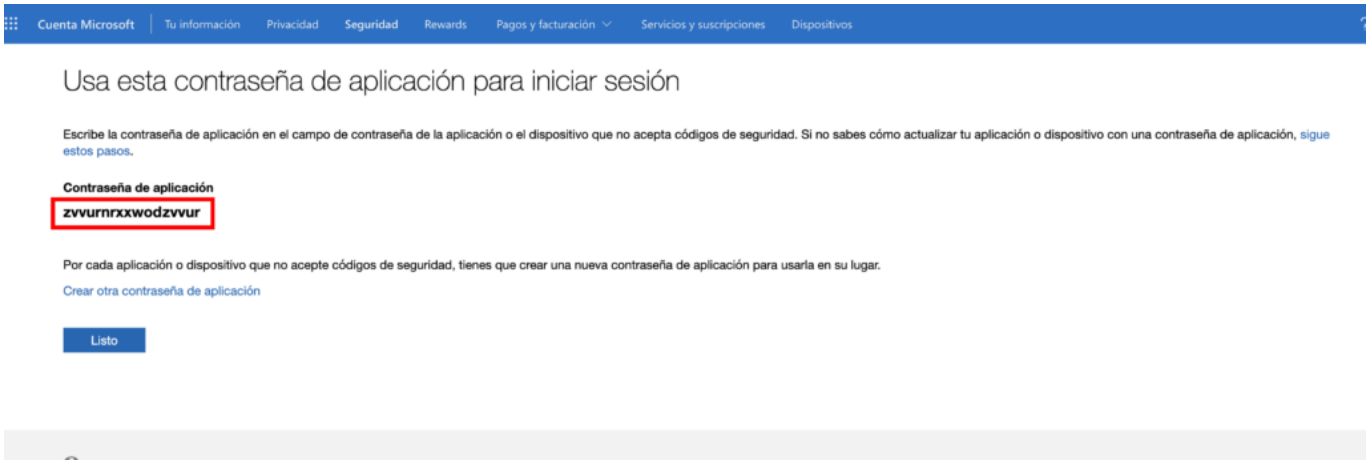
Dirígete a la página Conceptos básicos sobre seguridad e inicia sesión en tu cuenta de Microsoft y seleccionamos **«Opciones de Seguridad».**



En **Contraseñas de aplicación**, selecciona **«Crear una nueva contraseña de aplicación»**.



Se genera una nueva contraseña de aplicación que aparece en la pantalla.



Datos para Configuración

Luego de crear la contraseña de aplicación, podrá configurar su cuenta en el portal. Los datos de envio son los siguiente:

1. Correo/Usuario: sucorreo@outlook.com (en el caso de Office365 sería @sudominio.com)
2. Contraseña: La generada en el segundo paso.
3. Servidor SMTP : **outlook.office365.com**
4. Puerto: **587**
5. TLS : **SI**
6. Autenticación SMTP : **SI**

Aqui estaran disponibles los links a los modulos donde se puedan aplicar estas acciones :

Email Advanced : https://market.rocketbot.com/module/emailAdvanced

Outlook 365 : https://market.rocketbot.com/module/Outlook365

Studio Rocketbot – Variables internas

Rocketbot Studio cuenta con variables nativas que puedes utilizar para acceder a información rápidamente. Estas variables pueden ser utilizadas desde cualquier comando y no necesitas crearla en la sección de variables o generar una acción para generarlas.

Variable	Descripción
%rocketbot_last_status%	dict. Almacena el estado del último comando ejecutado. {‘status’: ‘True’, ‘message’: ‘setvar ‘, ‘img’: ”, ‘vars’: [], ‘ifs’: [], ‘extra’: [], ‘time’: ‘0.004066944122314453’}
%log_path%	str. Almacena la ruta del archivo log de la ejecución en curso
%base_path%	str. Almacena la ruta de la carpeta de rocketbot
%date%	str. Almacena la fecha y hora actual. El formato es YYYY-MM-DD HH:MM:SS.ms
%day%	str. Retorna el día al momento de usar la variable
%month%	str. Retorna el mes al momento de usar la variable
%year%	int. Retorna el año al momento de usar la variable
%hour%	int. Retorna la hora al momento de usar la variable
%minute%	int. Retorna el minuto al momento de usar la variable
%second%	int. Retorna el segundo al momento de usar la variable
%milisecond%	str. Almacena la parte de los milisegundos de la hora al momento de usar la variable
%machine%	str. Almacena el nombre de la máquina donde se está ejecutando el proceso
%tab%	str. Contiene un tab
%newline%	str. Contiene un salto de línea o nueva linea
%enter%	str. Contiene un salto de línea o nueva linea
%osname%	str. Almacena el nombre del sistema operativo
%username%	str. Almacena el nombre de usuario de la computadora
%db_path%	str. Almacena la ruta de la base de datos
%project_path%	str. Almacena la ruta de la carpeta de la base de datos. Para versiones superior a la 2024, corresponde a la carpeta del proyecto
%robot_name%	str. Contiene el nombre del robot

Variable	Descripción
	bool.
%production%	Retorna True si la ejecución es en producción. En caso contrario, retorna False

---

Server con Múltiples Usuarios

La duda recurrente es “¿Cómo es el licenciamiento en una máquina donde acceden múltiples usuarios al mismo tiempo?”.

Para este caso tenemos 3 opciones.

Opción 1:

Primero debemos revisar si dejando ***Rocketbot Studio*** en la raíz, pueden acceder todos los usuarios y utilizarlo, si es así, entonces sería 1 Licencia con X cantidad de ejecuciones en paralelo (la cantidad según el número de usuarios que se conectarán)

Opción 2:

Si no permite acceder a cada usuario a ***Rocketbot Studio*** instalado en la raíz (Opción 1), entonces debe ser una licencia por cada ambiente de usuario.

Opción 3:

Con la licencia Online se puede conectar cada usuario en cualquier máquina con ***Rocketbot Studio instalado***, siempre y cuando tenga disponible la cantidad de conexiones necesarias.

Automatización de Aplicaciones de Escritorio

Las aplicaciones de escritorio no son tan estandarizadas como los son las aplicaciones web, mientras que para aplicaciones web utilizamos en su mayoría comandos que **automatizan** el navegador, para aplicaciones de escritorio tenemos diferentes métodos según el tipo de aplicación. En esta entrada te mostraremos las **consideraciones** y **métodos** que debes conocer antes de comenzar a desarrollar un robot para automatizar **aplicaciones de escritorio**.

Procesos que pueden presindir de una aplicación

Existen algunos procesos que si bien utilizamos una aplicación para realizarlos, existen métodos que permiten a los robots hacerlo de forma más eficiente. Recuerda que automatizar no es solo hacer robots que repliquen el comportamiento humano frente a la computadora.

En este tipo de aplicaciones se encuentran aquellas actividades que se pueden realizar por terminal o scripts bash, aplicaciones que tienen su versión web y que cuentan con una API o que trabajan bajo algún protocolo. Estos procesos pueden ser:

- Automatización de correos electrónicos por conexión imap, smtp, pop.
- Manipulación de archivos como mover, renombrar, leer, crear, etc.
- Consultas a base de datos
- Aplicaciones con versión web como Slack, Google Drive, Telegram, One Drive
- Conexiones ssh o telnet
- Aplicaciones simuladas en una terminal

No necesitas tener instalada la aplicación para automatizar estos procesos, solo debes descargar el módulo correspondiente en nuestro Marketplace. Estos procesos en su mayoría permiten ejecuciones en paralelo.

Aplicaciones con su propio sistema de scripting

Algunas empresas ya conocen el poder de la automatización y nos permiten interactuar con sus aplicaciones a través de su propio sistema de scripting. De esta manera, podemos controlar la aplicación sin necesidad de utilizar el mouse o teclado y utilizar los métodos internos que tiene cada aplicación para automatizar nuestro proceso.

Para automatizar estas aplicaciones es necesario tener instalada la aplicación en la misma máquina donde se ejecutará tu robot. Aplicaciones de este tipo son: SAP, SAPB1, Excel, Word, Outlook y puedes encontrar módulos para automatizarlos en Marketplace. Si conoces alguna aplicación que necesitas automatizar y esta tiene sistema de scripting, comunicate con nosotros y desarrollamos el componente.

Dependiendo de la aplicación, puedes ejecutar más robots al mismo tiempo. Revisa si la aplicación permite abrir múltiples instancias.

Aplicaciones Java o Windows Forms

Aunque algunas aplicaciones no tengan su propio sistema de scripting, si pueden ser automatizados de esta forma a través de las dll de Java o Windows. Automatizar una aplicación por scripting es similar a automatizar una página web, solo necesitas obtener información del objeto que quieres controlar, como el título, la clase, el id, el tipo de objecto, etc.

Esto no puede obtenerse directamente de la aplicación, pero puedes usar

alguno de nuestros grabadores como el DesktopRecorder o JavaRecorder. Con esta herramienta puedes grabar las acciones que quieres realizar en la aplicación para luego replicar lo grabado en un robot. No es sencillo determinar que aplicaciones permiten automatizarlo por scripting. La mejor recomendación es descargar los grabadores y revisar si detectan tu aplicación.

La aplicación debe estar instalada en la misma máquina donde se ejecutará el robot y al hacer uso del mouse y/o teclado, no se puede trabajar con robots en paralelo que automaticen el mismo proceso

Otras aplicaciones

Si tu aplicación no se puede automatizar de las formas mencionadas anteriormente, puedes automatizar la aplicación por virtualización y/o comandos de teclado. Para esto, es necesario tener algunas consideraciones:

- Si el robot se ejecutará en un servidor, revisa la entrada RPA en servidores virtuales
- Si la aplicación permite utilizar combinaciones de tecla, utiliza el teclado por sobre virtualización. Puedes conocer los shortcut de la aplicación al ver una letra subrayada bajo la palabra donde quieres hacer click. Para replicar en Rocketbot, revisa la siguiente entrada Enviar Tecla
- Si se debe extraer información, revisa si puedes usar el portapapeles para copiar el contenido.
- Como se sacarán capturas de pantalla, se utilizará el mouse o el teclado. No se puede automatizar la misma aplicación en paralelo

---

Ejecutar en Paralelo

Rocketbot permite ejecuciones en paralelo, lo que quiere decir que podemos lanzar más de 1 robot al mismo tiempo.

Uno de los requisitos es contar con una licencia de Producción que permita este tipo de ejecuciones.

Al iniciar nuestro Rocketbot Studio, nos informará en la consola cuántos robots en paralelo nos permite:

> Tener en cuenta que al estar Rocketbot Studio abierto es considerado un proceso, por lo tanto si tengo un máximo de 2 procesos en paralelo, y requiero ejecutar 2 robots, debo cerrar el Studio.

Para iniciar más de un robot, lo podemos realizar a través de un .bat si estamos en Windows o desde un .sh si estamos en macOS/Linux.

Windows

El archivo .bat debe tener lo siguiente:

```
cd C:/ruta/a/carpeta/Rocketbot
START rocketbot.exe -start=robot1
START rocketbot.exe -start=robot2
```

macOS/Linux

El archivo .sh debe tener lo siguiente:

```
cd /ruta/a/carpeta/Rocketbot
./rocketbot -start=robot1 &
./rocketbot -start=robot2
```

Luego podemos ejecutar el .bat/.sh mediante un doble click o agregarlo a una tarea programada.

Para saber que procesos se pueden ejecutar en paralelo revise este link ¿que considerar para ejecucion en paralelo?

Buenas Prácticas en la creación de Bots

Para crear bots de una forma óptima se puede revisar el siguiente manual de Buenas Prácticas.

1. Robot
    a. Descripción en Comandos
    b. Nombre del Robot
    c. Variables
        a. Nombre de Variables
        b. Limpiar Variables
        c. Reglas de Variables
        d. Decodificación
        e. Variables heredadas Padre-Hijo
    d. Credenciales

    e. Validaciones
    f. Cerrar Procesos
2. Web
    a. Abrir Navegador
    b. Cerrar Navegador
    c. Cambiar a ventana
    d. Esperar por Objeto
3. Sistema
    a. Esperar
4. Scripts
    a. Ejecuta otro script Rocketbot
    b. Detener este Robot
    c. Detener todo y salir
5. Desktop
    a. Mover Mouse
    b. Mouse Click
    c. Alerta
    d. Enviar Tecla
6. Lógica
    a. IF
    b. While
7. XLSX
    a. Nuevo xlsx
    b. Abrir xlsx
    c. Contar Filas
    d. Obtener celda
8. Virtualización
    a. Buscar por Imagen
    b. Esperar por Imagen
    c. Tips
9. MySQL
    a. Configurar conexión MySQL
    b. Consulta MySQL
10. Email
    a. Configurar Servidor
    b. Listar todos los email IMAP
    c. Listar email nuevos IMAP
11. Excel
    a. Nuevo Libro
    b. Abrir archivo
    c. Guardar Archivo
    d. Contar Filas
    e. Obtener celda
    f. Cerrar Excel
12. Logs en Rocketbot
    a. Estructura de Logs de Rocketbot
    b. Visualizar errores en Rocketbot
    c. Recomendaciones para generar Logs
13. Robots en Paralelo
14. Escritorio Remoto

Video de ayuda Buenas Prácticas:

Robot

Buenas prácticas

Descripción en Comandos

Agregar descripciones en los comandos nos ayuda a comprender cuál es la acción específica que este realiza, lo que es de suma importancia a la hora de modificar un robot, ya sea para cambiar funcionalidades o identificar posibles errores.

Nombre del Robot

Nunca utilizar **espacios**, **ñ**, ni **caracteres especiales** en el nombre del robot, si el nombre fuera compuesto por varias palabras puede utilizarse el siguiente formato: snake_case, camelCase o **PascalCase**.

Variables

Nombre de Variables

En la definición de variables nunca se deben utilizar espacios, ñ, o caracteres especiales, en caso de utilizar palabras compuestas se pueden utilizar guiones bajos o comenzar la siguiente palabra con Mayúscula como por ejemplo: snake_case, camelCase o PascalCase.

Limpiar Variables

Es buena práctica limpiar las variables que no requieren valores fijos, al principio de nuestro bot, es recomendable utilizar el comando "**Limpiar Variable(s)**" del **módulo System++**, ya que si limpiamos una por una tendremos un comando por cada variable, lo que se pueden reducir a sólo uno con la siguiente instrucción:

Reglas de Variable

String

Cuando queremos trabajar con variables de tipo String en Rocketbot, debemos pasarlas a comillas dobles, ejemplo.

Tenemos una variable llamada **{texto}** la cual contiene lo siguiente:

Si queremos por ejemplo modificar parte del texto, podemos utilizar el método de string[1] llamado replace(), en este caso si solo pasamos la variable, nos arrojará un error de sintaxis:

Debemos indicar que la variable **{texto}** la trabaje como un string, y para eso debemos pasarla entre comillas dobles **"{texto}"**.

Si además nuestra variable es un string pero contiene saltos de líneas, entonces debemos pasarla entre triple comillas dobles y espacios **"""{texto}"""**, esto permite que nuestra cadena de texto utilice más de una línea, por ejemplo, ahora nuestra variable **{texto}** se ve de la siguiente forma:

[1] Revisar métodos de string

#	Nombre	Datos
1	texto	Esto es un nuevo texto con saltos de línea

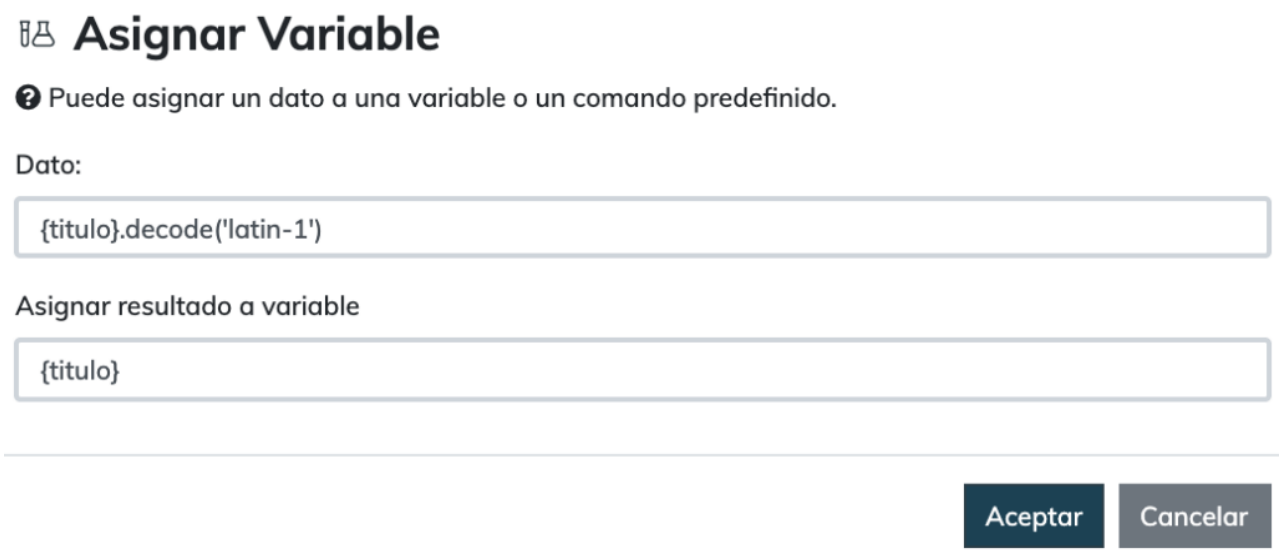
Si volvemos a ejecutar el comando anterior, nos arrojará un error:

Al pasarla a triple comilla doble y espacios, ejecutará el método sin problema:

Decodificación

Cuando tenemos variables codificadas, por ejemplo por haber obtenido el dato a través de JS, nos mostrará el texto de esta forma:

Para utilizar esta variable como STRING se debe utilizar {variable}.decode() o {variable}.decode('latin-1') en el comando **Asignar Variable**.



Quedará de esta forma y ya podemos utilizar la información correctamente:

Variables heredadas Padre-Hijo

Las variables del **PADRE** pueden ser utilizadas por todos los robots de la imagen, el **HIJO1** por ejemplo, no puede utilizar las variables de sus hermanos (**HIJO2–HIJO3**), el **NIETO1** puede usar las variables del **PADRE** y del **HIJO1**, pero no las del **HIJO2 – HIJO3**

Tiempos de ejecución (Esperar por Objeto)

En Studio (ambiente Desarrollo) el robot se ejecuta paso a paso con datos de debug que viajan al intérprete y regresan con datos de ejecución.
Esto hace que el comando tenga un tiempo de ejecución y milisegundos por cada comando para interpretar la respuesta. Significa que el robot en desarrollo es mucho más lento que en producción donde no necesita tiempos de debug.
Ej: En un robot extenso de manejo de Excel un robot tarda una hora en desarrollo pero solo 10 minutos en producción.
Para solventar esto y como parte de las buenas prácticas se debe usar los comandos **Esperar por Objeto**, de cada tipo de tecnología (Web, Desktop, Virtualización, etc.).

Credenciales

Como buena práctica es importante utilizar un baúl de credenciales para obtener los datos y no dejarlos fijos.

Si tenemos Mac o Linux, podemos utilizar el siguiente módulo Credentials_, si estamos en Windows, debemos usar el siguiente: Windows Credential Manager.

Podemos encontrar más información acerca de cómo realizarlo, en la documentación de los manuales linkeados arriba.

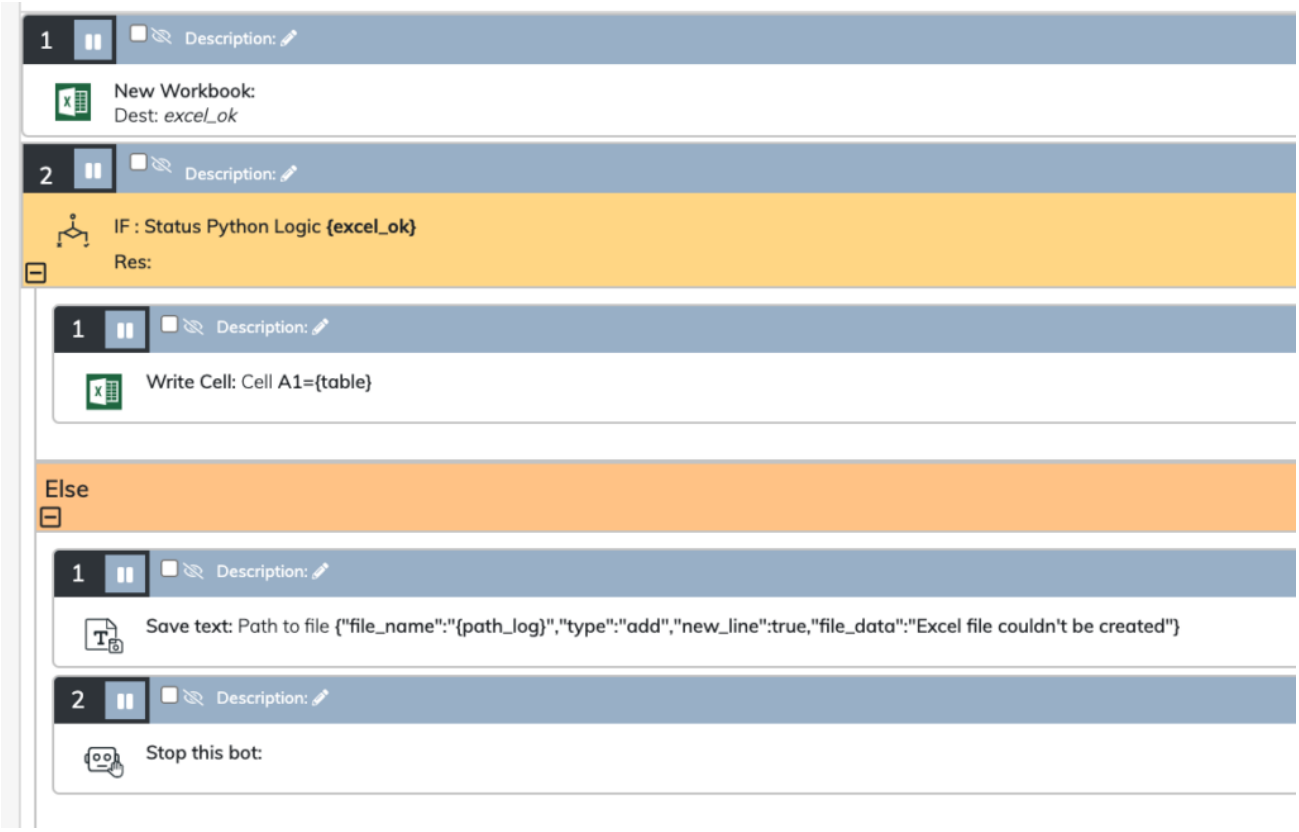
Validaciones

Se deben validar todos los comandos y/o módulos que lo permitan, por ejemplo si es una conexión a una BD, Correo, FTP, Abrir un Archivo, etc. se debe retornar a una variable y tomar una decisión con un IF.

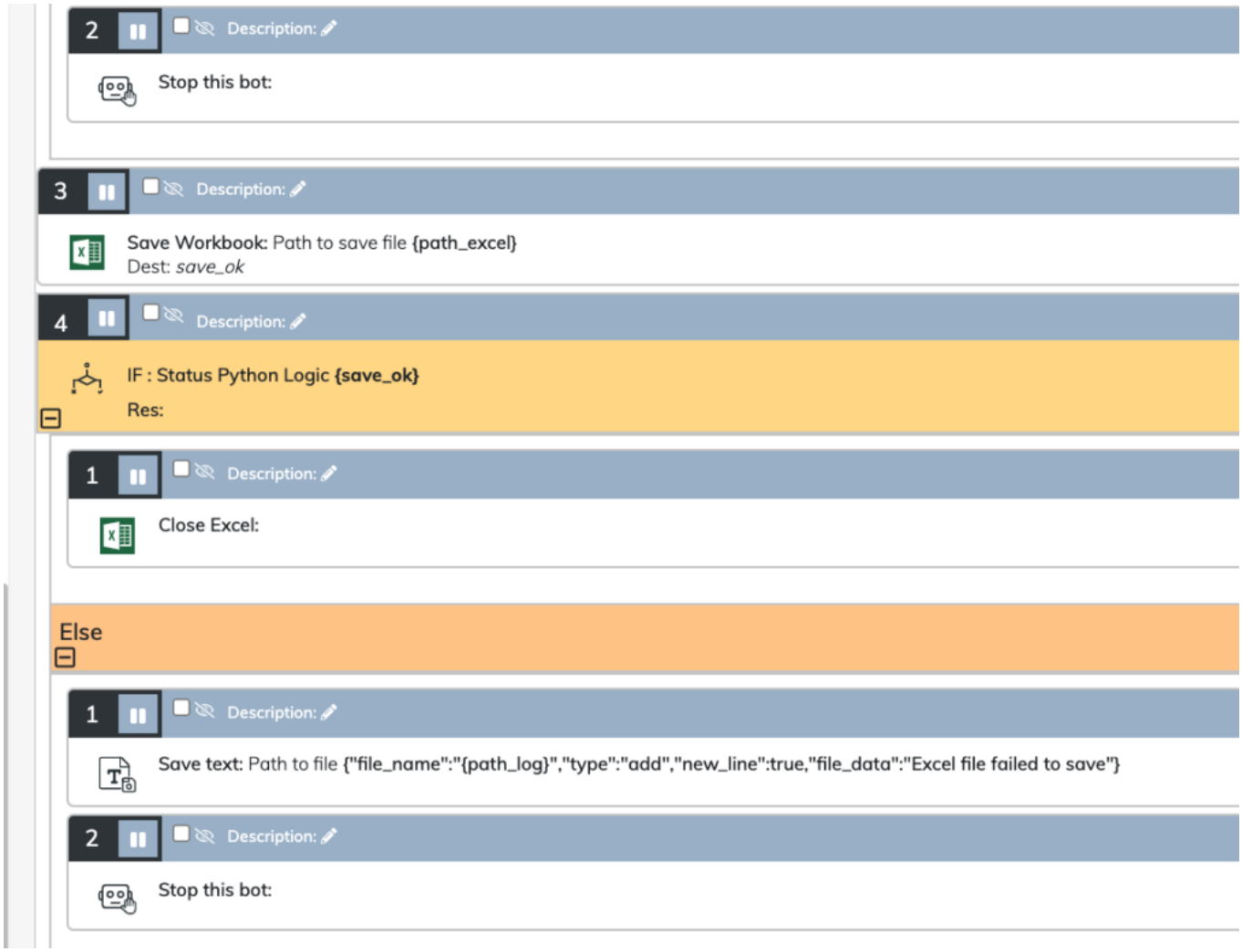
Esto nos permite poder controlar errores y tomar una decisión para que el robot no continúe ejecutándose cuando faltan partes del proceso o cuando no se realizó algo de forma correcta.

Ejemplo:

Crear un archivo excel y validar su correcta creación, si está ok, entonces se escribe en el archivo, sino, se deja un mensaje en el log y se detiene el robot.



Luego realizamos el guardado del archivo, y se valida, si se guardó de forma correcta, entonces cerramos la aplicación, sino, se deja un mensaje en el log y se detiene el robot.



Cerrar Procesos

Todos los comandos y/o módulos que permitan cerrar conexiones, se deben agregar al proceso, por ejemplo, cerrar una conexión a una BD, a un FTP, Correo, Cerrar un archivo Excel, XML, etc.

Es necesario siempre cerrar todo lo que se abre con Rocketbot, ya que esto nos permite liberar recursos y evitar errores de ejecución, si el comando o módulo no permite cerrar la conexión o los procesos no son cerrados de forma correcta, se pueden utilizar los módulos killProcess y/o killApp para terminar de matar procesos zombies.

Ejemplo de un error:

Problema

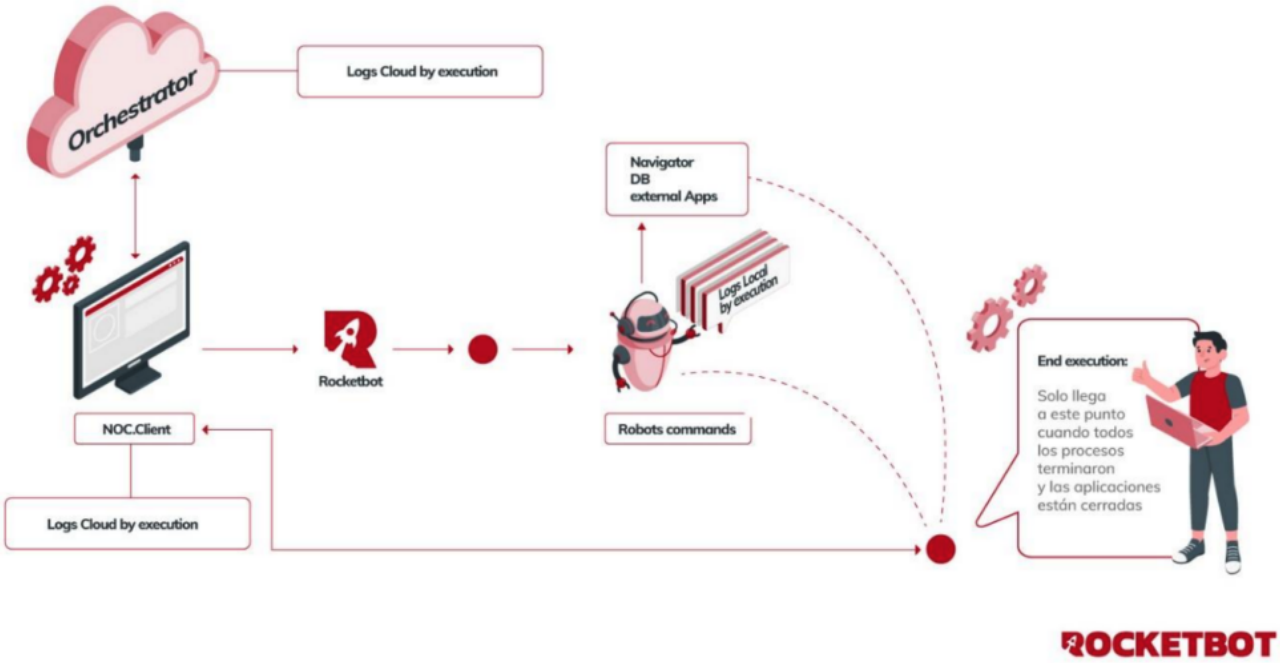
No se realizó la ejecución de bots desde el Orquestador dado que un proceso no finalizó.

Causa

No llega la respuesta de proceso finalizado al Orquestador. Esto ocurre cuando se trabaja con un proceso que queda en segundo plano o no finaliza al terminar el robot. Por ejemplo, abrir alguna aplicación.

Solución

Agregar el comando killProcess y/o killApp al final del flujo para que cierre cualquier proceso zombie que pudiera estar activo e impida notificar al Orquestador el término del proceso.



Web

Abrir navegador

Opciones:

- **Selecciona navegador:** Chrome/Firefox/IE
- **URL de Servidor:** Indicar la url de la página a la cual queremos acceder (Si no se elige ninguna abre por defecto https://www.google.com). Admite variables
- **Modo de Apertura:** Síncrona/Asíncrona.

Sincrónico es el método normal de apertura, el cual espera para continuar con los comandos, no se debe cambiar.

Buenas prácticas

Es óptimo agregar la url en una variable, pensando en escalabilidad, si en algún momento se requiere que la variable sea dinámica o si necesita cambiar, lo ideal es que Rocketbot lea algún archivo de configuración y setear el valor a la variable para no estar abriendo y modificando el robot.

Error común

Los navegadores son controlados por web drivers, estos se encuentran dentro de la carpeta ***Rocketbot/drivers/{sistema operativo}/{browser}***, los mismos deben ser actualizados si se actualiza la versión del navegador y el robot comienza a presentar problemas.

Los drivers los puedes encontrar en https://www.seleniumhq.org/download/

Error de driver al intentar abrir navegador, esto sucede cuando el navegador se actualiza, para solucionarlo se debe actualizar el driver según la versión del browser.

Chrome: https://chromedriver.chromium.org/downloads
Mozilla: https://github.com/mozilla/geckodriver/releases/tag/v0.29.0

Descargar y descomprimir el driver, copiar el archivo en la siguiente ruta:

Rocketbot/drivers/{sistema operativo}/{browser}

Ej: Rocketbot/drivers/win/chrome

Cerrar Navegador

Nos permite cerrar el navegador abierto con Rocketbot, cierra todo incluyendo las pestañas abiertas.

Buenas prácticas

Siempre que realicemos un robot de tipo Web, y ya no necesitemos utilizar más el navegador, debemos agregar el comando **Cerrar Navegador**, ya que esto además de cerrar el Browser, mata el proceso del driver abierto con Rocketbot. El cual se abre en cada ejecución, por lo que si no se mata y se acumulan demasiados, podría consumir bastante recurso de la máquina.

> Todo lo que se abre con Rocketbot, es óptimo que se cierre al final del proceso o cuando ya no se utilice más, ej: navegador, conexión a BD, conexión a email, Excel, etc.

Cambiar a ventana

Cuando la página que estamos controlando abre otra ventana (al clickar un botón, enlace, etc.) este comando nos permite cambiarnos a la nueva pestaña y dejarla en foco para poder trabajar con los elementos.

En este ejemplo tenemos en foco la primera pestaña (google),

si queremos hacer un click en la página de Rocketbot (pestaña 2), por ejemplo

al botón Descargar, obtendremos un error como este, donde nos indica que nose encuentra el elemento.

Debemos entonces primero cambiarnos de ventana y una vez quede en foco ya podremos realizar el click. Podemos cambiarnos indicando el handle o una de las siguientes opciones las cuales indican la posición de las pestañas abiertas:

Secuencia:

Buenas prácticas

Cada vez que nos cambiamos a otra ventana debemos utilizar el comando **Esperar por Objeto** con su respectivo **IF** de validación, ya que estamos ingresando a otra página y debemos asegurarnos de que cargó correctamente.

Esperar Por Objeto

Este comando permite esperar por un elemento en la web, dándole segundos máximos de espera hasta que nos retorna una respuesta (True o False, si lo encontró o no respectivamente). Se debe utilizar siempre que abrimos el navegador para confirmar si la página realmente cargó, también cuando clickeamos algún enlace o botón y nos abre otra pestaña, en general siempre que “cambiemos” de página, para así validar que cargó y los elementos ya existen.

Opciones:

- **Dato a buscar:** Valor del identificador del elemento.
- **Tipo de dato:** Tipo de identificador del elemento.
- **Esperar Antes:** Cantidad de segundos antes de que el comando “Esperar por Objeto” se ejecute. Es una espera fija, la cantidad de segundos que definamos se respetarán si o si, por lo cual se recomienda que sean tiempos bajos, ojalá no más de 2-5 segundos.
- **Acción Esperar máx:** Cantidad de segundos máxima que esperará el elemento. Es una espera variable, si ingresamos 10 segundos, no significa que los esperará si o si, sino que como máximo esperará esa cantidad, si nuestro elemento aparece antes en la página, entonces obtendremos antes nuestra respuesta y se continuará con el siguiente comando, en cambio, si nuestro elemento no aparece, estará consultando hasta que se cumplan los 10 segundos y luego nos retornará un False en nuestra variable.

- **Esperar Después:** Cantidad de segundos que esperará luego de que se ejecute el comando “Esperar por Objeto” y continúe con el siguiente. Es una espera fija, la cantidad de segundos que definamos se respetarán si o si, por lo cual se recomienda que sean tiempos bajos, ojalá no más de 2-5 segundos.
- **Asignar resultado a variable:** Variable donde retornaremos el valor de si encontró o no el elemento dentro de los segundos asignados (True o False).

En este caso se está esperando 10 segundos a que aparezca el input de Google para realizar una búsqueda.

Buenas prácticas

Esperar Antes y **Esperar Después**: Son tiempos fijos, y son campos opcionales, en caso de utilizarlos, se deben indicar tiempos bajos (ej: 2-5 segundos), ya que utilizar esperas fijas no es buena práctica.

Acción Esperar máx: Es un tiempo variable, y es un campo obligatorio, siempre debe ir un valor ya que sino se indica, el comando no estará cumpliendo su función, como es un tiempo variable y no fijo, a diferencia de los dos inputs indicados arriba, el tiempo indicado debe ser más amplio.

Siempre que utilizamos **Esperar por Objeto** al inicio de nuestro flujo, se debe utilizar el comando **IF** del menú lógica, ya que tendremos una respuesta (True o False) en nuestra variable, en este ejemplo, llamada **{wait}**, con esa respuesta debemos tomar una decisión, qué hacer si es **True** (seguir el flujo normal) o qué hacer si es **False** (arrojar excepción, cerrar navegador, escribir en un log, informar por correo, etc.)

La secuencia sería la siguiente:

- Abrir la página de google.
- Esperar 10 segundos máximo a que el input de búsqueda cargue en la página.
- Tomar una decisión con el IF
  - True: Clickear el input de búsqueda, enviar el texto a buscar y realizar un Enter.
  - False: Cerrar el navegador y detener el robot.

Sistema

Esperar

Hace esperar al sistema por N segundos y luego sigue ejecutando los comandos.

Utilizado para casos especiales, **no se deben utilizar esperas fijas** ya que solo aumentarán el tiempo de ejecución del robot, para Web se debe utilizar **Esperar por Objeto** y para Desktop con Virtualización se debe utilizar **Esperar por Imagen**.

Scripts

Ejecuta otro script RocketBot

Este comando nos sirve para llamar un robot desde otro, donde podemos compartir variables y dividir tareas entre cada robot.

Nos mostrará todos los robot que tenemos en nuestra base de datos para ejecutarlos.

Buenas prácticas

Como buena práctica siempre es recomendado seccionar el flujo (proceso), en varios **mini-bots**. Esto facilita el poder modificar el proceso, revisar dónde está fallando, corregirlo y reutilizar comandos.

Ejemplo:

Un robot principal (**padre**), que abra un navegador con una url, luego un mini robot (**hijo_descarga**), que descargue un archivo y otro mini bot (**hijo_mail**) que tome el archivo y lo envíe por mail.

Robot **"padre":**

1. Abre el navegador con una URL específica.
2. Maximiza la ventana.
3. Ejecuta un robot hijo (**hijo_descarga**).
4. Cierra el navegador.

Variables:

N/A

Robot "**hijo_descarga**":

1. Cuenta la cantidad de archivos existentes en la carpeta de Descarga.
2. Realiza click en el botón Descargar de la página abierta.

3. Espera hasta que la descarga haya finalizado.
4. Ejecuta un robot hijo (hijo_mail)

Variables:

Robot “**hijo_mail**”:

1. Configura un correo Gmail.
2. Envía un correo adjuntando el archivo descargado en el robot anterior (utilizando la variable {path} del robot hijo_descarga).

Variables:

Detener este robot

Detiene la ejecución del robot actual y continúa con los siguientes, para utilizarlo solo se agrega el comando.

Buenas prácticas

Si tenemos una validación IF por ejemplo, para validar que la página cargó correctamente, este comando nos serviría para no continuar con las demás instrucciones. En caso de que algo falle, se puede detener el robot.

Ejemplo:

En este caso al caer en el Else, detendrá el Robot actual, por lo que no continuará con las siguientes instrucciones como la ejecución del bot hijo llamado Excel.

Detener todo y salir

Este comando detiene el Script en curso y sus padres.

Buenas prácticas

Misma función que el comando anterior, con la diferencia de que este detendrá tanto el robot actual como sus robots padres.

Desktop

Mover Mouse

Mueve el mouse hasta la posición indicada por coordenadas X e Y (ej: 1045,207).

Para obtener las coordenadas podemos utilizar el comando **Obtener coordenadas del mouse**.

Buena práctica

Este tipo de instrucciones no se deben utilizar en procesos **Web**, son para utilizarlas en automatizaciones de aplicaciones de escritorio, sin embargo si requerimos mover el mouse para realizar algún click, es mejor utilizar el comando **Click en Imagen**, el cual busca la imagen en toda la pantalla, por lo tanto si el objeto cambia de posición lo encontrará igualmente, no así **Mover Mouse**, que solamente irá a una posición específica.

Mouse Click

Hace click en donde está posicionado el Mouse, si queremos que lo realice en un lugar específico, debemos utilizar primero el comando **Mover Mouse** para indicarle la posición y una vez esté donde lo queremos, utilizamos **Mouse Click**.

Podemos utilizar:

- **left** -> para hacer click con el **botón izquierdo**
- **right** -> para hacer click con el **botón derecho**
- **middle** -> para hacer click con el **botón central del Mouse**

Buena práctica

Este tipo de instrucciones no se deben utilizar en procesos **Web**, son para utilizarlas en automatizaciones de aplicaciones de escritorio, sin embargo si requerimos realizar algún click, es mejor utilizar el comando **Click en Imagen**, el cual busca la imagen en toda la pantalla, por lo tanto si el objeto cambia de posición lo encontrará igualmente, no así **Mouse Click**, que solamente lo hará en una posición específica.

Alerta

Este comando muestra un cartel de mensaje. Agregamos el comando: Mostrará la ventana de alerta: También le podemos pasar una variable para que nos muestre su valor. Nos mostrará la ventana de alerta con el valor que tiene la variable que ingresamos.

Buena práctica

Las alertas son sólo cuando estamos desarrollando un robot, pero se deben eliminar de un robot final, ya que éstas pausan la ejecución y el proceso no continuará hasta que sea aceptada.

Enviar Tecla

Al igual que el comando **Enviar Texto Web** éste sólo acepta una opción, si se escribe un texto (o una variable con texto) no se puede agregar una tecla especial, o no funcionará. Al revés lo mismo, si queremos enviar una tecla especial, no podemos también pasar un texto.

Buenas prácticas

No utilizar este comando en automatizaciones **Web**, en ese caso se debe utilizar **Enviar Texto Web**, ni en aplicaciones Office (Excel, Word, Powerpoint) para ellas se deben utilizar los menús o módulos respectivos.

En el input de Texto, podemos concatenar un texto y una tecla, de la siguiente forma:

Si se necesita repetir la misma tecla varias veces, también se puede indicar de esta forma:

Podemos encontrar más información respecto a combinaciones a utilizar en el siguiente link: pywinauto además podemos utilizar teclas predefinidas con el el módulo Keyboard.

Lógica

IF

La sentencia if se utiliza para ejecutar un bloque de código si, y sólo si, se cumple una determinada condición. Por lo tanto, IF es usado para la toma de decisiones.

Buenas prácticas

Nunca se debe dejar un IF vacío y poner solo la lógica en el Else, esto no es permitido en Python por lo tanto producirá fallos en nuestro robot.

Ejemplo:

La forma correcta es la siguiente, donde el Else si puede ir vacío:

While

Con la sentencia While podemos ejecutar un ciclo mientras se cumpla una condición, lo cual nos permite ejecutar instrucciones múltiples veces.

Buenas prácticas

La condición debe estar relacionada con una variable que tenga un cambio en la condición para que en algún momento esta se cumpla y se termine el ciclo, sinó se corre el riesgo de generar un bucle infinito.

XLSX

Si trabajamos con este menú no es necesario tener instalado Microsoft Excel, ya que no trabaja con la aplicación, abre o crea un archivo en segundo plano y no veremos que este se abra, todo pasará por debajo.

Nuevo xlsx

Nos permite crear un nuevo archivo XLSX en memoria, podemos indicar un Identificador (puede ser un número, letra o palabra) en el caso de que necesitemos crear y trabaja con más de un archivo xlsx abierto por Rocketbot, luego si necesitamos ir moviéndonos entre uno u otro, lo tendremos que realizar a través de su ID, también podemos indicar una variable, donde obtendremos un True si se puedo crear un nuevo archivo o False en caso contrario.

Buenas prácticas

Retornar a una variable para validar si se pudo abrir de forma correcta o no y con eso tomar una decisión a través del comando IF.

Ejemplo:

Abrir xlsx

Nos permite abrir un archivo XLSX en memoria, debemos indicar la ruta donde se encuentra, y al igual que el comando anterior también podemos indicar un Identificador en el caso que necesitemos abrir más de un archivo, además de la variable donde obtendremos un True si se pudo abrir de forma correcta, o un False en caso contrario.

Buenas prácticas

Retornar a una variable para validar si se pudo abrir de forma correcta o no y con eso tomar una decisión a través del comando IF.

Ejemplo:

Contar Filas

Este comando nos permite obtener la cantidad total de filas con información del archivo xlsx. Por defecto contará las filas de la primera hoja, si necesitamos trabajar con otra, primero debemos cambiarnos con el comando Cambiar de Hoja.

Buenas prácticas

Siempre luego de Abrir un archivo xlsx, debemos contar la cantidad de filas que tiene, ya que en el caso de que la cantidad sea variable y en otra ocasión el archivo tenga más o menos filas, es decir, en otra ocasión el archivo tenga más o menos filas tendríamos que estar modificando el comando **Obtener celda** de forma manual, lo cual no es óptimo, por lo tanto siempre se debe trabajar con una variable que contenga la cantidad de filas.

Obtener celda

Este comando nos permite obtener una celda o un rango de celdas y almacenar la información en una variable, al obtener un rango nos devolverá una lista o una lista de listas.

Buenas prácticas

Cuando trabajamos con un rango (ej: A1:F3) no debemos pasar un número fijo al final de este, sino, debemos utilizar el comando **Contar Filas** y asignar el valor a una variable, esta variable es la que se debe pasar al final del rango, ej: **A1:F{filas}**, esto en el caso de que la cantidad de datos en el Excel sea variable, para no estar modificando el comando cada vez que la cantidad de filas cambia, se debe utilizar esta buena práctica.

Virtualización

Este menú es nuestra última opción para realizar una automatización, el orden de prioridad sería:

1. Comandos nativos y/o módulos para Web, Office, Email, Base de datos, SAP, FTP, Archivos, etc.
2. Si necesitamos controlar una aplicación de escritorio, debemos primero intentar con los grabadores, Desktop Recorder o Java Recorder, si ninguno de los dos nos sirve, entonces la última opción será Virtualización ya que se deben considerar ciertas cosas como por ejemplo:
   a. La resolución del PC afectará al bot, por ejemplo, si se desarrolla un bot en X computador y luego se pasa al computador del cliente o donde se ejecutará finalmente, lo más probable es que no funcione correctamente, ya que se tienen resoluciones y componentes distintos.
   b. Las imágenes siempre deben estar enfrente para que el bot las reconozca, esto quiere decir que una persona no puede utilizar el computador al mismo tiempo que el robot, por lo tanto no se podrán realizar ejecuciones en paralelo.
   c. La pantalla no se puede bloquear, por lo que eso se debe desactivar o realizar ajustes para que no suceda.

> **Considerar:** Al trabajar con Virtualización, al sacar una captura de pantalla con los comandos del menú que lo permiten, siempre tomará la pantalla principal por defecto.

Buscar por Imagen

Busca una imagen de referencia en la pantalla y almacena el resultado en una variable retornando True si la encontró dentro del tiempo definido o False en caso contrario.

Buenas prácticas

Al igual que el comando **Esperar por Objeto** para la parte web, en Virtualización debemos utilizar este comando para asegurarnos de que el elemento con el cual queremos trabajar, existe, y luego tomar una decisión con un **IF**, si nos retorna **True**, osea que sí lo encontró, entonces dentro de ese bloque realizamos nuestro flujo, si nos retorna **False**, osea que no lo encontró entonces debemos decidir qué hacer.

Esperar por Imagen

Espera una imagen de referencia en la pantalla y almacena en una variable si está visible o no.

Realiza la misma función que el comando **Buscar por imagen.**

Buenas prácticas

Al igual que el comando **Esperar por Objeto** para la parte web, en Virtualización debemos utilizar este comando para asegurarnos de que el elemento con el cual queremos trabajar, existe, y luego tomar una decisión con un **IF**, si nos retorna **True**, osea que sí lo encontró, entonces dentro de ese bloque realizamos nuestro flujo, si nos retorna **False**, osea que no lo encontró entonces debemos decidir qué hacer.

Tips

Windows

Estos son algunos casos donde los comandos de imagen
no funcionen correctamente, por lo tanto se debe revisar resolución o ejecutar Rocketbot como administrador.

1. **Si no realiza los click en las imágenes**
    a. Instalar lo siguiente:
        i. NET Framework: Download .NET Framework | Free official downloads
        ii. Visual C++: Descargas más recientes compatibles de Visual C++
    b. Revisar resolución
        i. Si la configuración de pantalla está aumentada, dejarla al 100% y probar.

> **Importante:** Si se realizan cambios de escala, puede ser necesario cerrar sesión y volver a abrir para que los cambios se guarden.

2. Si realiza click en ciertas imágenes pero no en todas.

Por ejemplo realiza click en elementos del escritorio o barra de tareas pero no en una aplicación de escritorio interna, además al intentar abrir la aplicación con Rocketbot aparece lo siguiente:

Ejecutar Rocketbot como administrador y probar.

3. Si realiza click en otra parte

Especificar alguna referencia, por ejemplo si tenemos dos elementos iguales, debemos marcar como referencia algo que no se repita y como foco el lugar

donde realizar el click.

Por ejemplo acá seleccionamos como referencia el label Username el cual no se repite, y el foco es el input, lo que hará será calcular las coordenadas desde la imagen de referencia hasta la indicada en foco y hará click a lo que se encuentre en esa posición.

> Si sigue realizando el click en otra parte, y se está trabajando desde un escritorio remoto, debemos cambiar la resolución antes de ingresar al servidor.

Por ejemplo, el foco (donde hacer el click) es el ícono de Rocketbot, pero el click lo realiza en el punto rojo:

Configuración:

4. Si sigue sin encontrar las imágenes y es un Windows Enterprise (win 7 o win 10), revisar si está la siguiente .dll sino, agregarla:

- DLL: api-ms-win-downlevel-shlwapi-l1-1-0.dll

5. Virtualización en Windows server 2012 mediante rdp

En caso que las funciones de virtualización para “click en imagen” o “esperar por imagen” no funcionara correctamente podemos revisar lo siguiente:

Que no esté habilitada la opción de cambiar el nivel de escala:

Adicional a esto verificar que esté instalada las características dentro de “Interfaz de usuario e Infraestructura” en especial Experiencia de escritorio:

macOS

Permiso denegado al hacer OCR

Solución:

1. Abrir el terminal e ir a la ruta indicada en el error, ej: **cd /Users/usuario/Desktop/Rocketbot/drivers/mac/tesseract**
2. Escribir el comando **ls-l** para visualizar los permisos.
3. Veremos que en este caso no tiene permisos de ejecución:

4. Le damos los permisos con uno de los siguiente comandos:

- ejecución para todos los usuarios: **chmod 777 tesseract**
- ejecución sólo para el owner: **chmod 744 tesseract**

- Volvemos a ejecutar un **ls -l** para revisar los cambios:

- Ahora ejecutamos nuevamente el comando de OCR

MySQL

Configurar conexión MySQL

Para conectarnos con MySQL, debemos indicar la URL del servidor, puerto, nombre de la BD, usuario y contraseña, además podemos retornar el resultado a una variable donde obtendremos un True si la conexión fue exitosa, o un False, en caso contrario y podemos utilizar un ID en caso de que nos conectemos a varias Bases de datos MySQL.

Buenas prácticas

Siempre agregar una variable para retornar el status de conexión y con eso validar a través del comando IF para tomar una decisión.

Consulta MySQL

Realiza una consulta MySQL (Select, insert, delete, update), el resultado lo retornamos en una variable para posteriormente trabajar con la información.

Buenas prácticas

En el ambiente de Desarrollo de Rocketbot las consultas se deben realizar con límites y no traer todos los registros, ya que si son demasiados, Rocketbot puede detenerse por consumo de memoria, esto para realizar pruebas, ya en el ambiente de producción igualmente es recomendable seccionar la cantidad de

registros consultados.

Email

SMTP – IMAP

Configurar Servidor

Configura un servidor SMTP e IMAP con SSL para enviar y recibir correo. Al abrir el comando, por defecto vienen los servidores y puertos de gmail, debemos añadir el correo y su contraseña.

Buenas prácticas

Variables:

Es buena práctica, utilizar variables tanto en el nombre de usuario como en el caso de la contraseña y no escribirlos directamente en el comando. También debemos recordar siempre borrar las credenciales si por alguna razón tuvieras que compartir el robot.

Conexión:

Siempre se debe validar la correcta conexión de la cuenta de correo, en el caso de los **módulos**, éstos permiten validar la conexión retornando el resultado a una variable, por lo tanto esto se debe validar con un IF.

Ejemplo:

Una vez que ya no utilizaremos nada respecto a correos, debemos cerrar la conexión con el comando "**Cerrar Conexión**" del módulo que estemos utilizando.

Módulos:

Se recomienda utilizar los módulos específicos para los distintos tipos de servidores de correo, pero si se requiere utilizar uno corporativo o que por el momento no tenga su respectivo módulo, es óptimo utilizar este comando para configurar y complementarlo con el módulo Email Advanced en vez de utilizar los comandos nativos que son más limitados.

Listar todos los email IMAP

Este comando nos permite obtener una lista de todos los email que coincidan con el filtro[2] indicado, debemos retornar el resultado a una variable para trabajar con la información.

Buenas prácticas

Una vez que obtenemos el resultado de listar email es necesario decodificar los datos.

[2]Filtros de email

La decodificación se realiza con el comando Asignar Variable y la función de Python decode().

Ejemplo:

Quedará de la siguiente forma:

> Si queremos realizar el decode para toda la lista obtenida, debemos recorrerla con un For.

Listar email nuevos IMAP

Este comando nos permite obtener una lista de todos los email que coincidan con el filtro[3] indicado, la diferencia con el comando anterior es que este nos traerá los id's sólo de los correos que no hayan sido leídos aún. Debemos retornar el resultado a una variable para trabajar con la información.

[3]Filtros de email

Al igual que el comando anterior, también debemos decodificar los id's entregados.

Integración con Aplicaciones

Excel

Si trabajamos con este menú si es necesario tener instalado Microsoft Excel, ya que trabaja con la aplicación, abre o crea un archivo en primer plano y veremos que este se abre.

Nuevo Libro

Nos permite crear un nuevo archivo Excel, podemos indicar un Identificador (puede ser un número, letra o palabra) en el caso de que necesitemos crear y/o trabaja con más de un archivo Excel abierto por Rocketbot, luego si necesitamos ir moviéndonos entre uno u otro, lo tendremos que realizar a través de su ID, también podemos indicar una variable, donde obtendremos un

True si se puedo crear un nuevo Excel o False en caso contrario.

Buenas prácticas

Retornar a una variable para validar si se pudo abrir de forma correcta o no y con eso tomar una decisión a través del comando IF.

Ejemplo:

Abrir archivo

Nos permite abrir un archivo Excel existente, debemos indicar la ruta donde se encuentra, y al igual que el comando anterior también podemos indicar un Identificador en el caso que necesitemos abrir más de un archivo, además de la variable donde obtendremos un True si se pudo abrir de forma correcta, o un False en caso contrario.

Buenas prácticas

Retornar a una variable para validar si se pudo abrir de forma correcta o no y con eso tomar una decisión a través del comando IF.

Ejemplo:

Guardar Archivo

Este comando guarda el archivo excel en la ruta que especifiquemos (usando el botón buscar o pasando una variable), es posible guardar el resultado de la operación en una variable, donde obtendremos un True si fue exitosa o un False en caso contrario.

Buenas prácticas

Retornar a una variable para validar si se pudo guardar de forma correcta o no y con eso tomar una decisión a través del comando IF.

Ejemplo:

Contar Filas

Este comando nos permite obtener la cantidad total de filas con información del archivo Excel. Por defecto contará las filas de la primera hoja, si necesitamos trabajar con otra, primero debemos cambiarnos con el

comando **Cambiar de Hoja**.

Considerar que sólo contará la cantidad en la columna A, por lo tanto si necesitamos cambiar de columna podemos utilizar el comando Contar Filas del módulo Advanced Excel.

Buenas prácticas

Siempre luego de Abrir un archivo Excel, debemos contar la cantidad de filas que tiene, ya que en el caso de que la cantidad sea variable, es decir, en otra ocasión el archivo tenga más o menos filas tendríamos que estar modificando el comando **Obtener celda** de forma manual, lo cual no es óptimo, por lo tanto siempre se debe trabajar con una variable que contenga la cantidad de filas.

Obtener celda

Este comando nos permite obtener una celda o un rango de celdas y almacenar la información en una variable, al obtener un rango nos devolverá una lista o una lista de listas.

Buenas prácticas

Cuando trabajamos con un rango (ej: A1:F3) no debemos pasar un número fijo al final de este, sino, debemos utilizar el comando **Contar Filas** y asignar el valor a una variable, esta variable es la que se debe pasar al final del rango, ej: **A1:F{filas}**, esto en el caso de que la cantidad de datos en el Excel sea variable, para no estar modificando el comando cada vez que la cantidad de filas cambia, se debe utilizar esta buena práctica.

Cerrar Excel

Este comando cerrará todos los Excel abiertos, independientemente de si fueron abiertos por Rocketbot o no, ya que mata la aplicación de Excel.

Si se requiere cerrar solo el archivo abierto pro Rocketbot, podemos utilizar el comando Cerrar XLSX del módulo Advanced Excel.

Buenas prácticas

Una vez que dejemos de utilizar el Excel debemos cerrarlo, sino el proceso quedará abierto y al acumular demasiados podría afectar en algo el funcionamiento correcto del bot.

Logs en Rocketbot

Los logs se encuentran en la ruta:
Rocketbot/logs/YYMMDD/logfile_YYMMDD_HHMMSS.log

Estructura de Logs de Rocketbot

- Fecha: formato yyyy-mm-dd
- Hora: formato H:M:S.f
- mod/core: ej: Rocketbot, script, system
- Tipo de log: ej: INFO, ERROR, EXCEPTION
- Comando/módulo: ej: clcikWeb, for, alert, module – {module_name: “System++”}

Ejemplo:

Visualizar errores en Rocketbot

En el Studio, cuando un comando falle, se mostrará en color rojo y además indicará el texto del error, también podemos complementar la información revisando la consola de Rocketbot, la cual nos irá mostrando la información en tiempo real, o también revisando el archivo generado en la carpeta logs.

Si obtenemos un error en algún módulo, en la consola podremos ver la línea exacta donde ocurrió.

Ejemplo:

Recomendaciones para generar Logs.

Revisar en el momento de desarrollo todos los errores posibles ya que en el Studio la interfaz muestra todos los errores en forma gráfica. En producción no existe la interfaz de desarrollo por lo que no se muestra ningún error en forma gráfica, por lo tanto se recomienda programar acciones para reportar anomalías en las ejecuciones.

Se recomienda incluir información como Fecha, Hora, BOT, Acción y Resultado de la acción, que sean compatibles con las buenas prácticas de monitoreo y supervisión a través de visualizadores externos como Power BI, Kibana, etc. a través de correo electrónico, mensajes de texto, etc.

Robots en Paralelo

El siguiente cuadro grafica, cuándo y en qué condiciones se pueden ejecutar robots en paralelo.

Tipo de aplicación a automatizar	Posibilidad de implementar Bots en paralelo	Requisitos	Solución Alternativa
Sistemas WEB internos	**SI**	El sistema debe permitir correr múltiples sesiones por equipo	
Sistemas WEB externos	**SI**		
Bases de datos	**SI**		
APIs	**SI**		
Correo	**SI**		
SAP con GUI Scripting	**SI**	Habilitar en SAP GUI Scripting y una sesión por Bot	
Aplicación Java Cliente Servidor	**Probable**	-La aplicación debe permitir correr múltiples sesiones en la misma máquina. -Cada BOT requiere tener una cuenta de acceso. -La aplicación cliente debe ser una aplicación Java(basada en swing y awt) que se pueda grabar en Java Recorder	
Aplicación Windows Cliente Servidor	**SI**	-La aplicación cliente debe permitir correr múltiples sesiones en la misma máquina. -Cada BOT requiere tener una cuenta de acceso. -La aplicación cliente debe ser una aplicación Windows(basada en C# o .net)que se pueda grabar con Desktop Recorder Rocketbot	
Microsoft Word	**SI**	Solo si los BOTs trabajan sobre Archivos Word diferentes	
Microsoft Excel	**Si**	Solo si los BOTs trabajan sobre Archivos Excel diferentes	

Aplicación que obligue al uso de tratamiento de imágenes(OCR) o virtualización para su automatización	**No**	No se puede automatizar en paralelo sobre una misma máquina	Usar VMs sobre un mismo Servidor o equipos físicos. Con licencias individuales Onpremise S(Si requiere usar Xperience).

Escritorio Remoto

No se recomienda trabajar desde el escritorio remoto a menos que sea necesario, por ejemplo si es una web y podemos acceder desde fuera, entonces lo óptimo es realizar el bot desde nuestros computadores, si en cambio el sistema que queremos automatizar está en las dependencias del cliente, entonces tendremos que conectarnos.

Al trabajar desde un Escritorio Remoto podemos presentar lentitud en las acciones, si a esto sumamos conexión a través de alguna VPN y una velocidad de internet más lenta, entonces la lentitud puede aumentar. Esto afecta al tiempo de desarrollo de un bot, ya que al intentar mover un comando a otra posición, el delay de esta acción puede tardar.

Buenas prácticas

Si presentamos lentitud en las acciones, se recomienda primero que todo, definir bien el proceso para no estar modificando cada vez y tener que arrastrar los comandos desde el final hacia la posición deseada, también dividir aún más el flujo para trabajar con mini bots que contengan una cantidad no mayor a 40 instrucciones.

Para arrastrar comandos y empezar a subirlos, se debe posicionar entre el término del menú superior y el inicio del Dashboard (marcado en el recuadro rojo de la imagen de abajo)

Al dejar el comando en esa posición, sin soltar el mouse, empezará a hacer scroll y nuestra instrucción irá subiendo.

Y en el caso contrario, si queremos bajar una instrucción, el scroll lo realizará posicionando el comando entre el final del dashboard y el inicio del footer

---

TIPS COMANDOS ROCKETBOT

FECHAS

MODIFICAR FORMATO DE UNA FECHA CON PYTHON

Variable: fecha: 2019-04-01

Comando "Asignar Variable":

datetime.strptime("{fecha}", '%Y-%m-%d').strftime('%d.%m.%Y')

Resultado: 01.04.2019

MODIFICAR FORMATO DE UNA FECHA CON JAVASCRIPT

Variable: fecha

Comando "Ejecuta JS":

```
fecha = "06/12/2018"
fecha = fecha.split("/").reverse().join("");
return fecha;
```

FECHA DE HOY CON FORMATO

Comando “Asignar Variable”

datetime.now().strftime("%d/%m/%Y %H:%M:%S")

Obtener el día actual (número)

datetime.today().weekday()

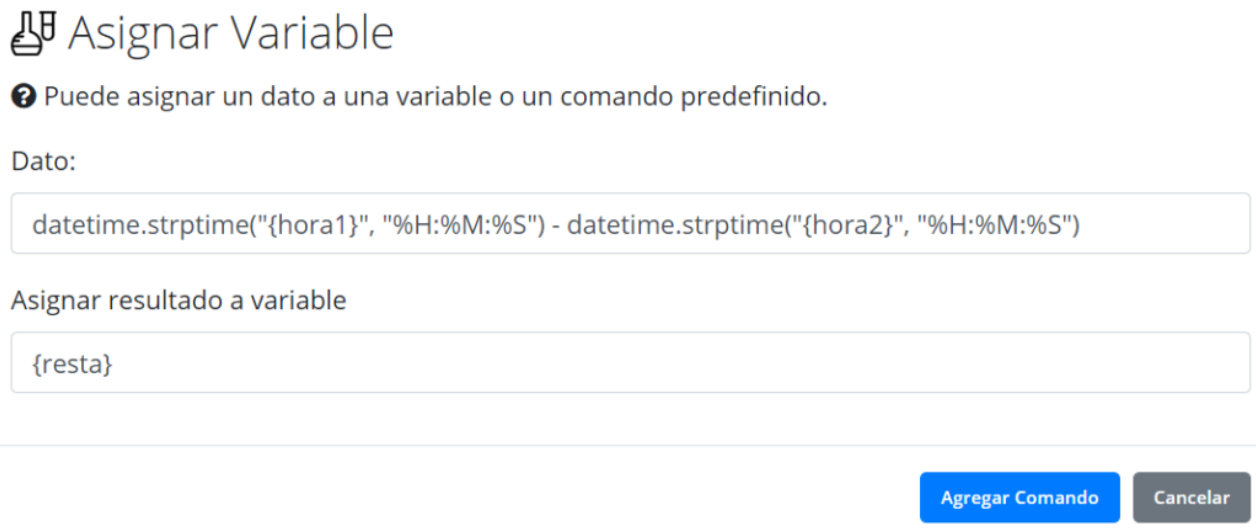
Lunes = 0
Martes = 1
Miércoles = 2

Jueves = 3
Viernes = 4
Sábado = 5
Domingo =6

RESTAR DOS HORAS

Comando "Asignar Variable":

```
datetime.strptime("{hora1}", "%H:%M:%S") - datetime.strptime("{hora2}", "%H:%M:%S")
```



#		Nombre de variable	Datos de variable
1		hora1	10:30:50
2		hora2	09:20:30
3		resta	1:10:20

ÚLTIMO DÍA DEL MES CON SCRIPT PYTHON

Variables:

Archivo last_day.py

Con GetVar estamos trayendo la información de las variables "mes" y "anio" que tenemos declaradas en Rocketbot.

Con SetVar estamos seteando a la variable "resultado" que tenemos declarada en Rocketbot, el valos que nos entrega calendar.monthrange.

Resultado:

DÍA LABORAL ANTERIOR

Comando "Asignar Variable":

```
datetime.today() - timedelta(days={days_delta}[datetime.today().weekday()])
```

variable:
time_delta = [2,3,1,1,1,1,1]

OMITIR ALERT DEL NAVEGADOR POR JS

Comando "Ejecuta JS":

alert = function(a){console.log(a)}

opción 2:
alert = console.log

SEPARAR LETRAS Y NÚMEROS DE VARIABLE

Variables de ejemplo

var = 1234567MANDATO
var_num -> donde se guardarán los números

var_letras -> donde se guardaran las letras

Validar si es dígito:

Validar si es letra:

Resultado:

DEJAR SOLO EL NÚMERO

Variables:

Comando "Asignar variable":

```
"".join([ b for b in "{valor}"  if b.isdigit() ])
```

Resultado:

LEER UN ARCHIVO .TXT Y ASIGNARLO A UNA VARIABLE

Comando "Asignar Variable":

```
open("C:/Users/User/Desktop/archivo.txt").read()
```

Resultado:

ELIMINAR ACENTOS

Tenemos una variable con acento (var), en este caso pasaremos el resultado a var2 para ver el cambio, pero se podría pasar directamente a la misma

variable.

Variables:

Comando "Asignar Variable":

unicodedata.normalize('NFKD', "{var}").encode("ascii", "ignore").decode()

Resultado:

OMITIR VENTANA DE CONFIRMACIÓN DEL NAVEGADOR

Comando "Ejecuta JS":

window.confirm = function () { return true }

O con el módulo: AcceptAlert

MODIFICAR DATOS DE UN ARRAY

Variable con Array:

Modificar posición 1 ("Mantenedores") por la palabra "nuevo_valor"

Comando "Asignar Variable":

> **Se indica que primero tome desde la posición 0 a la 1 (se excluye el valor de la posición 1, 'Mantenedores'), luego que agregue "nuevo_valor" y que continúe desde la posición 2 en adelante y se asigna a la variable "dato" .**

Resultado:

CREAR ARRAY Y AÑADIR VARIABLES

Crear variable con []

Comando "Asignar Variable":

{array}.__add__(["{dato1}","{dato2}"])

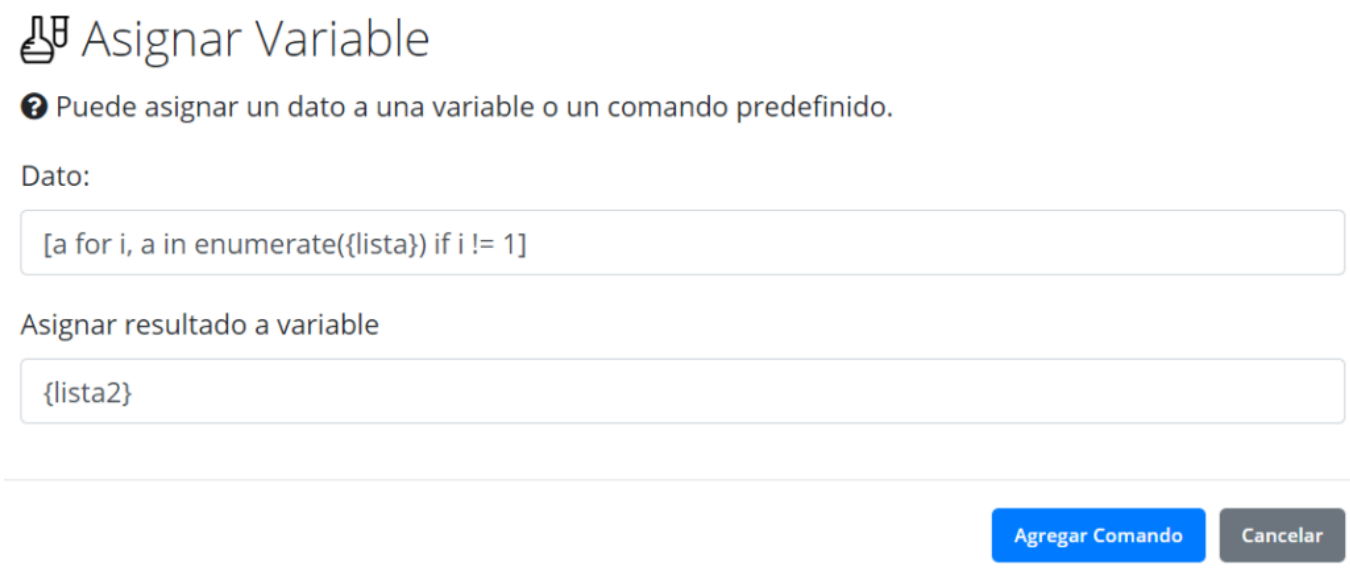
Resultado:

Otra forma:{array} + ["{dato1}","{dato2}"]

ELIMINAR DATOS DE ARRAY

Comando Asignar Variable

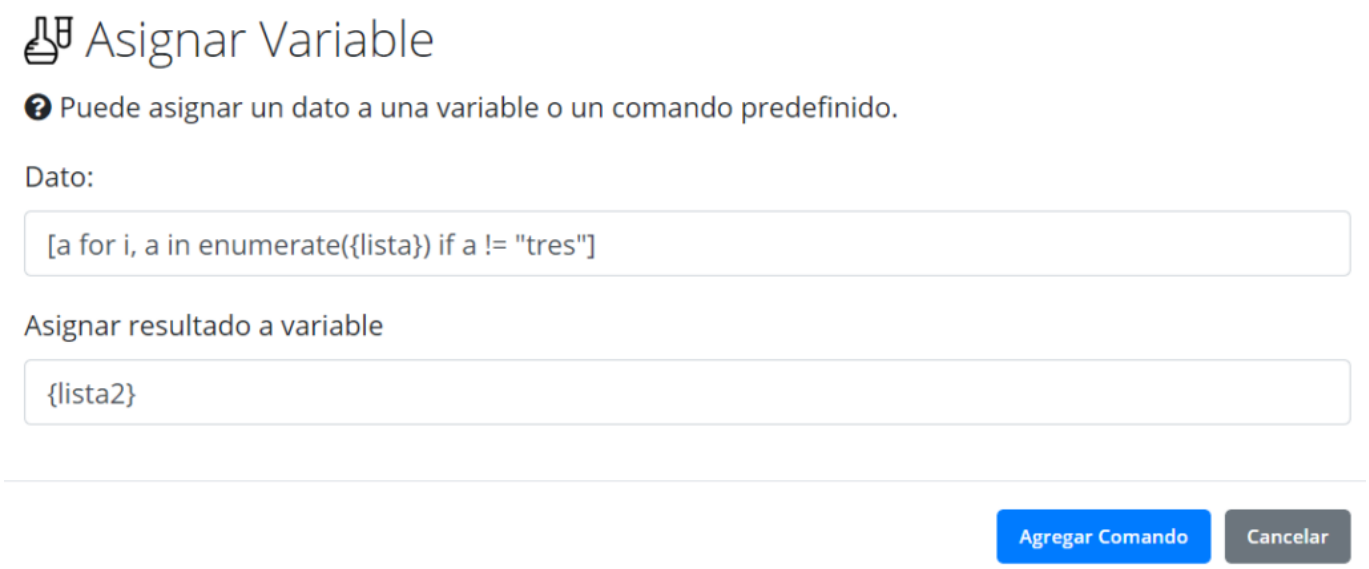
***Eliminar datos de un array indicando la posición:**



Resultado:

#		Nombre de variable	Datos de variable
1		lista	['uno','dos','tres']
2		lista2	['uno', 'tres']

***Eliminar datos de un array indicando el valor:**



Resultado:

#		Nombre de variable	Datos de variable
1		lista	['uno','dos','tres']
2		lista2	['uno', 'dos']

ABRIR “EJECUTAR” DE WINDOWS

Comando "Enviar Tecla":

^{ESC} Ejecutar {ENTER}

Resultado:

O con el módulo: Enviar Teclas Predefinidas

SETEAR UN ATRIBUTO A UN ELEMENTO

Comando "Ejecuta JS":

document.getElementsByClassName("r4nke ")[10].setAttribute("tag","prueba")

DAR CLICK A UN ELEMENTO POR SU QUERYSELECTOR

Comando "Ejecuta JS":

document.querySelectorAll('[data-date="20190529"]')[0].click()

ó

document.querySelector('[data-date="20190529"]').click()

MARCAR UN CHECKBOX POR SU ÍNDICE

Comando "Ejecuta JS":

$("input[type=checkbox]").eq(0).prop("checked",true);

MARCARLOS TODOS:

$("input[type=checkbox]").prop("checked",true);

DESMARCARLOS TODOS:

$("input[type=checkbox]").prop("checked",false);

DAR CLICK A TODOS (MARCA Y DESMARCA):

$(":checkbox").click();

DAR CLICK AL ÚLTIMO:

$(":checkbox").last().click();

SABER SI UN CHECKBOX ESTÁ MARCADO O NO

Inspeccionar el checkbox:

Obtener el selector:

Click derecho -> copiar -> copiar selector

Comando "Ejecuta JS":

return document.querySelector('SELECTOR_COPIADO').cheched

Ejemplo:

> **Retornar el resultado a una variable, en este caso “checkbox”, al ejecutarlo obtendremos un True o False, dependiendo de si está marcado o no**

Para decodificar (sacar la b')

Resultado:

PARA MARCAR UN CHECK DEPENDIENDO DEL VALOR

Ejemplo:

Necesitamos marcar el checkbox si la fecha es igual a "2019-07-02"

Comando "Ejecuta JS":

$('td:contains(2019-07-02)').last().parent().find('input').prop('checked',true).change()

Resultado:

CLICK EN UN BOTÓN SUBMIT

Comando "Ejecuta JS":

$(":submit").click()

GUARDAR EN UNA VARIABLE EL STATUS (RESPUESTA) DEL COMANDO EJECUTADO

Comando "Asignar Variable":

%rocketbot_last_status%

Ejemplo:

Abrir navegador y obtener la respuesta en la variable {res}

Resultado:

CREAR UNA CARPETA DESDE ROCKET

Comando "Asignar Variable":

os.mkdir("C:/el/path")

COPIAR VALOR DE UNA VARIABLE AL PORTAPAPELES

Comando “Asignar Variable”

Windows:

os.system("echo {variable} | clip")

MAC:

os.system("echo {variable} | pbcopy")

o con el módulo para Windows: Portapapeles

OBTENER CURRENT WORKING DIRECTORY

Comando “Asignar Variable”:

os.getcwd()

OBTENER NOMBRE DE ARCHIVO DE UN PATH

Variables:

Comando "Asignar Variable":

os.path.basename(r"{path}")

Resultado:

COMANDO DETENER

El comando Detener, si está en un hijo, detendrá ese Robot y continuará con los pasos del Padre.

Si está dentro de un if detiene el if y todo el robot.

CUANDO HAY EVENTOS CHANGE Y NECESITAMOS ACTIVAR EL CAMBIO DEL VALOR

Ejemplo:

***Por ejemplo esta fecha, al cambiar el value con JS lo modificaba pero no "activaba" la búsqueda, para eso, podemos realizar lo siguiente con JQuery y utilizar change():**

Comando "Ejecuta JS":

```
$('input[class="form-control ng-pristine ng-untouched ng-valid ng-isolate-scope ng-valid-required ng-valid-date"]').val('{fecha}').change()
```

REQUEST API

Si se realiza un GET en un script Python y arroja este error:

('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))

En el Script se debe agregar el "User-Agent":

```
headers = {
   'User-Agent': 'My User Agent 1.0',
}
data = requests.get(URL_, headers=headers)
data = data.json()
```

MINIMIZAR TODO PARA MOSTRAR EL ESCRITORIO (WINDOWS)

Comando "Enviar Tecla":

{VK_LWIN down}D{VK_LWIN up}

o con el módulo: Enviar Teclas Predefinidas

EN EXCEL AL OBTENER CELDA CON NÚMEROS LOS TRAE COMO FLOAT

Validar si termina en .0 para hacer un "replace"

Validamos si la respuesta en {var2} es True, y si es reemplazamos el .0 por nada

Secuencia:

VALIDAR CONEXIÓN A INTERNET

Comando "Ejecuta JS":

return navigator.onLine

Devuelve True o False.

REALIZAR SCROLL

Comando "Ejecuta JS"

window.scrollTo(0,500);

***500 px horizontalmente**

o con el módulo: WebPro

OBTENER EL TOTAL DE FILAS DE UNA TABLA

Comando "Ejecuta JS":

document.getElementsByClassName('tabla_indicadores')[4].getElementsByTagName("tr").length;

***Donde tabla_indicadores es la clase de la tabla de la cual queremos obtener las filas.**

SELECCIONAR UNA OPCIÓN DE UN SELECT EN BASE A UNA VARIABLE

Comando "Ejecuta JS":

$(‘span:contains({var})’).click()

ENVIAR EMAIL CON ACENTOS DESDE UN ROBOT HIJO

Para los tildes y eñes, deberíamos colocar el texto según estas opciones:

á -> á

é -> é

í -> í

ó -> ó

ú -> ú

ñ -> ñ

Ejemplo con el módulo del Gmail:

Correo:

ELIMINAR LOS NONE DE UNA LISTA

En Script Python:

```
test_list = [1, None, 4, None, None, 5, 8, None]
# printing original list
print ("The original list is : " + str(test_list))
# to remove None values in list
res = []
for val in test_list:
    if val != None :
        res.append(val)
```

Modificar texto para que empiece con letra mayúscula:

Comando: "Asignar Variable"

Resultado:

TIPS VIRTUALIZACIÓN

Estos son algunos casos donde los click en imagen por ejemplo no funcionen correctamente, por lo tanto se debe revisar resolución o ejecutar Rocketbot como administrador.

1.- Si no realiza los click en las imágenes.

Instalar lo siguiente:

.NET Framework: https://dotnet.microsoft.com/download/dotnet-framework

Visual C++:
https://support.microsoft.com/es-es/help/2977003/the-latest-supported-visual-c-downloads

Revisar:

Si la configuración de pantalla está aumentada, dejarla al 100% y probar

> **Importante**: Si se realizan cambios de escala, cerrar sesión y volver a abrir para que los cambios se guarden.

2.- Si realiza click en ciertas imágenes pero no en todas

Por ejemplo realiza click en elementos del escritorio o barra de tareas pero no en una aplicación de escritorio interna, además al intentar abrir la aplicación con Rocketbot aparece lo siguiente:

- Ejecutar Rocketbot como administrador y probar.

3.- Si realiza click en otra parte

Especificar alguna referencia, por ejemplo si dos íconos iguales en las pestañas del navegador y en este caso queremos hacer click en la segunda, debemos especificar una referencia, sino hará click en la primera coincidencia.
Entonces para especificar nuestro click identificamos una referencia cercana: *En este caso vemos que la palabra “Ayuda” está cerca de la pestaña a la cual queremos hacerle click, por lo que la dejaremos como Referencia, y el ícono lo especificamos como foco (donde hará el click), con esto ya

realizará el click donde queremos.

> Si sigue realizando el click en otra parte, y se está trabajando desde un escritorio remoto, debemos cambiar la resolución antes de ingresar al servidor.

-Por ejemplo, el foco (donde hacer el click) es el ícono de Rocketbot, pero el click lo realiza en el punto rojo:

Configuración:

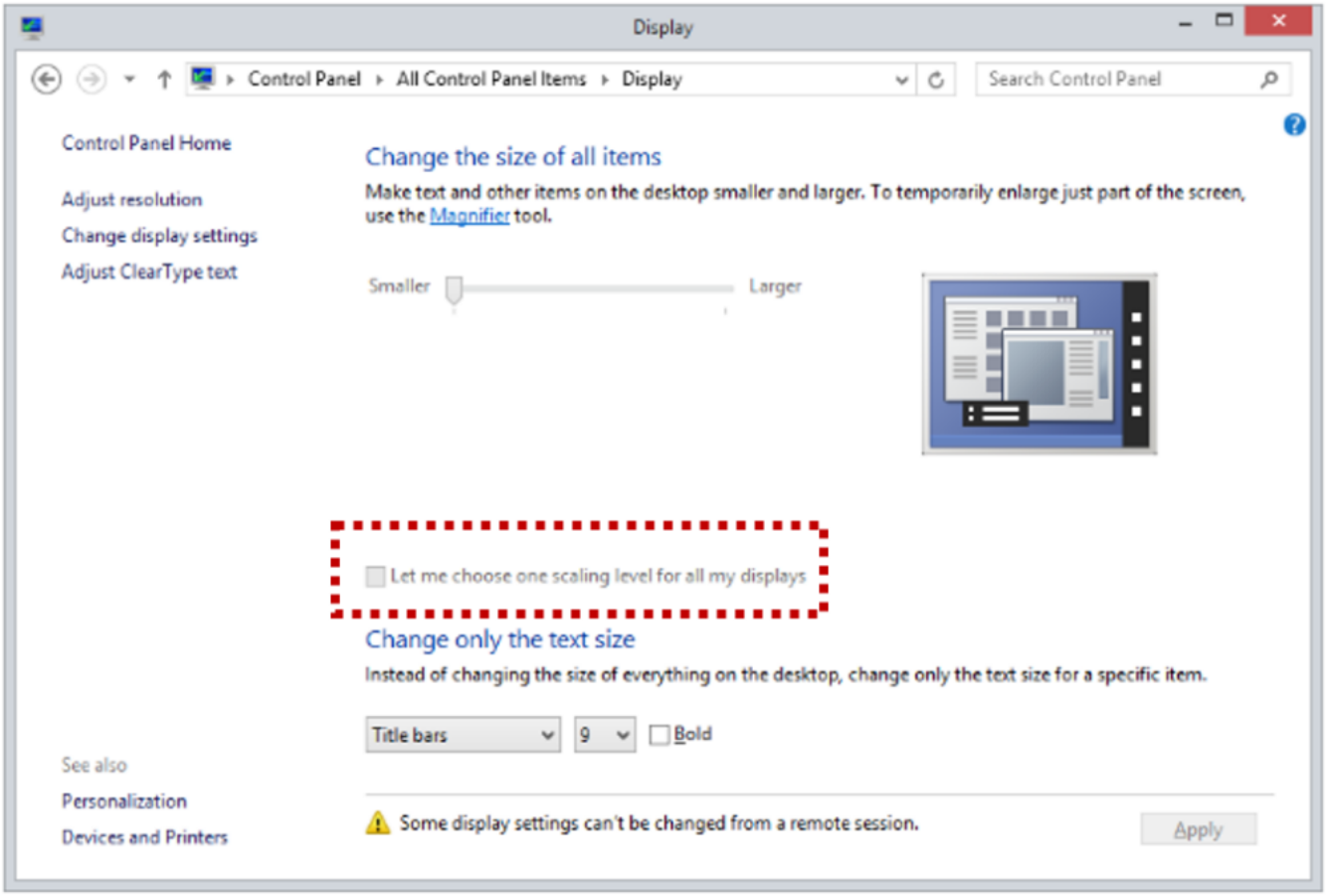
4.- Si sigue sin encontrar las imagenes y es un Windows Enterprise (win 7 o win 10), revisar si está la siguiente .dll sino, agregarla:

DLL: api-ms-win-downlevel-shlwapi-l1-1-0.dll

5.-VIRTUALIZACIÓN EN WINDOWS SERVER 2012 MEDIANTE RDP

En caso que las funciones de virtualización para “click en imagen” o “esperar por imagen” no funcionara correctamente revisar:

Que no esté habilitada la opción de cambiar el nivel de escala:



Adicional a esto verificar que este instalada las características dentro de “Interfaz de usuario e Infraestructura” en especial y en especial “Experiencia de escritorio”:

En MAC

Permiso denegado al hacer OCR

1. Abrir el terminal e ir a la ruta indicada en el error, ej:

   cd /Users/usuario/Desktop/Rocketbot/drivers/mac/tesseract

2. Escribir el siguiente comando para visualizar los permisos:

   ls -l

3. Veremos que en este caso no tiene permisos de ejecución:

```
marcerotten@Macbook-Pro: ~/Desktop/Rocketbot/drivers/mac/tesseract
> ls -l
total 12160
-rw-r--r--@ 1 marcerotten  staff  6223280 May 24 23:27 tesseract
```

4. Le damos los permisos con el siguiente comando:

ejecución para todos los usuarios: chmod 777 tesseract
ejecución solo para el owner: chmod 744 tesseract

```
marcerotten@Macbook-Pro: ~/Desktop/Rocketbot/drivers/mac/tesseract
> chmod 744 tesseract
```

5. Volvemos a ejecutar un ls -l para revisar los cambios:

```
marcerotten@Macbook-Pro: ~/Desktop/Rocketbot/drivers/mac/tesseract
> ls -l
total 12160
-rwxr--r--@ 1 marcerotten  staff  6223280 May 24 23:27 tesseract
```

6. Ahora ejecutamos nuevamente el comando de OCR