



Documentación de Rocketbot Studio

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}**:

Asignar Variable


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

Dato:

Asignar resultado a variable

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


Asignar Variable


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

Dato:

Asignar resultado a variable

- 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.

 **Enviar tecla**

 Escribe el texto en la ventana que esta en focus


Texto

Tecla especial

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, se debe usar el envío de teclas y el portapapeles para buscar y/o esperar información en el terminal.


Asignar Variable

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


Dato:

Asignar resultado a variable

- 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.)






Ejecuta otro script RocketBot

 Ejecuta un script RocketBot guardado en la base de datos.

Nombre del script RocketBot:

- **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++.

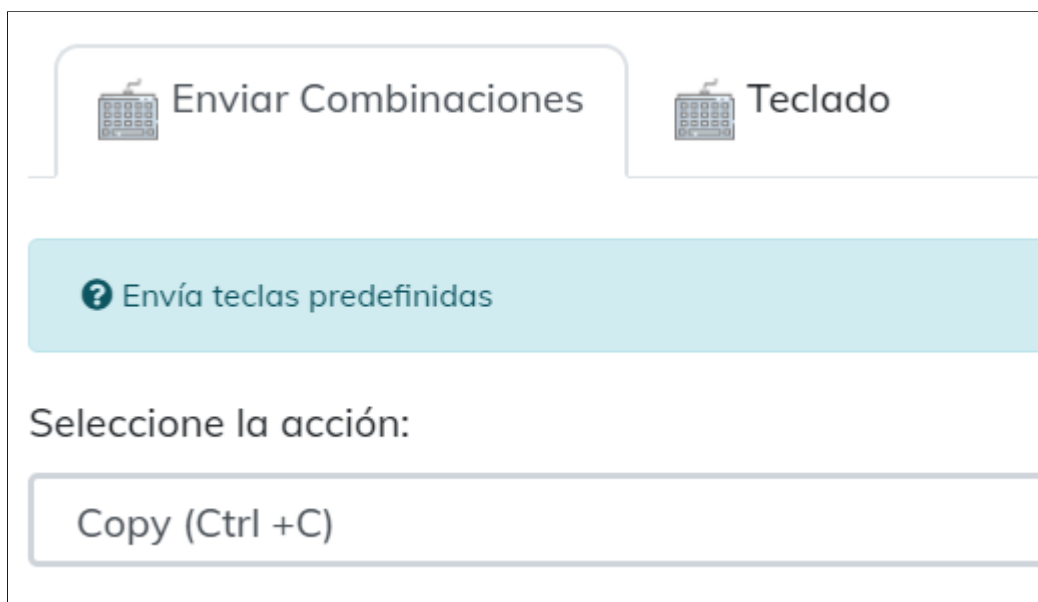
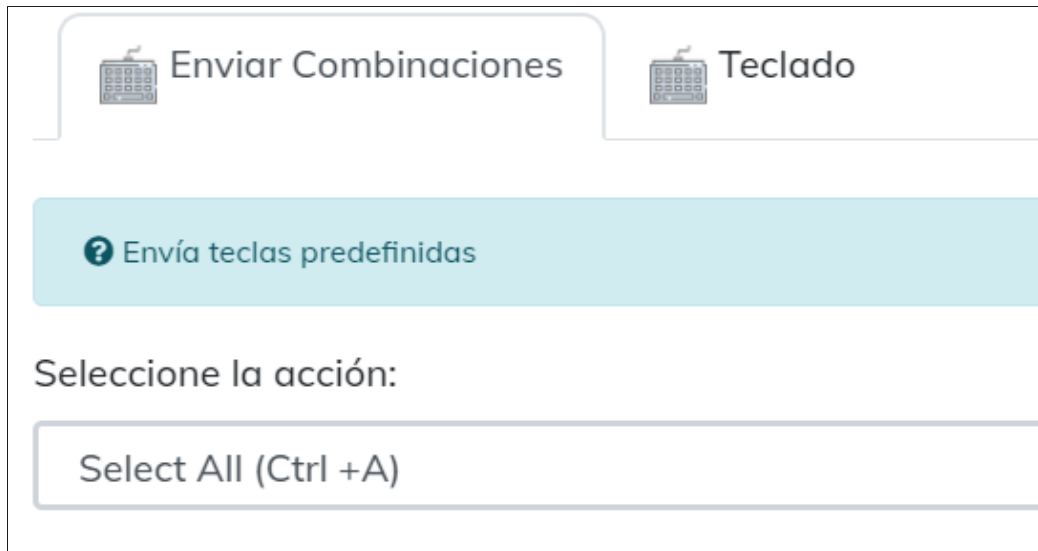
 Limpiar variable(s)  System++

 Ingrese las variables a limpiar separadas por coma

Variables

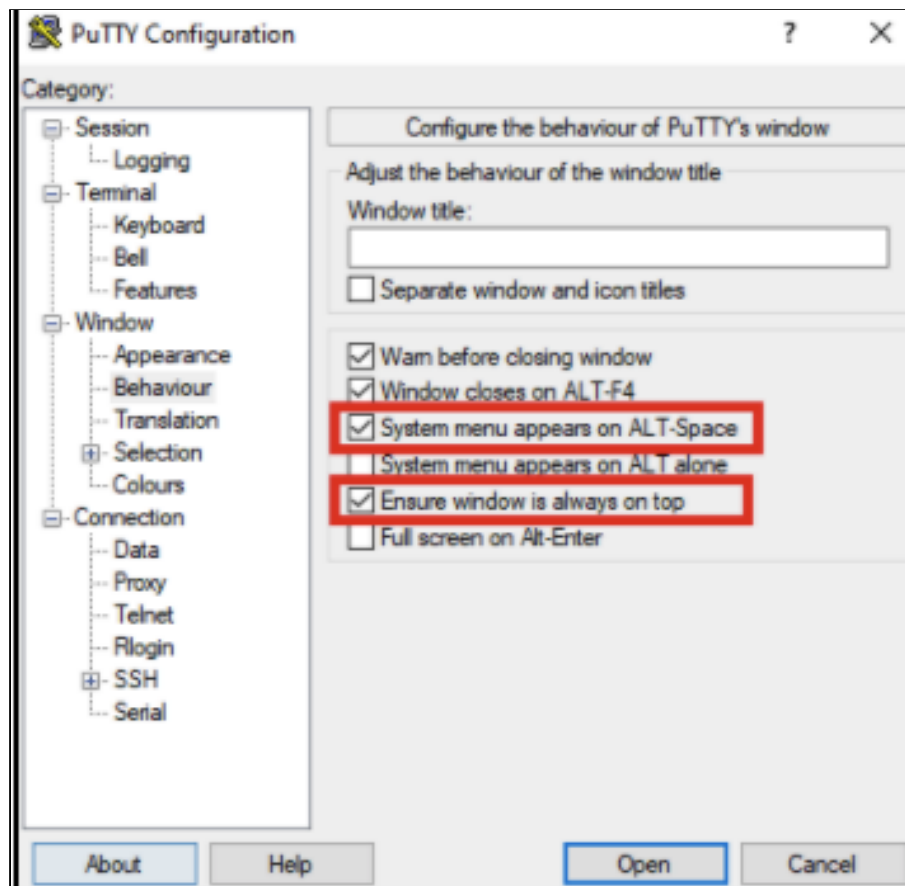
Nombres de variables sin {} y separadas por coma

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

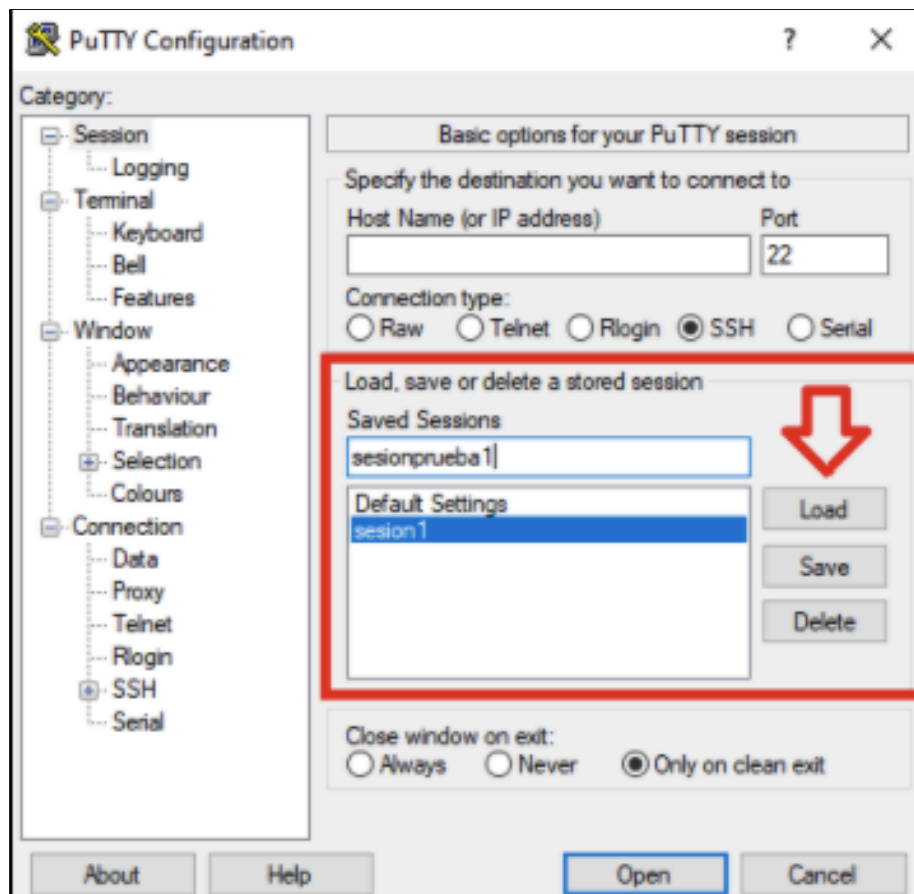


**Si se está trabajando con Putty, no permitirá el envío de teclas ctrl+a o ctrl+c, pero se pueden realizar los siguientes pasos para poder copiar toda la información del terminal:

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

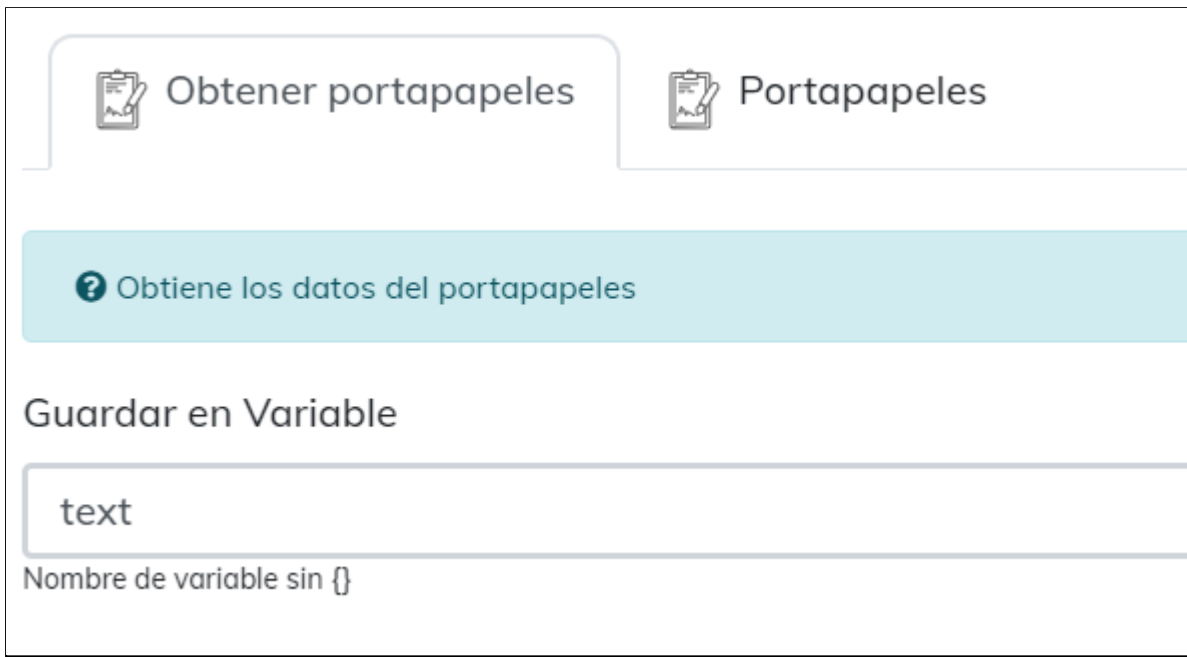


2. Ir a Session -> Escribir un nombre para guardar la sesión en “Saved Sessions”



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 $\wedge\{\text{SPACE}\}$), 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](#)”



The screenshot displays the configuration interface for the 'Obtener portapapeles' module. At the top, two tabs are visible: 'Obtener portapapeles' (selected) and 'Portapapeles'. Below the tabs, a light blue bar contains a question mark icon and the text 'Obtiene los datos del portapapeles'. Underneath, the section 'Guardar en Variable' is shown, with a text input field containing 'text'. Below the input field, the text 'Nombre de variable sin {}' is displayed.



- **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.

The image shows a workflow editor interface with a 'While' loop and four steps:

- Step 5 (While Loop):** Descripción: Mientras la palabra no exista en el texto extraído se debe volver a buscar. Estado Lógica Python: `"{word}" not in r"""{text}""`. Resultado: `true`.
- Step 1:** Descripción: Espera de 1 segundo para que el While no sea tan rapido. Acción: Esperar: Segundos 1.
- Step 2:** Descripción: ctrl+a para seleccionar todo de la pantalla. Acción: Enviar Combinaciones `{"module_name":"Keyboard_","module":"sendKeys","key_":"ctrla"}`.
- Step 3:** Descripción: ctrl+c para copiar y dejar en portapapeles. Acción: Enviar Combinaciones `{"module_name":"Keyboard_","module":"sendKeys","key_":"ctrlc"}`.
- Step 4:** Descripción: Enviar texto del portapapeles a la variable. Acción: Obtener portapapeles `{"module_name":"clipboard","module":"getClipboard","var_":"text"}`.

- **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.


 Limpiar variable(s)  System++

 Ingrese las variables a limpiar separadas por coma

Variables

Nombres de variables sin {} y separadas por coma

Asignar Variable


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

Dato:


Asignar resultado a variable

5. Ejecutar bot hijo

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



Ejecuta otro script RocketBot

 Ejecuta un script RocketBot guardado en la base de datos.

Nombre del script RocketBot:

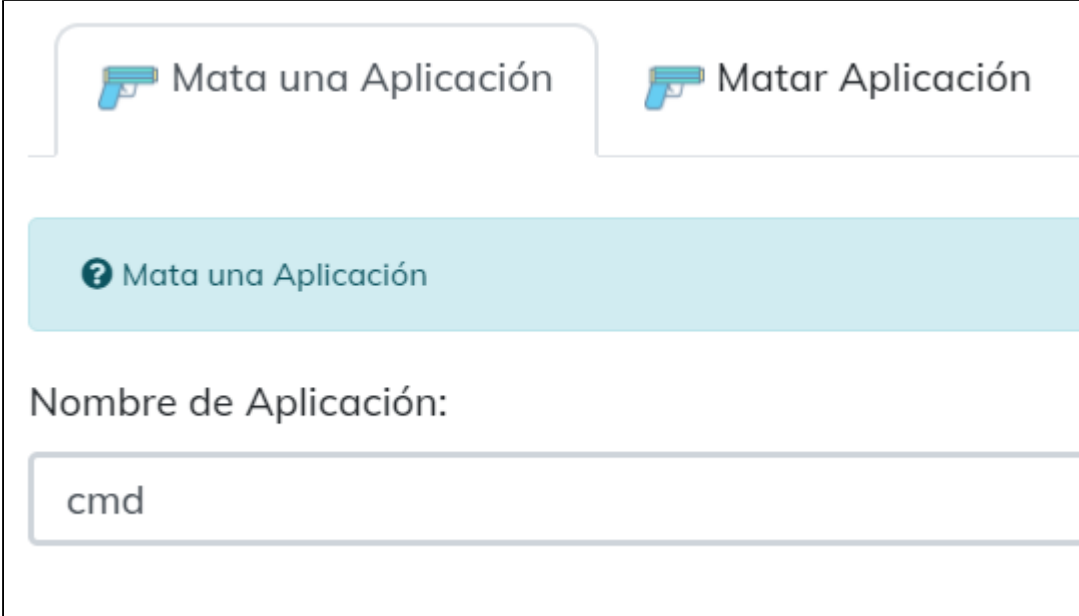
- **screen2_legacy**

- **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.



The screenshot displays the configuration for a script named 'screen2_legacy'. It features a conditional execution block (IF) based on the Python logic variable '{validate}'. If the condition is met, the script performs two actions: sending the text 'cd Desktop' followed by the ENTER key, and saving the text to a file named '{ruta_log}'. The interface includes a description field, a list of actions, and an 'Else' section for alternative logic.

- 6. Cerrar todo** Al final de las instrucciones del robot principal, debemos cerrar la aplicación, en este caso la cmd, utilizaremos el comando “Matar una Aplicación” del módulo “Matar Aplicación”





The screenshot displays the configuration interface for the 'Matar una Aplicación' (Kill an Application) module. At the top, there are two tabs: 'Mata una Aplicación' (Kill an Application) and 'Matar Aplicación' (Kill Application), both featuring a handgun icon. Below the tabs is a light blue bar with a question mark icon and the text 'Mata una Aplicación'. Underneath this bar, the label 'Nombre de Aplicación:' (Application Name:) is followed by a text input field containing the value 'cmd'.

Secuencias:

main_legacy:

Secuencia de comandos - main_legacy

- Descripción: obtenemos la ruta del usuario 
 Asignar Variable: **path** = `(os.environ['USERPROFILE']).replace('\', '/')`
- Descripción: Iniciamos la cmd en la carpeta de usuario 
 Asignar Variable: = `os.system("start cmd /K cd {path}")`
- Descripción: espera de 2 segundos 
 Esperar: Segundos 2
- Descripción: enviar texto a cmd 
 Enviar tecla: Texto dir {ENTER}
- Descripción: setear en variable el texto a buscar en la cmd 
 Asignar Variable: **word** = `"Desktop"`
- Descripción: Ejecutar bot hijo encargado de buscar hasta que encuentre la palabra 
 Ejecuta otro script RocketBot: Nombre del script RocketBot: `validateScreen_legacy`
- Descripción: Ejecutar bot hijo luego de encontrar palabra en cmd 
 Ejecuta otro script RocketBot: Nombre del script RocketBot: `screen2_legacy`
- Descripción: Cerrar cmd 
 Mata una Aplicación
`{"module_name":"killApp","module":"killApp","app_":"cmd"}`

validateScreen_legacy:

The screenshot displays a workflow with the following steps:

- Step 1:** Descripción: Limpiar variable validacion del padre. Limpieza de la variable 'validate'.
- Step 2:** Descripción: ctrl+a para seleccionar todo de la pantalla. Envío de la combinación de teclas Ctrl+A.
- Step 3:** Descripción: ctrl+c para copiar y dejar en portapapeles. Envío de la combinación de teclas Ctrl+C.
- Step 4:** Descripción: Enviar texto del portapapeles a la variable. Obtención del contenido del portapapeles y asignación a la variable 'text'.
- Step 5:** Descripción: Mientras la palabra no exista en el texto extraído se debe volver a buscar. Condición While: Estado Lógica Python `"{word}" not in r"{}{text}"`. Resultado: `true`.
- Step 6:** Descripción: Espera de 1 segundo para que el While no sea tan rapido. Acción de espera de 1 segundo.
- Step 7:** Descripción: ctrl+a para seleccionar todo de la pantalla. Envío de la combinación de teclas Ctrl+A.
- Step 8:** Descripción: ctrl+c para copiar y dejar en portapapeles. Envío de la combinación de teclas Ctrl+C.
- Step 9:** Descripción: Enviar texto del portapapeles a la variable. Obtención del contenido del portapapeles y asignación a la variable 'text'.
- Step 10:** Descripción: Limpiar variable de busqueda del padre. Limpieza de la variable 'word'.
- Step 11:** Descripción: Setear variable del padre con un True para continuar. Asignación de la variable 'validate' a `True`.

screen2_legacy:

Secuencia de comandos - screen2_legacy

1 Descripción:

IF : Estado Lógica Python {validate}
Res:

1 Descripción:

Enviar tecla: Texto cd Desktop (ENTER)

Else

1 Descripción:

Guardar texto en archivo: Nombre de archivo {"file_name":"{ruta_log}","type":"add","new_line":"","file_data":"No encontró la palabra en la pantalla"}