



MANUAL DE REVISIÓN PRÁCTICAS DE ROCKEBOT DE RxP

Código: [MNL-A-FI_CT](#)

Responsable: David Sánchez

Ejecutor: CompuSoluciones

Versión marzo 2025

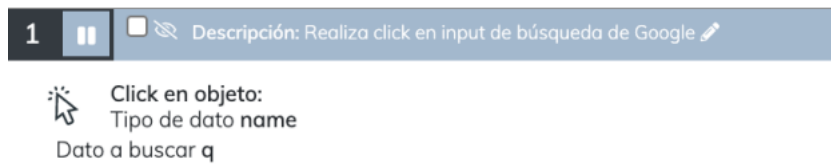
Objetivo del manual

Mostrar los pasos a seguir para el correcto funcionamiento de RockeBot.

1. Generalidades Robot

1.1 Descripción de 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.

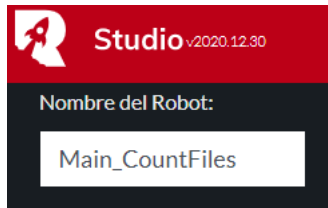


Nota: Se hace una excepción de descripción en el comando cuando se tienen instrucciones que refieren a: Cierre de Navegador, Alertas.

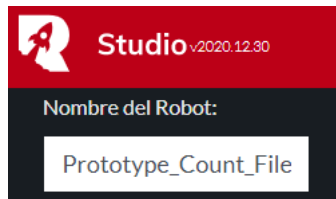
Nota: Al menos el 80% de todas las instrucciones ejecutadas por el robot tienen que estar documentadas.

1.2 Nombre del Robot

Para el nombramiento de los robots implementados es una buena práctica que el robot principal inicie por ejemplo con: [Main_LoQueQuieras](#).



No se deben utilizar **espacios**, **ñ**, ni **caracteres especiales** en el nombre del robot, si el nombre fuera compuesto por varias palabras puede utilizarse el siguiente formato: **Nombre_Robot**.



1.3 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: **nombre_variable**, **nombreVariable**, **Nombre_Variable** o **NombreVariable**.

 A screenshot of the 'Agregar' (Add) dialog box. It has a dark grey header with the title 'Agregar'. Below the header, there are three sections: 'Nombre:' with a text box containing 'Nombre_Variable'; 'Tipo:' with a dropdown menu set to 'General'; and 'Datos:' with a large empty text area. At the bottom right, there are two buttons: 'Agregar' (orange) and 'Cancelar' (grey).

1.4 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:

 A screenshot of the 'Limpiar variable(s)' command block. The block has a light blue header with the title 'Limpiar variable(s)' and icons for 'Error Handling' and 'System++'. Below the header, there is a light blue instruction box that says 'Ingrese las variables a limpiar separadas por coma'. Underneath, there is a section labeled 'Variables' with a text box containing 'var1,var2,var3,var4'. At the bottom, there is a small note: 'Nombres de variables sin () y separadas por coma'.

1.5 Variables Estáticas

Es buena práctica que las variables comiencen con un guion bajo, por ejemplo: **_VariableEstatica**, **_variabledeejemplo**, **_Variable_De_Ejemplo**, con esto ayuda a que estos valores estarán fijos y no se deben limpiar.

1.6 Reglas de Variables

String

Cuando se requiere trabajar con variables de tipo **String** en Rocketbot, se debe pasar la comilla doble, ejemplo.

- Se tiene una variable llamada **{texto}** la cual contiene lo siguiente:

#	Nombre	Datos
1	texto	Esto es un texto

Si se quiere por ejemplo modificar parte del texto, se puede utilizar el método de **String** llamado **replace()**, en este caso solo se pasa la variable, esto arrojará un error de sintaxis:

Se debe indicar que la variable **{texto}** la trabaje como un String, y para eso pasarla entre comillas dobles **"{texto}"**.

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

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

Nombre :

texto

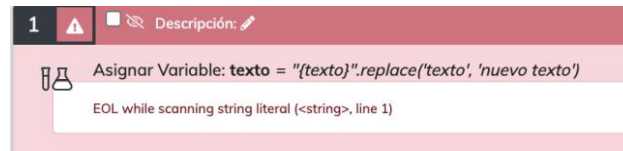
Tipo:

General

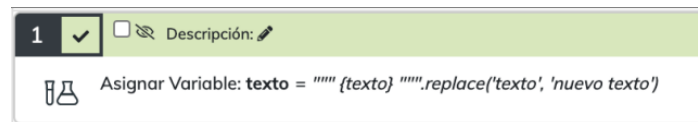
Datos :

Esto es un nuevo texto con saltos de línea

Si se vuelve a ejecutar el comando anterior, arrojará un error:

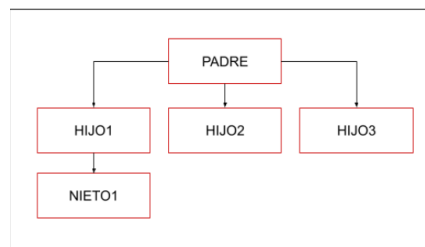


Al pasarla a triple comilla doble y espacios, ejecutará el método sin problema:



1.7 Variables Heredades Padre-Hijo

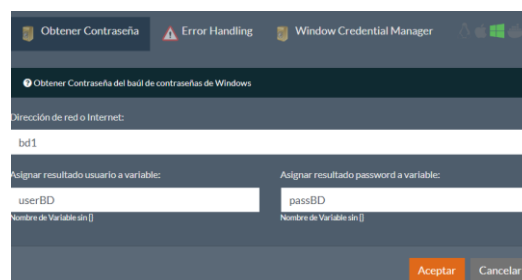
Diagrama de flujo



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**.

1.8 Credenciales

Como buena práctica es importante utilizar un baúl de credenciales para obtener los datos y no dejarlos fijos. Si se tiene Mac o Linux, se puede utilizar el siguiente módulo **Credentials**, si se está en Windows, se debe usar el siguiente: **Windows Credential Manager**. Se puede encontrar más información acerca de cómo realizarlo, en la documentación de los manuales enlazados arriba.



1.9 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 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:

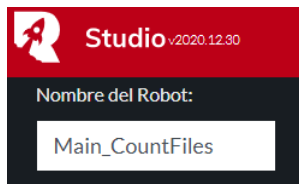
Crear un archivo Excel y validar su correcta creación, si está bien, entonces se escribe en el archivo, si no, se deja un mensaje en el log y se detiene el robot.



1.10 Estándar de Rutas

Como buena práctica las rutas para acceder a recursos extras que las utilicen los robots padres, hijos y nietos se deben declarar esas variables en el robot **Main**, para así tener un mejor acceso y control de estas, se busca evitar estar buscando dentro de todos los robots, con esto, una vez que se modifica en el **Main**, en todos los demás robots esa variable estará actualizada.

Nota: Si la variable solo la utiliza el hijo o nieto, se puede declarar en dicho sub-robot.

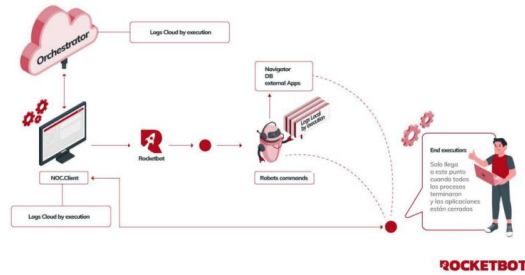


Variables		Importar	Exportar	Agregar
path_main_excel_send	/Users/josanchez/Desktop/filesC2/cert2_12092022.xlsx			
path_log	/Users/josanchez/Desktop/filesC2/path_log_c2.txt			
download_ftp	/Users/josanchez/Desktop/filesC2			
path_legacy_log	C:/Users/josanchez/Desktop/filesC2/legacy_log.txt			

1.11 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 importante siempre cerrar todo lo que se abre con **Rocketbot**, ya que esto 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.



2. Web

2.1 Abrir Navegador

La interfaz 'Abrir navegador' permite configurar la apertura de un navegador. Incluye pestañas para 'General' y 'Extra'. En la pestaña 'General', se puede seleccionar el navegador (actualmente 'chrome'), ingresar la URL de servidor (con un placeholder '[url]') y elegir el modo de apertura: 'Síncronico' (seleccionado) o 'Asíncronico'. Se proporcionan explicaciones para ambos modos. Al final, hay botones para 'Aceptar' y 'Cancelar'.

Opciones:

- **Seleccionar 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

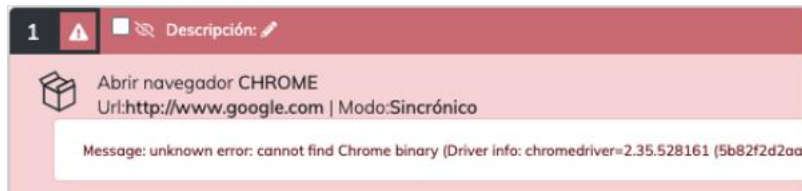
Es óptimo agregar la **URL** en una variable, pensando en escalabilidad, si en algún momento se requiere que la variable sea dinámica o si necesita cambiar, lo ideal es que **Rocketbot** lea algún archivo de configuración y así poder modificar el valor a la variable para no estar abriendo y modificando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

2.2 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>

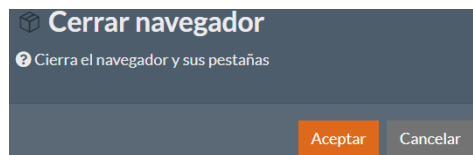
Mozilla: <https://github.com/mozilla/geckodriver/releases/tag/v0.29.0>

Descargar y descomprimir el driver, copiar el archivo en la siguiente ruta:
[Rocketbot/drivers/{sistema operativo}/{browser}](#) .

Ej: Rocketbot/drivers/win/Chrome.

2.3 Cerrar Navegador

Permite cerrar el navegador abierto con [Rocketbot](#), cierra todo incluyendo las pestañas abiertas.



Buenas Prácticas

Siempre que se hace un robot de tipo [Web](#), y ya no se requiera utilizar más el navegador, se debe 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.

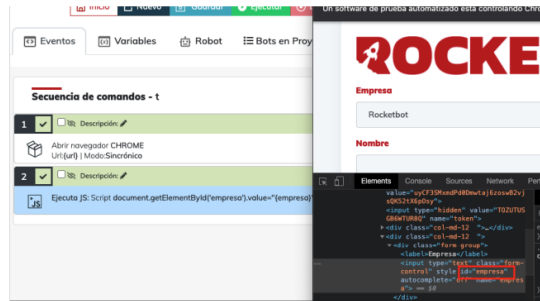
Nota: Todo lo que se abre con [Rocketbot](#), es óptimo que se cierre al final del proceso o cuando ya no se utilice más, ej: [navegador](#), [conexión a BD](#), [conexión a email](#), [Excel](#), etc.

2.4 JavaScript

Ejecuta JS

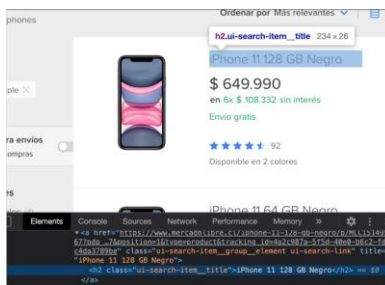
Se puede ejecutar código [JS](#) directamente desde [Rocketbot](#), a través de la miniconsola. Por ejemplo, si se quiere enviar un valor a un input, se puede utilizar un identificador de este (en este ejemplo el identificador es un ID llamado empresa) y enviarlo de esta forma:





Si se necesita obtener un resultado desde la página hacia [Rocketbot](#), se debe utilizar el [return](#) y asociar la respuesta a una variable.

Por ejemplo, si se requiere obtener el Nombre del artículo, al inspeccionar se ve que tiene una clase, la cual se utilizara.



De esta forma y con [innerText](#) se está obteniendo el texto y asignándolo a la variable `{titulo}`.



Obtendremos el texto de esta forma:

titulo	b'iPhone 11 128 GB Negro'
--------	---------------------------

Importante: Para utilizar esta variable como [String](#) se debe utilizar `{variable}.decode()` o `{variable}.decode('latin-1')` en el comando [Asignar Variable](#).

Asignar Variable

Puede asignar un dato a una variable o un comando predefinido.

Dato:

Asignar resultado a variable

Aceptar
 Cancelar

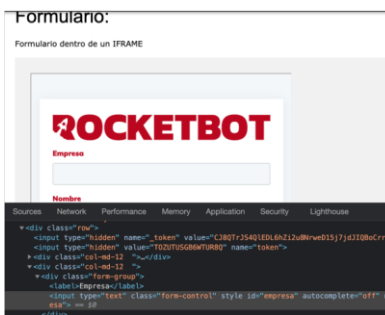
Quedará de esta forma y ya se puede utilizar la información correctamente:

titulo	iPhone 11 128 GB Negro
--------	------------------------

2.5 Cambiar a IFRAME

Mueve el focus al **IFRAME** seleccionado. Ahora todos los comandos **JS** o acceso a objetos web tomará como referencia el **IFRAME** seleccionado.

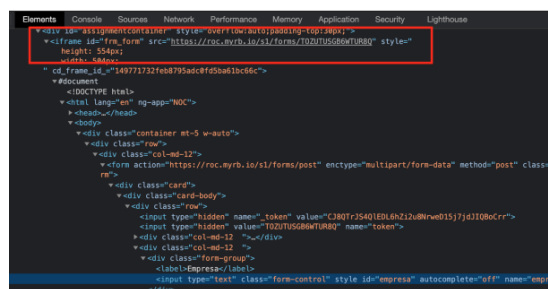
En este ejemplo se requiere realizar un click en el input empresa:



Pero arroja el siguiente error:



Indica que no encuentra el elemento, uno de los errores más comunes es que se encuentre dentro de un iframe, por lo que se debe buscar si el elemento está dentro de alguna etiqueta **<iframe>**.



Revisar que el input está dentro de un **IFRAME**, por lo que se debe utilizar el comando indicando algún identificador de este, en este caso se usará el **ID**.

Cambiar a IFRAME

● Entra a un iframe y deja su contenido por defecto

Dato a buscar

Tipo de dato

id

Aceptar

Cancelar

Este comando debe ir antes de que se interactúe con el elemento, al ejecutarlo se dejará en foco este **IFRAME** y luego al ejecutar el click ya funcionará correctamente.



2.6 Cambiar a Contenido Por Defecto

Mover el focus al cuerpo del documento. Ahora todos los comandos **JS** o acceso a objetos web tomará como referencia el cuerpo de la página.

Siguiendo con el ejemplo anterior, si ahora se requiere volver el foco a la página por defecto y salir del **IFRAME**, se debe utilizar el comando.

Cambiar a Contenido por defecto

● Deja el cuerpo de la página como contenido por defecto

Aceptar

Cancelar

3. Scripts

3.1 Ejecutar archivo Python

Archivo python: se puede utilizar las variables de **Rocketbot** llamándolas entre llaves, tal cual se utilizan en los robots. Para devolver un valor desde el script hacia el robot, se debe utilizar **SetVar()** donde primero se indica entre comillas el nombre de la variable en **Rocketbot** y luego el valor que se le quiere asignar.

```
from datetime import timedelta

fecha_rocket = "{fecha}"

fecha_nueva = datetime.datetime.strptime(fecha_rocket, '%d-%m-%Y')
fecha_nueva = fecha_nueva - timedelta(days=1)

SetVar('result', fecha_nueva.strftime('%d-%m-%Y'))
```

Buenas Prácticas

Se debe utilizar **SetVar()** donde primero se indica entre comillas el nombre de la variable en **Rocketbot** y luego el valor que se le quiere asignar.

3.2 Execute Python

Con este comando se puede ejecutar código python con las librerías básicas incluidas.

Ejemplo: Se utilizará la librería random para obtener un número aleatorio y guardarlo en una variable a través de **SetVar()**.



Buenas Prácticas

Se debe utilizar **SetVar()** donde primero se indica entre comillas el nombre de la variable en **Rocketbot** y luego el valor que se le quiere asignar.

Nota: Al menos las líneas de código importantes deben estar comentadas con “#” Definiendo el funcionamiento de lo que hace la misma. Esta nota aplica para las secciones “3.1 Ejecutar Archivo Python” y “3.2 Execute Python”.

3.3 Ejecuta otro Script Rocketbot

Buenas Prácticas

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.

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



4. Lógica

4.1 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.

IF

● Ejecuta un evento evaluando un comando lógico en Python

Lógica Python

`"{var}" != "Emisor"`

ej: `{variable} > 8 | {variable} == 1`

Aceptar Cancelar

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:

1 IF : Estado Lógica Python "{var}" != "Emisor"

Res:

Else

1 Escribir Celda: Celda G(cont)="Realizado"

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

1 IF : Estado Lógica Python "{var}" == "Emisor"

Res:

1 Escribir Celda: Celda G(cont)="Realizado"

Else

4.2 While

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

© **While**

● Ejecuta un ciclo mientras se cumpla una condición

Lógica Python

`{variable} > 8 | {variable} == 1`

Aceptar Cancelar

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, si no, se corre el riesgo de generar un bucle infinito.

5. Archivos

5.1 Guardar Texto en Archivos (Logs)

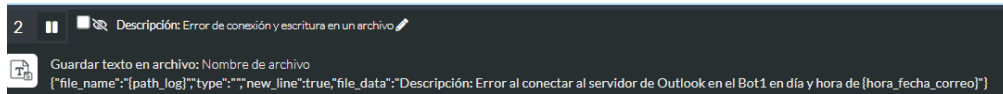
Este comando sirve para pasar información a un archivo, se puede utilizar por ejemplo para crear el propio log e ir almacenando información del robot en caso de que hubiera un error, se usa en los ELSE cuando una validación fue errónea.

- **Ruta del archivo:** Indicar la ruta de nuestro archivo, si no existe se creará.
- **Modo:**
 - **Agregar:** Irá adicionando información al archivo, por ejemplo, en el caso de crear un archivo log.
 - **Sobrescribir:** Irá pisando la información existente en el archivo.
- **Agregar salto de línea:** Seleccionarlo si se quiere incluir un salto de línea antes del texto enviado.
- **Texto:** Indica si es texto fijo o pasar una variable de Rocketbot.

Buenas Prácticas

Para esta ejecución se podría incluir un path log donde se muestre el historial en las partes que ha fallado el robot, ejemplo de uso:

- Descripción: Error al conectar al servidor de **Outlook**(Módulo que se está utilizando) en el Bot1 (Numero de bot o nombre) en día y hora de **{hora_fecha_correo}** (Hora y fecha en que falló).



6. XLSX

Si se trabaja 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 se ve que este se abra, todo pasará por debajo.

6.1 Nuevo XLSX

Permite crear un nuevo archivo **XLSX** en memoria, se puede indicar un Identificador (puede ser un número, letra o palabra) en el caso de que se necesite crear y trabajar con más de un archivo **xlsx** abierto por **Rocketbot**, luego si se necesita ir moviéndose entre uno u otro, se tendrá que realizar a través de su **ID**, también podemos indicar una variable, donde obtendremos un True si se pudo crear un nuevo archivo o False en caso contrario.

A screenshot of the 'Nuevo xlsx' dialog box. It has a title bar 'Nuevo xlsx' and a subtitle 'Crea un archivo XLSX en memoria'. There is a label 'Identificador (vacío = default)' above a text input field. Below that is a label 'Asignar resultado a variable' above another text input field containing '{variable}'. At the bottom right are two buttons: 'Aceptar' and 'Cancelar'.

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:



6.2 Abrir XLSX

Permite abrir un archivo **XLSX** en memoria, se debe 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.

Abrir.xlsx

Lee una hoja de cálculo XLSX en la memoria para trabajar, puede usar el identificador para cambiar entre otros libros de trabajo la próxima vez si abre más libros de trabajo

Nombre del archivo:

Asignar resultado a variable: Id:

Buenas Prácticas

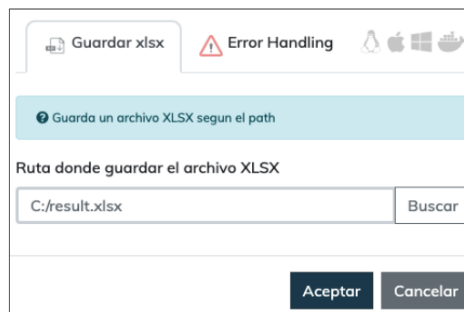
Retornar a una variable para validar si se pudo abrir de forma correcta o no y con eso tomar una decisión a través del comando **IF**.

Ejemplo:



6.3 Guardar XLSX

Permite guardar el archivo **XLSX** que está en foco indicando la ruta y nombre, si estamos trabajando con más de uno, primero se debe dejar activo el que queremos guardar, con el comando **Cambiar de Archivo**.



The dialog box titled 'Guardar xlsx' has a tab labeled 'Error Handling'. Below the title bar, there is a light blue bar with the text 'Guarda un archivo XLSX segun el path'. Underneath, the text 'Ruta donde guardar el archivo XLSX' is followed by a text input field containing 'C:/result.xlsx' and a 'Buscar' button. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

6.4 Cambiar Hoja

Por defecto se trabaja con la primera hoja del archivo, si se necesita trabajar con otra, se debe cambiar de Hoja indicando su nombre.



The dialog box titled 'Cambiar de Hoja' has a subtitle 'Cambia de Hoja por defecto'. Below this, the text 'Nombre de Hoja' is followed by a text input field. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

6.5 Contar Filas

Este comando permite obtener la cantidad total de filas con información del archivo **xlsx**. Por defecto contará las filas de la primera hoja, si se necesita trabajar con otra, primero se debe cambiar con el comando **Cambiar de Hoja**.



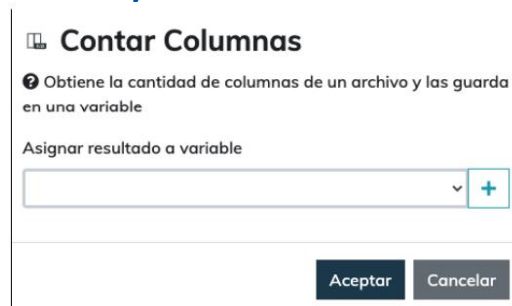
The dialog box titled 'Contar Filas' has a subtitle 'Obtiene la cantidad de filas de un archivo y las guarda en una variable'. Below this, the text 'Asignar resultado a variable' is followed by a dropdown menu and a '+' button. At the bottom right, there are 'Aceptar' and 'Cancelar' buttons.

Buenas Prácticas

Siempre luego de Abrir un archivo **xlsx**, se debe 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.

6.6 Contar Columnas

Este comando permite obtener la cantidad total de columnas con información del archivo [xlsx](#). Por defecto contará las columnas de la primera hoja, si se necesita trabajar con otra, primero se debe cambiar con el comando [Cambiar de Hoja](#).

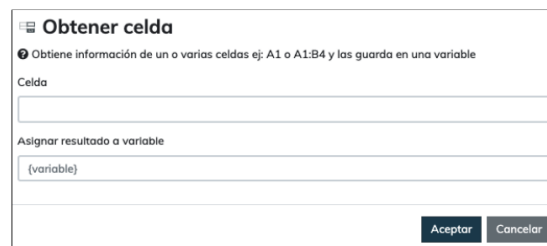


The screenshot shows a dialog box titled "Contar Columnas". It contains a description: "Obtiene la cantidad de columnas de un archivo y las guarda en una variable". Below this is a label "Asignar resultado a variable" followed by a text input field and a small dropdown arrow. To the right of the input field is a blue square button with a white plus sign. At the bottom right are two buttons: "Aceptar" and "Cancelar".

6.7 Obtener Celda

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

Ejemplo:



The screenshot shows a dialog box titled "Obtener celda". It contains a description: "Obtiene información de un o varias celdas ej: A1 o A1:B4 y las guarda en una variable". Below this is a label "Celda" followed by a text input field. Underneath is a label "Asignar resultado a variable" followed by a text input field containing the placeholder "{variable}". At the bottom right are two buttons: "Aceptar" and "Cancelar".

Buenas Prácticas

Cuando se trabaja con un rango (ej: A1:F3) se debe pasar un número fijo al final de este, si no, se debe 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.

6.8 Escribir Celda

Este comando permite escribir un texto en una celda específica o un rango si se define una matriz en el input Texto.



The screenshot shows a dialog box titled "Escribir Celda". It contains a description: "Escribe un dato en la celda especificada en la hoja activa". Below this are two labels: "Celda:" and "Texto:". Under "Celda:" is a text input field. To its right is an equals sign (=). To the right of the equals sign is another text input field under the "Texto:" label. At the bottom right are two buttons: "Aceptar" and "Cancelar".

Ejemplo:

Para escribir en un rango se debe indicar en Celda, la celda de inicio y en Texto lo siguiente: `[["Titulo1", "Titulo2"], ["valor1", "valor2"]]`.



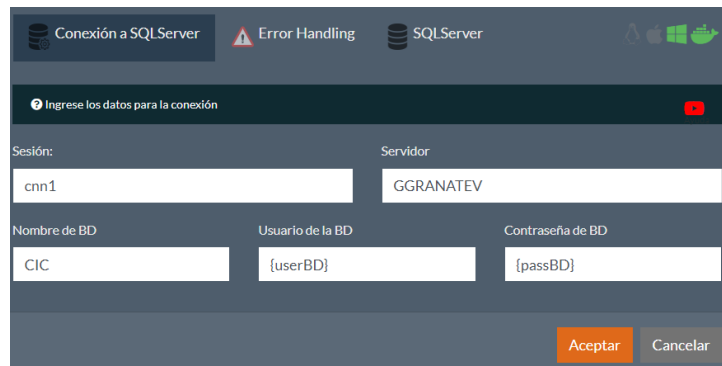
Resultado:

	A	B	C
1	Titulo1	Titulo2	
2	valor1	valor2	
3			

7. SQLServer

7.1 Configurar conexión SQLServer

Para conectarse con **SQLServer**, se debe indicar la **URL** del servidor, nombre de la **BD**, **usuario** y **contraseña**. En el caso de esta conexión a la base de datos no hay donde se asigne un resultado cuando la conexión fue exitosa o no exitosa.

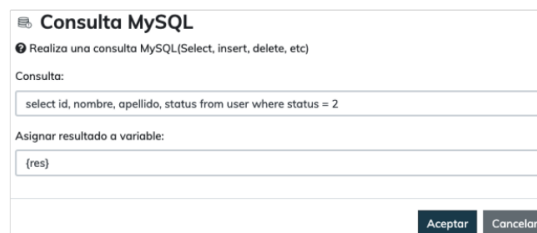


Buenas Prácticas

En este caso en específico como no se puede validar la conexión con una variable es necesario poner un bloque de **Try-Catch** para cuando la conexión no sea exitosa entre en dicho bloque.

7.2 Consulta MySQL

Realiza una consulta **MySQL** (Select, insert, delete, update), el resultado se retorna 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.

Consulta MySQL

Realiza una consulta MySQL(Select, insert, delete, etc)

Consulta:

select id, nombre, apellido from user limit 100;

Asignar resultado a variable:

(res)

Aceptar Cancelar

Otra buena práctica para este módulo es que el resultado de la consulta se compare en un **IF** para verificar que los datos de la consulta si los obtenga la variable



8. Email

8.1 Outlook

Configurar servidor

Configura un servidor de **Outlook** para enviar correos. Al abrir el comando solo viene para llenar una variable que con ella ayudará a validar la conexión.

Conectar a Outlook

Error Handling

OfficeOutlook

Conectar a una instancia de la aplicación de Outlook

Variable donde guardar el resultado:

correo_error

Aceptar Cancelar

Buenas Prácticas

Es una buena práctica el validar que la conexión con el servidor de **Outlook** haya sido exitosa con un **IF**.

Ejemplo:



8.2 Gmail

Configurar servidor

Configura un servidor de **Outlook** para enviar correos. Al abrir el comando viene para llenar una variable que con ella ayudará a validar la conexión, otra variable para el usuario y la contraseña.

The 'Configurar Servidor' dialog box contains the following fields:

- Usuario:** [gmail]
- Contraseña:** [pass]
- SSL Connection:** [checked]
- Asignar resultado a Variable:** [conn]
- Nombre de variable sin []:** []

Buttons: **Aceptar** and **Cancelar**

Buenas Prácticas

Validar si la conexión fue exitosa y en caso de que no con una **IF**.

Ejemplo:



9. Outlook 365

Outlook 365

9.1 Configurar Servidor

Se deben ingresar los siguientes datos:

- **User:** Mail que se quiere configurar
- **Contraseña:** Contraseña del mail a configurar.

- **Asignar resultado a variable:** Nombre de variable donde retornaremos True o False, indicando si se pudo conectar o no.

9.2 Enviar email

Se deben ingresar los siguientes datos:

- **Para:** Mail de destino.
- **Asunto:** Asunto del mensaje a enviar .
- **Mensaje:** Cuerpo del mensaje.
- **Archivo Adjunto:** Ruta del archivo a enviar (opcional).

9.3 Listar todos los emails

Se deben ingresar los siguientes datos:

- **Filtro:** Filtro por el cual se quiere buscar un mail, en este caso se está buscando que el subject sea "Nuevo mail". Si se deja vacío traerá todos los emails.

- **Asignar a variable:** Nombre de la variable donde se retornar el o los id's de los emails. El nombre de la variable debe ir sin llaves {}.

Robot: RocketBot

Lista todos los email Outlook365

Lista todos los email, se puede especificar un filtro

Filtro:

SUBJECT "nuevo mail"

Dejar vacío para traerlos todos

Asignar a variable:

var

Nombre de variable sin {}

Agregar Comando Cancelar

Resultado Variable:

var ['3']

Para utilizar el id, se debe indicar en este caso, lo que se quiere, el primer elemento del array, para esto se utiliza el comando **“Asignar Variable”**:

Asignar Variable

Puede asignar un dato a una variable o un comando predefinido.

Dato:

{var}[0]

Asignar resultado a variable

{var}

Agregar Comando Cancelar

Resultado de Variable:

var 3

9.4 Listar emails no leídos

**A diferencia del comando anterior, este retorna los id's de los emails que no hayan sido leídos. Se deben ingresar los siguientes datos:

- **Filtro:** Filtro por el cual se quiere buscar un mail, en este caso se está buscando que el subject sea **“Test”**. Si se deja vacío traerá todos los emails no leídos.

- **Asignar a variable:** Nombre de la variable donde se retornará el o los id's del email. El nombre de la variable debe ir sin llaves {}.

The screenshot shows the 'Robot PocketBot' window. At the top, a command bar displays 'var' followed by '[3]'. Below this, a light blue bar contains the text 'Lista todos los email, se puede especificar un filtro'. Underneath, there is a 'Filtro:' section with a text input field containing 'SUBJECT "nuevo mail"' and a note 'Dejar vacío para traerlos todos'. The 'Asignar a variable:' section has a text input field containing 'var' and a note 'Nombre de variable sin ()'. At the bottom right, there are two buttons: 'Agregar Comando' (highlighted in blue) and 'Cancelar'.

Resultado Variable:

Para utilizar el id, se debe indicar en este caso, que se requiere el primer elemento del array, para esto se utiliza el comando **“Asignar Variable”**:

The screenshot shows the 'Asignar Variable' window. At the top, it says 'Puede asignar un dato a una variable o un comando predefinido.' Below this, the 'Dato:' section has a text input field containing '{var}[0]'. The 'Asignar resultado a variable' section has a text input field containing '{var}'. At the bottom right, there are two buttons: 'Agregar Comando' (highlighted in blue) and 'Cancelar'.

Resultado de Variable:

The screenshot shows the command bar in the Robot PocketBot interface. It displays 'var' followed by the number '3', indicating that the variable 'var' now holds the value 3.

9.5 Leer email por ID

Se deben ingresar los siguientes datos:

- **ID del email:** ID del correo que queremos leer.
- **Asignar a variable:** Variable donde retornaremos los datos del correo. El nombre de la variable debe ir sin llaves {}.

- **Ruta para descargar adjuntos:** Buscar carpeta donde se descargarán los adjuntos.

Robot: RocketBot

Leer email por ID Outlook365

Leer email por ID

ID del email: {var} Asignar a variable: correo
Nombre de variable sin {}

Ruta para descargar adjuntos: C:\Users\usuario\Desktop Buscar

Agregar Comando Cancelar

Resultado Variable:

correo	{'subject': 'test', 'from': 'fromuser@rocketbot.cl', 'to': 'user@rocketbot.cl', 'body': 'testtttt', 'files': ['Saldo_y_Mov_No_Facturado (1).pdf']}
--------	--

10. Integración con aplicaciones

Excel

Si se trabaja 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 se verá que este se abre.

10.1 Abrir Archivo

Permite abrir un archivo [Excel](#) existente, se debe indicar la ruta donde se encuentra, y al igual que el comando anterior también se puede indicar un Identificador en el caso que se necesite abrir más de un archivo, además de la variable donde se obtendrá un True si se pudo abrir de forma correcta, o un False en caso contrario.

Abrir archivo

Abre un libro de Excel y proporciona un ámbito para las actividades de Excel. Cuando finaliza la ejecución de esta actividad puede cerrar el libro especificado y la aplicación Excel. Esta actividad solo se puede utilizar si la aplicación Microsoft Excel está instalada en su máquina.

Nombre del archivo: Buscar

Asignar resultado a variable: Id: default

Aceptar Cancelar

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:



10.2 Contar Filas

Este comando permite obtener la cantidad total de filas con información del archivo Excel. Por defecto contará las filas de la primera hoja, si se necesita trabajar con otra, primero se debe cambiar con el comando [Cambiar de Hoja](#).

Se debe considerar que sólo contará la cantidad en la columna A, por lo tanto, si se requiere cambiar de columna se puede utilizar el comando **Contar Filas** del módulo [Advanced Excel](#).

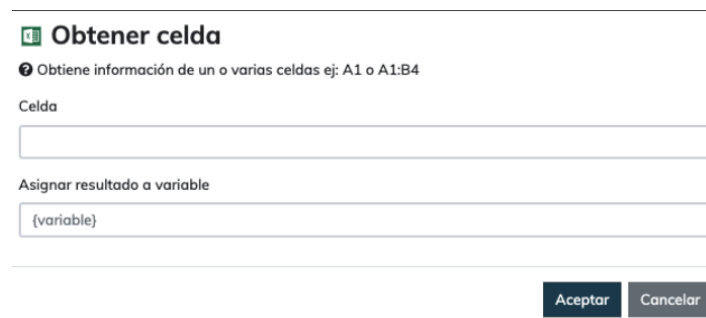


Buenas Prácticas

Siempre luego de Abrir un archivo [Excel](#), se debe 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 se tendrá 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.

10.3 Obtener Celda

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




Buenas Prácticas

Cuando trabajamos con un rango (ej: **A1:F3**) no se debe pasar un número fijo al final de este, sino, que se debe 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.

10.4 Escribir Celda

Este comando permite ingresar datos en las celdas, se puede escribir una celda en particular o especificar un rango



Ejemplo:

En este caso se escribe en Celda especificando el rango **A2:B5** y en Texto se escribe la palabra prueba.

Al ejecutarlo, el **Excel** queda de esta forma.

	A	B	
1			
2	prueba	prueba	
3	prueba	prueba	
4	prueba	prueba	
5	prueba	prueba	
6			
7			

10.5 Guardar Excel

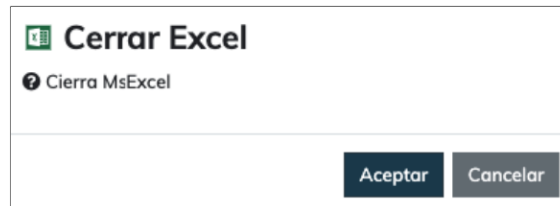
Permite guardar el archivo **XLSX** que está en foco indicando la ruta y nombre, si estamos trabajando con más de uno, primero se debe dejar activo el que se quiere guardar, con el comando **Cambiar de Archivo**.



10.6 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 por **Rocketbot**, se puede utilizar el comando Cerrar **XLSX** del módulo **Advanced Excel**.



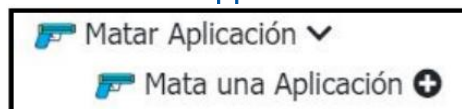
Buenas Prácticas

Una vez que se deje de utilizar el **Excel** se debe cerrar, sino el proceso quedará abierto y al acumular demasiados podría afectar en algo el funcionamiento correcto del bot.

11. Matar aplicación

Este módulo sirve para terminar (matar) una aplicación específica, pasándole como parámetro el nombre.

11.1 Ir al menú **"Mods"** y abrir el módulo **"KillApp"**.



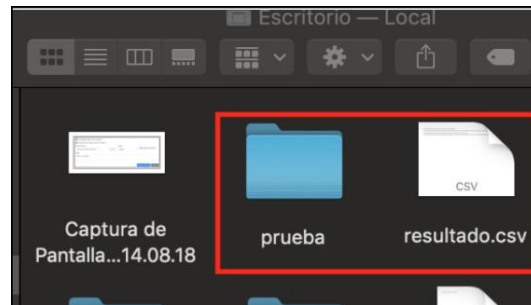
11.2 Matar una aplicación

Para terminar y cerrar una aplicación, se debe indicar el nombre:



12. Borrar Archivo

Este módulo sirve para borrar un archivo con la ruta que se pasó.



 Borrar archivo

 Borra un archivo

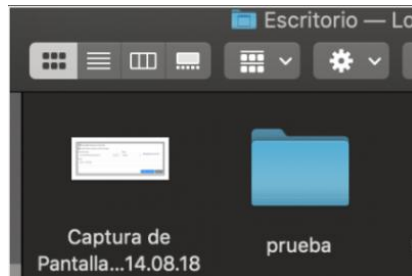
Nombre de archivo:

Asignar resultado a variable

Agregar Comando

Cancelar

Se eliminó el archivo:



13. 0365

Este módulo sirve para realizar las tareas referentes al correo de [Outlook 365](#).

13.1 Conectar a O365

Se deben ingresar los siguientes datos obtenidos de la configuración del portal de [Azure](#).

Robot: Mail_AzureConfig

Conectar a O365

Error Handling

O365

Conectar a una instancia de la aplicación de O365

client_id

client_secret

tenant_id

session

Conectarse a Sharepoint

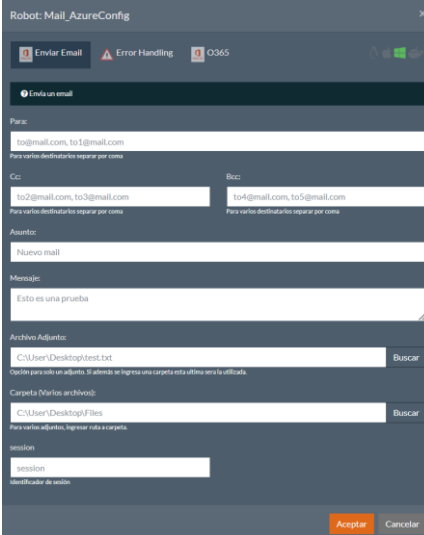
Por defecto, el robot se conecta a Outlook. Si se desea, se puede configurar para que se conecte a Sharepoint.

Aceptar

Cancelar

Nota: En esta versión del manejo de correos no hay un valor de retorno cuando se hace la conexión con el servidor de **O365** a diferencia de los demás módulos de correo como; **Gmail** o **Outlook365**, por lo que no hay con que validar cuando la conexión fue exitosa.

13.2 Envío de Email



The screenshot shows a software interface titled "Robot: Mail_AzureConfig". It features a top navigation bar with icons for "Enviar Email", "Error Handling", and "O365". The main section is titled "Envío de email" and contains several input fields for configuring an email send operation. The "Para:" field is filled with "to2@mail.com, to3@mail.com". Below it, a note says "Para varios destinatarios separar por coma". The "Cc:" field is filled with "to2@mail.com, to3@mail.com" and the "Bcc:" field is filled with "to4@mail.com, to5@mail.com", both with the same note below them. The "Asunto:" field is filled with "Nuevo mail". The "Mensaje:" field contains the text "Esto es una prueba". The "Archivo Adjunto:" section has a text input "C:\User\Desktop\test.txt" and a "Buscar" button, with a note "Opción para solo un adjunto. Si además se agrega una carpeta esta opción sera la utilizada." Below this, the "Carpeta (Varios archivos):" section has a text input "C:\User\Desktop\Files" and a "Buscar" button, with a note "Para varios adjuntos, ingresar ruta a carpeta." At the bottom, there is a "session" field with the value "session" and a label "Identificador de sesión". The interface concludes with "Aceptar" and "Cancelar" buttons.

Nota: Al enviar/listar/leer entre otras opciones, es necesario que en el campo de "**Session**" poner lo mismo que se puso al momento de configurar el paso **13.1**.

Nota: Si al revisar este documento detectas un **cambio en la información**, es necesario notificarlo al área de procesos para su actualización.