

REALIZAR UN MÓDULO EN ROCKETBOT

Para realizar un módulo para Rocketbot debemos considerar lo siguiente:

- Editor de texto donde escribiremos el código.
- Archivo `__init__.py` el cual contendrá el código Python.
- Archivo `package.json` el cual contendrá la vista.

Template de package.json:

[package.json](#)

Template de __init__.py:

[__init__.py](#)

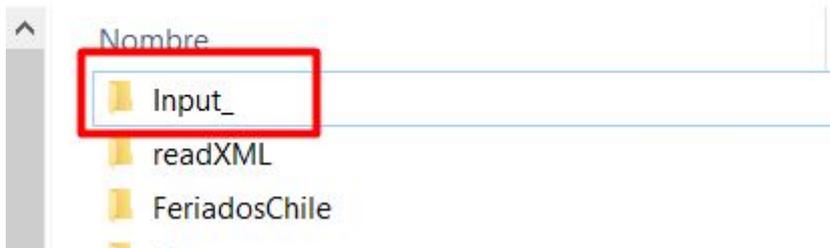
***Realizaremos el módulo **Input** como ejemplo.*



Este módulo levantará una ventana donde solicitará datos al usuario como por ejemplo, título y texto de la ventana y una variable donde almacenar lo ingresado por el usuario.

1. Crear una carpeta con el nombre del módulo a realizar en la carpeta “**modules**” de Rocketbot. Le pondremos **Input_**

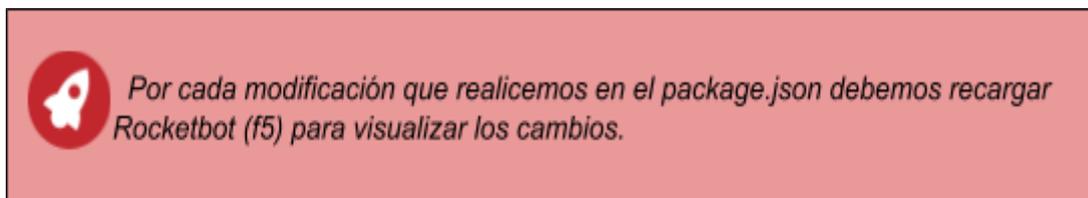
Este equipo > Escritorio > Rocketbot > modules



2. Pegar los archivos de template **package.json** e **__init__.py** dentro de la carpeta **Input_**



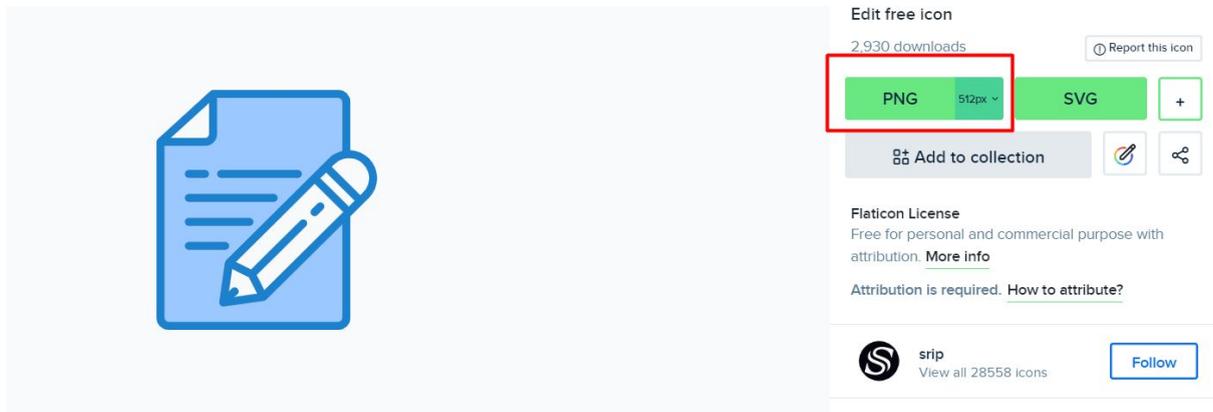
3. Editar archivo **package.json**:



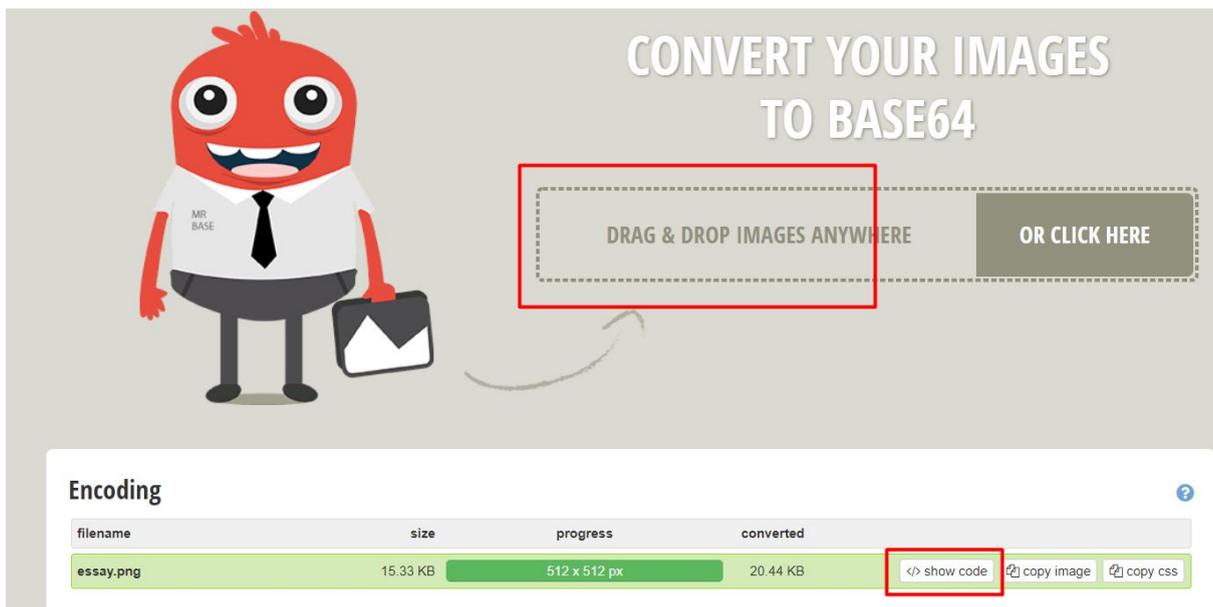
Primero editaremos la parte informativa del módulo.

author: Nombre del creador del módulo
description: descripción de lo que hace el módulo
disclaimer: Dejar por defecto,
version: Versión del módulo (Empieza en 1.0, si se realizan cambios actualizar la versión)
license: Dejar por defecto
homepage: Página de Rocketbot (<http://rocketbot.co>)
linux: true o false si el módulo funciona en Linux o no
windows: true o false si el módulo funciona en Windows o no
mac: true o false si el módulo funciona en Mac o no
docker: true o false si el módulo funciona en Docker o no
name: Nombre del módulo
dependencies: Nombre de librería utilizada y su versión
title:
 en: Título del módulo en Inglés
 es: Título del módulo en Español
icon: ícono del módulo en base64

Para el ícono recomendamos por ejemplo [flaticon](https://flaticon.com), buscar la imagen y descargarla



Luego convertirla a base64, puede ser en el siguiente link [base64-image](https://base64-image.com), arrastrar la imagen y dar click en **show code**



Copiar la primera opción:



Filename: essay.png
Filesize: 15.33 KB
Encoded: 20.44 KB
Width: 512 px
Height: 512 px

X

toggle background

For use in elements:

select all

copy to clipboard

```
data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACAYAAAD0eNT6AA
```

For use as CSS background:

select all

copy to clipboard

```
url("data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACAYAAAD0eNT6AA")
```

y eso pegarlo entre comillas dobles en la etiqueta "icon"

Quedará así:

```
"author": "Tu nombre",  
"description": "Solicita ingresar un texto y lo almacena en  
una variable",  
"version": "0.1",  
"name": "Input_",  
"dependencies": {  
  "PyAutoGUI": "0.9.48"  
},  
"o": {  
  "en": "Enter Text",  
  "es": "Ingresar Texto"  
},  
"icon": "data:image/png;base64,iVBORw0KGgoAAAANS..."
```

Ya lo podemos ver en el menú **"Mods"**

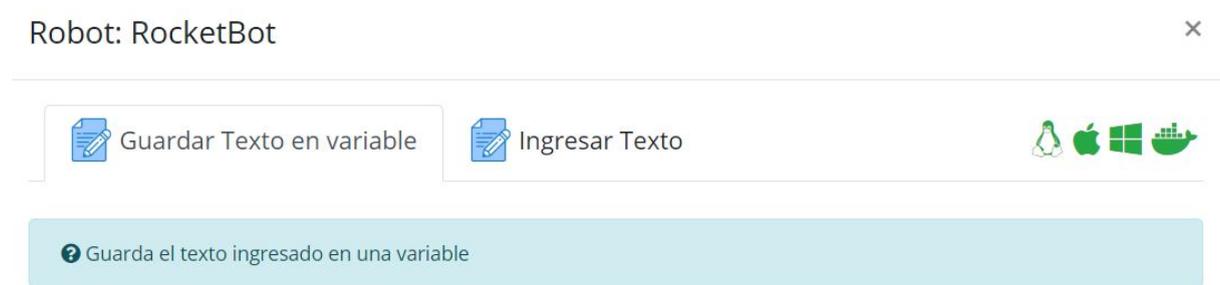


Luego viene la etiqueta **children**, donde agregaremos los input para que el usuario ingrese datos.

Primero tenemos el título del comando y su descripción, ambos en Inglés y Español

```
{
  "en": {
    "title": "Store text in a variable",
    "description": "Store text in a variable",
    "title_options": "Select Option",
    "options": null
  },
  "es": {
    "title": "Guardar Texto en variable",
    "description": "Guarda el texto ingresado en una variable",
    "title_options": "Seleccione una opción",
    "options": null
  },
}
```

Al visualizarla en Rocketbot Studio se verá así:



Ahora modificaremos los datos de la etiqueta **forms**.

Primero en **inputs** definiremos lo siguiente:

type: Tipo de elemento, en este caso será un input

placeholder: Podemos agregar una ayuda en el input para que el usuario entienda qué dato debe ingresar

title:

en: Título del comando en Inglés

es: Título del comando en Español

help:

id: Identificador del elemento, con el cual luego en el archivo `__init__.py` obtendremos el dato ingresado por el usuario.

css: Donde indicaremos el tamaño de la ventana del comando, en este caso ventana mediana con 12 columnas, es decir que el input utilice todo el ancho de la ventana (`col-md-12`).

Para este ejemplo necesitaremos 3 input, uno donde el usuario ingresará el título de la ventana, otro para ingresar el texto y por último el de la variable donde almacenaremos el dato ingresado, por lo tanto debemos repetir 3 veces las etiquetas dentro de la opción "inputs" indicando en cada una sus respectivos datos.

Input 1

```
{
  "type": "input",
  "placeholder": "Prompt",
  "title": {
    "es": "Ingrese el título de la ventana:",
    "en": "Enter the window title:"
  },
  "id": "title_",
  "css": "col-md-12"
},
```

Con estos datos la vista de este input en Rocketbot estudio será la siguiente:

Ingrese el título de la ventana:

Input 2

```
{
  "type": "input",
  "placeholder": "Ingrese texto:",
  "title": {
    "es": "Ingrese el texto de la ventana:",
    "en": "Enter the window text:"
  },
  "id": "text_",
  "css": "col-md-12"
},
```

Con estos datos la vista de este input en Rocketbot estudio será la siguiente:

Ingrese el texto de la ventana:

Input 3

```
{
  "type": "input",
  "placeholder": "Variable",
  "title": {
    "es": "Asignar resultado a variable:",
    "en": "Assign result to a variable:"
  },
  "help": {
    "es": "Nombre de variable sin {}",
    "en": "Variable name without {}"
  },
  "id": "var_",
  "css": "col-md-12"
}
```

Con estos datos la vista de este input en Rocketbot estudio será la siguiente:

Asignar resultado a variable:

Nombre de variable sin {}

La estructura completa se verá así:

```
"form": {
  "css": "modal-lg",
  "inputs": [
    {
      "type": "input",
      "placeholder": "Prompt",
      "title": {
        "es": "Ingrese el título de la ventana:",
        "en": "Enter the window title:"
      },
      "id": "title_",
      "css": "col-md-12"
    },
    {
      "type": "input",
      "placeholder": "Ingrese texto:",
      "title": {
        "es": "Ingrese el texto de la ventana:",
        "en": "Enter the window text:"
      },
      "id": "text_",
      "css": "col-md-12"
    },
    {
      "type": "input",
      "placeholder": "Variable",
      "title": {
        "es": "Asignar resultado a variable:",
        "en": "Assign result to a variable:"
      },
      "help": {
        "es": "Nombre de variable sin {}",
        "en": "Variable name without {}"
      },
      "id": "var_",
      "css": "col-md-12"
    }
  ]
},
```

Finalmente configuraremos los siguientes datos

"**video_youtube**": quedará por defecto
"**icon**": ícono del submódulo en base64
"**module**": Nombre del submódulo
"**module_name**": Nombre del módulo (carpeta)
"**visible**": valor por defecto
"**options**": valor por defecto
"**father**": valor por defecto
"**group**": valor por defecto
"**linux**": true o false si el módulo funciona en Linux o no
"**windows**": true o false si el módulo funciona en Windows o no,
"**mac**": true o false si el módulo funciona en Mac o no
"**docker**": true o false si el módulo funciona en Docker o no

En este caso quedará así:

```
"video_youtube": "aMV6cXPMMSo",  
"icon": "data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAg",  
"module": "sendText",  
"module_name": "Input_",  
"visible": true,  
"options": false,  
"father": "module",  
"group": "scripts",  
"linux": true,  
"windows": true,  
"mac": true,  
"docker": true
```

La vista final con los input se verá así:

 Guardar Texto en variable  Ingresar Texto    

 Guarda el texto ingresado en una variable

Ingrese el título de la ventana:

Ingrese el texto de la ventana:

Asignar resultado a variable:

Nombre de variable sin {}

Agregar Comando Cancelar

4. Editar archivo `__init__.py`:

La primera parte la dejamos por defecto ya que son los comentarios de las instrucciones que podemos utilizar en los módulos

```
# coding: utf-8
"""
Base para desarrollo de modulos externos.
Para obtener el modulo/Funcion que se esta llamando:
    GetParams("module")

Para obtener las variables enviadas desde formulario/comando Rocketbot:
    var = GetParams(variable)
    Las "variable" se define en forms del archivo package.json

Para modificar la variable de Rocketbot:
    SetVar(Variable_Rocketbot, "dato")

Para obtener una variable de Rocketbot:
    var = GetVar(Variable_Rocketbot)

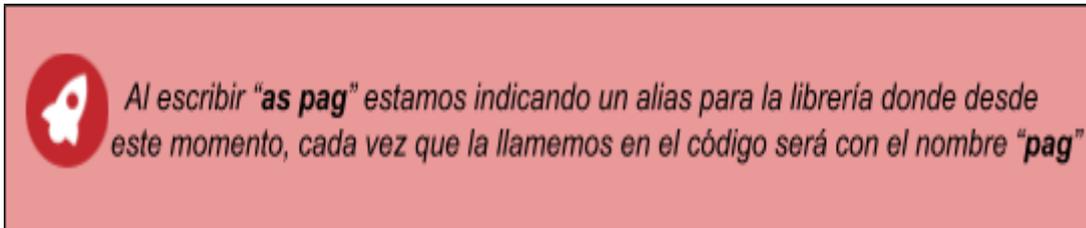
Para obtener la Opcion seleccionada:
    opcion = GetParams("option")

Para instalar librerias se debe ingresar por terminal a la carpeta "libs"

    pip install <package> -t .

"""
```

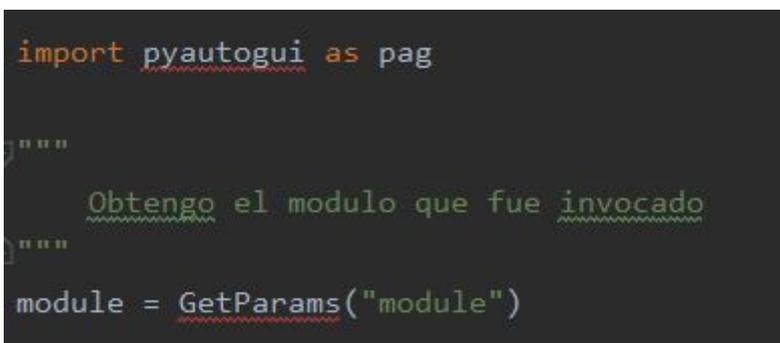
Luego, debemos importar la librería a utilizar en el caso de ser necesario, este ejemplo requiere **pyautogui**, por lo tanto la primera instrucción será:



```
import pyautogui as pag
```

Ahora con la siguiente instrucción obtendremos el nombre del submódulo utilizado

```
module = GetParams("module")
```



Recordemos que es el nombre ingresado en el package.json al final.



Dependiendo del módulo utilizado por el usuario es que realizaremos las tareas específicas, para eso agregamos lo siguiente:

```
if module == "sendText":
```

Si el módulo que se está utilizando corresponde a “**sendText**”, entonces obtendremos los datos ingresados por el usuario mediante los id declarados para cada input en el **package.json**

```
text_ = GetParams('text_')
title_ = GetParams('title_')
var_ = GetParams('var_')
```

Ya con las variables listas, agregaremos la siguiente instrucción:

```
text = pag.prompt(text=text_, title=title_)
```

donde **text_** y **title_** son las variables con los datos ingresados por el usuario

Finalmente con la siguiente instrucción indicaremos que la variable de Rocketbot (id **var_**) será seteada con el valor de la variable **text_** la cual contiene el dato ingresado por el usuario.

```
SetVar(var_, text)
```

Toda la instrucción quedará así:

```
if module == "sendText":
    text_ = GetParams('text_')
    title_ = GetParams('title_')
    var_ = GetParams('var_')

    text = pag.prompt(text=text_, title=title_)

    SetVar(var_, text)
```