Documentación Robot Template

Rocketbot necesita facilitar la revisión y aprobación en el menor tiempo posible de los BOTs enviados por los Partners para su publicación en el Marketplace. Se hace necesario estandarizar el proceso y la forma de construcción, por esta razón se ha habilitado el nuevo módulo ProbeForms en Rocketbot Studio, que les permitirá resolver la interfaz hacia el usuario y el ingreso de los datos requeridos para la operación del BOT. La idea es que los construyan usando este modulo de ingreso de datos y no desarrollar su propia solución, esto para agilizar la revisión y aprobación del mismo. Como una guía para ustedes se construyó este BOT con el fin de usarlo como plantilla para la creación de otros robots en MarketBot, proporcionando una estructura organizada para desarrollar robots especializados. Facilita la creación enfocados en dividir tareas específicas, como configuración, trabajo con Excel, envío de correos, etc. además de validación e ingreso de información a través de un formulario local, optimizando así el uso por parte del usuario. El proceso de ejemplo de este robot recorre y obtiene la información de un Excel para luego llenar un formulario web, obteniendo la respuesta y volviendo al Excel a indicar si fue procesado o no, finalmente envía un correo con el archivo adjunto.

INPUTS

VARIABLES MODIFICABLES

Robot	Variable	Valor Actual	Explicación
main	{espera_datos}	120	Son los 120 segundos de espera a que el usuario ingrese la información en el formulario, esta variable es utilizada en los comandos **Wait submit** del módulo **ProbeForms**

ESTRUCTURA

Proyecto –
|- docs (no usada)
|- logs
|- logs_bot
|- modules (no usada)
|- resources
|- robots (no usada)
|- screenshot (no usada)
|- main.bat
|- robot.db
|- README.md
|- package.json

MÓDULOS

Nombre	Versión
SystemAdvanced	2.8
ProbeForms	1.2.0
gmail_	1.12.7

ROBOTS

main

Robot principal, encargado de dejar todas las variables limpias antes de iniciar, posteriormente llama a los robots hijos para ejecutar cada parte del proceso.

config

Llama al robot para validar los procesos y valida si faltan datos para la configuración, si faltan o es la primera ejecución (todas las variables de config vacías) llamará al robot encargado del ingreso, sino, mostrará el formulario de reinicio para que el usuario decida si quiere reiniciar la configuración e ingresar de nuevo los datos o si prefiere continuar con los datos que ya se tienen.

termina_proceso

Valida si existen procesos de ProbeForms activos y los mata si corresponde.

ingresa_config

Muestra el formulario de ingreso de información y obtiene los datos ingresados por el usuario.

lee_excel

Abre el archivo Excel y extrae la información.

recorre_datos

Recorre los datos obtenidos del Excel y los guarda en sus variables correspondientes, dentro del ciclo llama al robot encargado de ir llenando el formulario web con estos datos, finalmente cierra el navegador y guarda los cambios en el Excel.

llena_form

Llena la información extraída del archivo Excel en el formulario web, valida

la respuesta entregada y escribe el status en la celda correspondiente.

envia_correo

Envía el correo a el/los destinatarios, adjuntando el archivo Excel.

escribe_log

Escribe la información enviada desde los robots en el archivo log customizado, además valida la información de la variable {detener}, si es True dentendrá todo el proceso.

valida_res

Valida la respuesta de la variable {res}, si es False, cerrará el navegador y llamará al robot para escribir el log.

VIDEO EJECUCIÓN

¿CÓMO PROBARLO?

1- Descarga el siguiente .zip □□ y descomprimelo en la carpeta que desees

template_marketbotDownload

2- Descomprime el siguiente .zip □□ que contiene los módulos a utilizar

mods-templateDownload

3- Las 3 carpetas de los módulos (gmail_, ProbeForms y SystemAdvanced) pégalos en tu_carpeta_rocketbot/modules

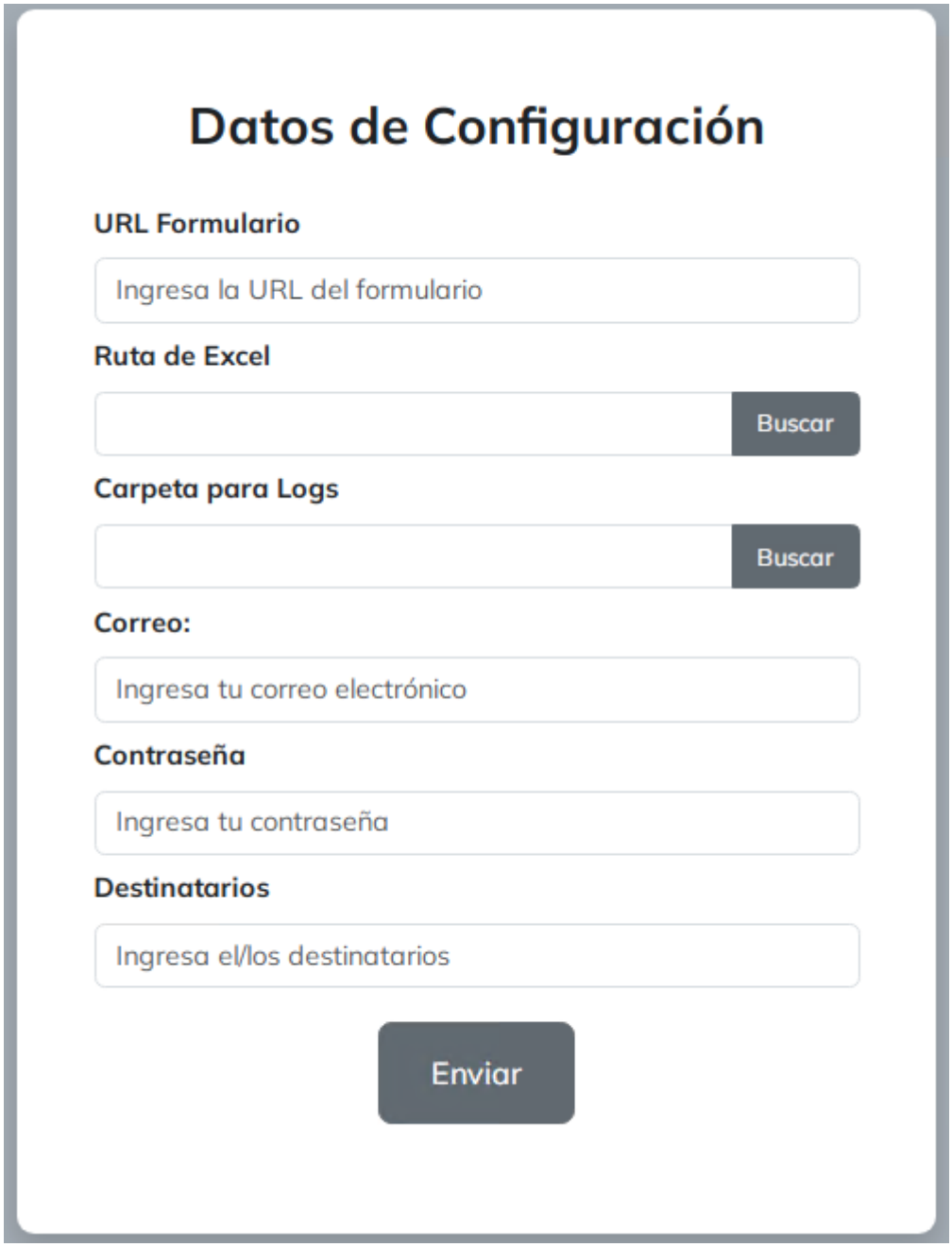
4- Abre Rocketbot Studio y carga la base de datos robot.db que está en la raíz de la carpeta template_marketbot

5- Se mostrará una lista de robots, clickea en el que se llama main

6- Ejecuta el robot con el botón verde → Ejecutar

7- Te mostrará el formulario inicial donde deberás ingresar tu información

□ Considerar que los datos se deben ingresar en un tiempo máximo de 2 minutos, si no se ingresan dentro de ese rango, verás una ventana de que el robot se detuvo y esto significa que finalizó con error. Este tiempo está configurado por defecto, si necesitas aumentarlo, debes modificar la variable {espera_datos} en el robot main



URL Formulario: Debe ser https://roc.myrb.io/dev/forms/INGKKJ0HGGXOJNRZ
Ruta de Excel: debes buscar la ruta de tu carpeta template_marketbot/resources y elegir el archivo datos_form.xlsx
Carpeta para Logs: debes buscar la ruta de tu carpeta template_marketbot/logs_bot
Correo: Debes ingresar el correo desde donde se enviará la información, debe ser un **gmail**
Contraseña: Debes ingresar la ***contraseña de aplicación*** de la cuenta gmail
Destinatarios: Debes ingresar el correo del destinatario, si es más de uno los debes separar por coma

8- Una vez ingresada el robot continuará con la ejecución

9- Una vez finalizada la ejecución clickear el botón Guardar, realizar esto sólo en la primera ejecución.

 Con esto en la próxima ejecución preguntará si se quieren reiniciar o no

los datos de configuración, donde si se elige la opción SI, mostrará un formulario para volver a ingresarlos, en caso contrario el robot continuará con los datos previamente ingresados, ésto para evitar estar escribiendo lo mismo en cada ejecución y sólo realizarlo en caso de que alguna información cambie.

Ejemplo de Bot Padre e Hijo

Ejemplo simple de un Robot Padre encargado de abrir el navegador y setear una variable de búsqueda, posteriormente se ejecuta el bot Hijo, encargado de realizar el click en el input de búsqueda y utilizar la variable del padre, al final desde el hijo se seteará una variable del Padre con un nuevo valor.

Descargar BD de ejemplo: **PadreHijo**

Automatizar Sistema Legacy

Descargar Manual: **Controlar_Sistema_Legacy**

Descargar Manual y Base de Datos con ejemplo: **Legacy_Bot_Example**

Sistema Legacy

Los sistemas Legacy son aplicaciones obsoletas o desactualizadas que siguen dentro de una organización y que no pueden reemplazarse fácilmente, en este caso nos enfocaremos en aplicaciones de tipo Terminal.

Simularemos un sistema Legacy utilizando la cmd.

Estructura

Robot Principal:

- main_legacy
  - **Variables:**
    - **word:** Palabra a buscar en pantalla cmd.
    - **validate:** Variable que recibirá un True para indicar que la palabra asignada a ***{word}*** fue encontrada en la pantalla.
    - **path:** Ruta hacia la carpeta de usuario.

Robot Hijos:

- validateScreen_legacy
  - **Variables:**
    - text: Variable que guardará el texto copiado del cmd.
- screen2_legacy
  - **Variables:**
    - Ninguna

---

1. Iniciar Aplicación: Para abrir la aplicación lo podemos realizar mediante el comando **Ejecutar aplicación** o crear un **.bat** y llamarlo desde ese mismo comando.

> En este caso se trabajará con la cmd, por lo que la abriremos mediante el comando **Asignar Variable,** necesitamos abrir la cmd en la carpeta del usuario, por lo que realizaremos lo siguiente:

- Obtendremos la ruta del usuario del PC controlado reemplazando los backslash (\) por slash (/) y lo guardamos en la variable ***{path}***:



- Iniciamos la cmd en la ruta especificada en la variable ***{path}***:

2. Envío de Instrucción: Enviaremos las instrucciones mediante el comando "**Enviar Tecla**", en este caso necesitamos realizar el envío del texto "**dir**" y luego un **ENTER**.



3. Asignar palabra a buscar: Asignaremos la palabra que necesitamos buscar en la cmd a una variable, en este caso buscaremos el texto "Desktop" y lo guardaremos en ***{word}***.

> **Importante:** No es óptimo utilizar Virtualización en este tipo de Sistemas, puedes usar el envío de teclas y el portapapeles para buscar y/o esperar información en el terminal.

4. Ejecutar bot hijo: Ejecutamos el bot hijo encargado de buscar la palabra asignada a la variable ***{word}*** hasta que la encuentre (dentro de un While.)



- **validateScreen_legacy**
  - **Limpiar Variable: Limpiamos la variable *{validate}* del robot padre antes de iniciar el flujo, con el comando "Limpiar variable(s)" del módulo System++.**



- **Enviar combinaciones: Enviaremos las combinaciones de tecla ctrl+a y ctrl+c mediante el comando “Enviar Combinaciones” del módulo “Teclado”.**

Si se está trabajando con Putty, no permitirá el envío de teclas ctrl+a o ctrl+c, pero se pueden realizar los siguientes pasos para poder copiar toda la información del terminal:

1.- Ir a Window -> Behaviour.
Luego marcar la casilla “***System Menu Appears on ALT-SPACE***” y “***Ensure Windows is always on top***”.

2.- Ir a Session -> Escribir un nombre para guardar la sesión en “**Saved Sessions**”

Cuando esté conectado a la terminal, puede realizar la siguiente combinación de tecla **ALT+SPACE** (en el comando **Enviar Tecla** del menú **Desktop** debes escribir **^{SPACE}**), esperar 1 segundo, y luego enviar la letra **O**, con eso se guardará en el portapapeles.

- **Guardar en Portapapeles: Lo que acabamos de copiar con el comando anterior (ctrl+a y ctrl+c) o ^{SPACE} y luego la letra O si es putty , lo asignaremos a una variable *{text}* mediante el comando “Obtener portapapeles” del módulo “Clipboard”**

- **Buscar hasta: Repetiremos las instrucciones anteriores (ctrl+a , ctrl+c y obtener portapapeles) hasta que la palabra asignada a *{word}* se encuentre en el texto obtenido en *{text}*.**
  Dejaremos la repetición dentro de un **While**, esto simula una espera, buscará hasta que la pantalla que queremos cargue y logre encontrar un texto que nos indique que estamos en la vista correcta, una vez la encuentre, saldrá del While y continuará el flujo.

  - Podemos también agregar un contador en el while para que después de X intentos se salga y no continúe en un ciclo infinito

- **Asignación de variables: Fuera del While limpiaremos la variable *{word}* del padre y luego asignaremos como valor un True a la variable *{validate}* del padre, para indicar que encontró la palabra deseada y nos encontramos en la pantalla correcta.**



5. Ejecutar bot hijo: Ejecutamos el bot hijo encargado de realizar las acciones en la pantalla correspondiente.



- **screen2_legacy**
  - **Validar ejecución anterior: Validaremos mediante un IF si la variable *{validate}* del robot padre es True, eso significa que sí nos encontramos en la pantalla correcta, por lo que ejecutaremos las instrucciones necesarias, en este caso "cd Desktop" y un ENTER para acceder a la carpeta.**



6. Cerrar todo: Al final de las instrucciones del robot principal, debemos cerrar la aplicación, en este caso la cmd, utilizaremos el comando "**Mata una Aplicación**" del módulo "**KillApp**"

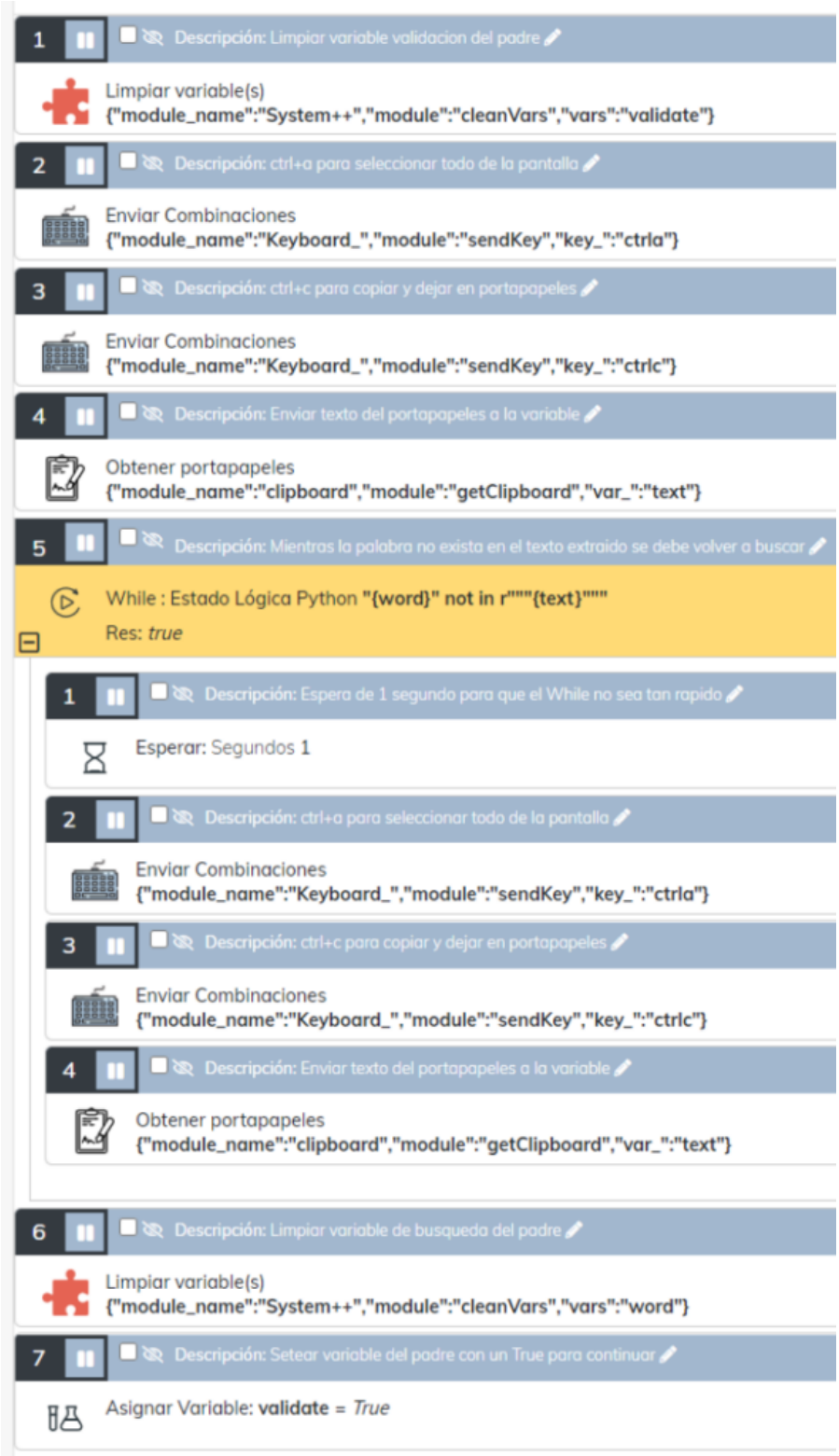


Secuencias:

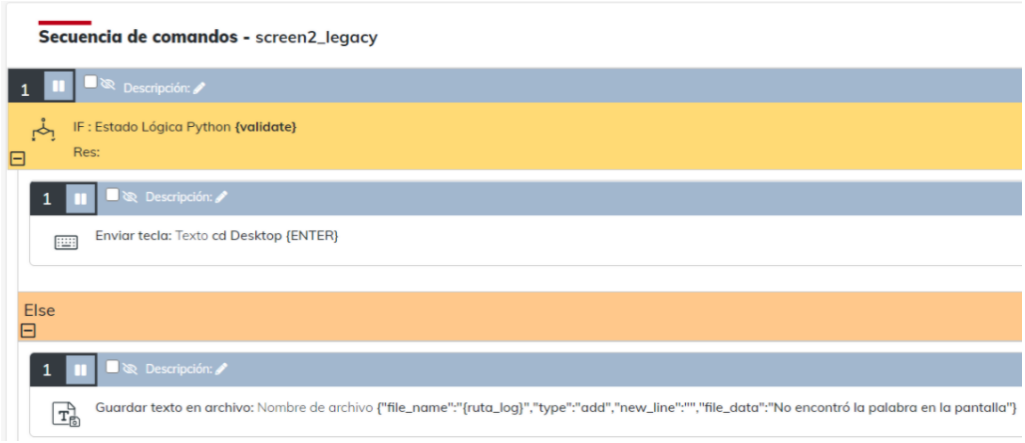
main_legacy:



validateScreen_legacy:



screen2_legacy:



MERCADOLIBRE CON PAGINACIÓN + EXCEL

Robot para ingresar a la página de mercadolibre, realizar una búsqueda, extraer el título y el precio de cada artículo, además de ir recorriendo una a una las páginas y sacando esta información para guardarla en un Excel.

> En el Robot1 debes cambiar la URL para guardar el Excel por la tuya y el texto de la búsqueda por la que quieras.

Descargar Robot1: mercadolibre

Descargar Robot2: mercadolibre_while

Manual: ROBOT_MERCADOLIBRE

BANCO CENTRAL CON EXCEL EXISTENTE

Descargar robot: bcentral_existente

Excel Prueba: indicadores

> En el robot deben cambiar las URL de los archivos y las credenciales del mail, por las suyas.

- Este robot inicia guardando en una variable la fecha actual:





- luego va a la página del Banco Central y espera que cargue un dato, la respuesta se almacenará en la variable “res”



- tomaremos una decisión con un “If”, si el valor de la variable “res” es True (que encontró el elemento que le indicamos) entonces extrae los valores de la UF y el Dólar los almacena en sus respectivas variables y cierra el navegador :



- También dentro del IF, abre un Archivo Excel existente, el cual ya contiene datos,



Cuenta las filas que contiene el Excel y las almacena en la variable “filas”



Incrementamos la cantidad de filas en 1, para que los datos que escribamos queden en la fila siguiente de la última con datos



- Indicamos como Celda **A{filas}** donde **{filas}** es la cantidad de filas que contienen datos más 1 (osea la siguiente) en texto pasamos una lista con las variables que contienen los valores, para que los agregue hacia la derecha (fecha,uf,dolar)

- Esto lo agregará en la fila siguiente



- Guarda el Excel con el nombre más la extensión y lo cierra.



- Finalmente se configura el servidor de correo con los datos y credenciales de GMAIL (permitir aplicaciones poco seguras en la configuración de tu cuenta de Gmail) y envía un mail adjuntando el archivo que creó el robot.





- Si cae en el “Else”, mostraremos una alerta indicando que la página se demoró en cargar.



BANCO CENTRAL

Descargar robot: bcentral_

En el robot deben cambiar las URL de los archivos y las credenciales del mail, por las suyas.

- Este robot inicia guardando en una variable la fecha actual:





- luego va a la página del Banco Central y espera que cargue un dato, la respuesta se almacenará en la variable “res”



- tomaremos una decisión con un “If”, si el valor de la variable “res” es True (que encontró el elemento que le indicamos) entonces extrae los valores de la UF y el Dólar los almacena en sus respectivas variables y cierra el navegador :



- También dentro del IF, crea un Archivo Excel vacío, desde la celda A1 hacia la derecha escribe los títulos (Fecha,UF,DOLAR) y desde la celda A2 hacia la derecha, los valores para cada uno.





- Guarda el Excel con el nombre más la extensión y lo cierra.



- Finalmente se configura el servidor de correo con los datos y credenciales de GMAIL (permitir aplicaciones poco seguras en la configuración de tu cuenta de Gmail) y envía un mail adjuntando el archivo que creó el robot.





- Si cae en el “Else”, mostraremos una alerta indicando que la página se demoró en cargar.

