

Saturn Studio – Módulo Webhooks

Módulo para manejo de Webhooks dentro de la plataforma de automatización. Permite recibir datos externos mediante solicitudes HTTP y responderlos automáticamente o tras ejecutar un robot.

Saturn Studio/Rocketbot – File Storage

La sección **File Storage** (almacenamiento de archivos) permite gestionar archivos para el desarrollo de robots dentro de **Saturn Studio**.

Es importante considerar la capacidad máxima de almacenamiento disponible. Dentro del entorno de desarrollo, es necesario dirigirse al módulo **File**, donde se podrá configurar su uso según las necesidades del robot.

Es posible hacer uso de los archivos para diversas funciones, por ejemplo, enviar correos electrónicos (**módulo Gmail**) o enviar mensajes con imágenes al asistente de ChatGPT (**módulo OpenAI**), entre otras opciones.

Se pueden realizar las siguientes acciones:

- Subir archivos.
- Visualizar la cantidad de archivos almacenados.
- Visualizar el almacenamiento utilizado y el almacenamiento máximo disponible.

- Visualizar la información del archivo, incluyendo su nombre, proveedor (usuario), tamaño y fecha de creación.
- Eliminar archivos de manera individual o en su totalidad.
- Consultar las tareas asignadas a cada archivo.

The screenshot shows a 'File Storage' interface. At the top, it indicates '1 Files' and '16.94 KIB / 100 MIB' of storage used. Below this is an 'Upload a file' section with a search box containing 'ticket test.jpg'. A table below lists the file details:

File Name	Provider	Size	Created	Actions
ticket test.jpg	user	16.94 KIB	in a few seconds	[Delete] [Refresh]

Files (Desarrollo de Bots)

Para utilizar los archivos previamente cargados en el sistema, es necesario acceder al módulo **Files** y seleccionar el comando **Get Files** (obtener archivos).




En la configuración, se podrá filtrar la búsqueda de archivos por nombre.

Se puede asignar el resultado de la ejecución a una variable, para confirmar la existencia del archivo. Si el archivo existe, la variable asignada contendrá su **ID**, nombre, espacio ocupado en almacenamiento y el proveedor correspondiente.


Get files ✕

Add a description 


 Get files uploaded to the system

{ x }

 Test Command

Filter by name 

ticket test

Assign result to variable 

file

Cancel

Ok

Robot: 1 ✕

Add

Name:

file

Type :

▼

Disable

Group:

Data: [RAW view](#) [JSON view](#) [Table view](#)

```
[{"id":"775e63b67073754a3aa9337b6304e867b87a4b824cb8f851ea7ddc7e7c6989ee","filename":"ticket test.jpg","size":17348,"provider":"user"}]
```

Edit

Cancel

Ejemplo de Desarrollo

Una vez que el archivo ha sido subido correctamente a **File Storage**, durante el desarrollo del bot se deberá acceder al módulo **Files** y seleccionar **Get Files**.

File Storage

Here you can view all the files stored in the system.
You can delete files that are no longer needed. Can config your robot to delete files automatically after stopping the robot.

1
Files

16.94 KIB / 100 MIB
Used storage / Max storage

Upload a file

Seleccionar archivo

File Name	Provider	Size	Created	Actions
ticket test.jpg	user	16.94 KIB	in a few seconds	✖ ✔

- Email IMAP and SMTP
- **Files**
- Get files
- Gemini
- Gmail

Start

Get files
`{"module_name": "files", "m..."`

En este proceso:

1. Se establecerá el nombre del archivo como criterio de búsqueda.
2. Se asignará una variable que verificará su existencia.

Get files
✖

Add a description

? Get files uploaded to the system { x } 🔧 Test Command


Filter by name ?




Assign result to variable ?

Cancel
Ok

Para evaluar la presencia del archivo, se utilizará un comando **IF**, que analizará la lógica basada en la variable asignada. Si la variable confirma la existencia del archivo, el flujo de ejecución continuará.

IF

Add a description 

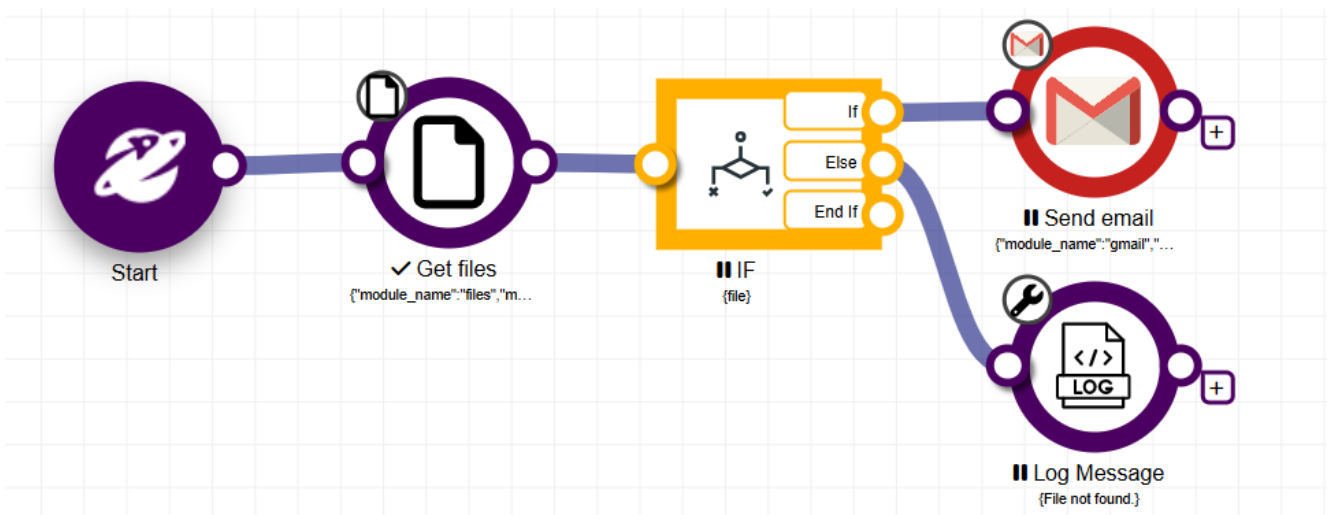
 Evaluate a condition with JS logic   Test Command

Javascript Logic

{file}










Cancel Ok

La función final de este bot será enviar un correo electrónico con el archivo adjunto. Para ello, se empleará el módulo **Gmail**, seleccionando la opción **Send Email**.



Una vez completados los campos requeridos, en la sección **Attached Files** (archivos adjuntos), se podrán visualizar los archivos contenidos en la variable. Si la carga del archivo es correcta, se procederá a asignar una variable que confirme el envío exitoso del correo.


Message ?

B I U  " </>    Normal  A  Sans Serif   

Hello

You can use variables between {} to customize the message for example: Hello {name}

Attached files ?

file 

Files in variable

- ticket test.jpg - USER

Assign result to variable ?

send_file 

Cancel

Ok

Si todo el proceso se ejecuta correctamente, el archivo adjunto será enviado a la dirección de correo electrónico designada.

Saturn Studio/Rocketbot: Data Store/Cache

En esta sección se almacenarán y podrán compartirse datos entre distintos robots. Los robots tendrán la posibilidad de crear, actualizar, obtener registros de datos o eliminarlos durante su desarrollo.

New Data Store

Desde el dashboard, en el menú lateral, encontraremos la opción **Data Store/Cache**, donde ingresamos a esta sección que nos mostrará la siguiente información:

- La cantidad de datos almacenados en el sistema.
- El espacio de almacenamiento ocupado.

The screenshot shows the 'Data store / Cache' section of a software interface. At the top, it displays '0 Data Stores' and '0 Bytes/10 MIB' used storage. Below this is a table with columns for Name, Description, Items, Size, and Actions. A '+ New Data Store' button is visible above the table.



Name	Description	Items	Size	Actions
------	-------------	-------	------	---------

Podemos crear un almacén de datos (**New Data Store**), para lo cual se nos pedirá un nombre y una descripción.

The 'New Data Store' dialog box is shown with a dark red header. It contains two input fields: 'Data store name' with the text 'Test Data Store' and 'Data store description' with the text 'Is a test'. At the bottom, there are 'Close' and 'Save changes' buttons.

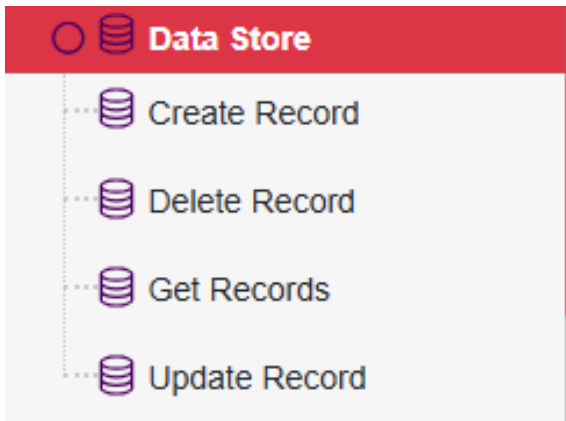
Allí podremos observar el nombre, la descripción, los ítems almacenados, el espacio que ocupan y las acciones disponibles, como editar o eliminar el almacén de datos.

The screenshot shows the 'Data store / Cache' section after creating a new data store. It now displays '1 Data Stores' and '0 Bytes/10 MIB' used storage. The table below contains one entry: 'Test Data Store' with description 'Is a test', 0 items, and 0 Bytes size. The Actions column for this entry shows edit and delete icons.

Name	Description	Items	Size	Actions
Test Data Store	Is a test	0	0 Bytes	 

Data Store (desarrollo)

En el desarrollo del robot, tendremos las siguientes opciones:



- Crear un registro en el almacén de datos (**Create a record in Data Store**).
- Borrar un registro en el almacén de datos (**Delete a record in Data Store**).
- Obtener registros del almacén de datos (**Get records from Data Store**).
- Actualizar un registro en el almacén de datos (**Update a record in Data Store**).

Create Record

Luego de la creación de un almacén de datos, podemos generar un registro desde el desarrollo de un robot.



Para ello, el comando pedirá seleccionar el **Data Store** (datos almacenados a editar).

Create Record

Add a description

Create a record in Data Store {x} Test Command

Data Store

Test Data Store

Record

`{"name": "John", "age": 30}`

Use a variable or a direct value. Example: `{"name": "John", "age": 30}`

Assign result to variable

Select one

En **Record** (registro), se decidirá si se utilizará una variable o un valor directo y, posteriormente, se asignará el resultado a una variable.

Create Record

Add a description

Create a record in Data Store {x} Test Command

Data Store

Test Data Store

Record

this test is for:data store

Use a variable or a direct value. Example: `{"name": "John", "age": 30}`

Assign result to variable

result_create_record

En la variable donde se guardó el resultado, se obtendrá el **ID** y el valor asignado previamente. Este **ID** servirá para actualizar o borrar registros en otros comandos y/o desde otros robots.

Add

Name:

Type :

Disable

Group:

Data: [RAW view](#) [JSON view](#) [Table view](#)

```
[{"id":"8128eae0e68bf3e9b375ab62acb4b1c1","values":"this test is for: data store"}]
```

[Edit](#)[Cancel](#)

Delete Record

Para borrar un registro, se seleccionarán los datos almacenados y se ingresará el **ID** que fue entregado por la variable cuando se creó el registro.

Delete Record

Add a description 

 Delete a record in Data Store

{ x }

 Test Command

Data Store 


Test Data Store



Record ID 

8128eae0e68bf3e9b375ab62acb4b1c1

The ID of the record that will be deleted

Assign result to variable 


delete_data_store













Cancel

Ok

Por último, se asignará una variable para confirmar si el registro fue eliminado exitosamente. El resultado será **true** o **false**.

 Add

	Disable	Name	Data	
	<input type="checkbox"/>	result_create_record	<pre>[[{"id":"8128eae0e68bf3e9b375ab62acb4b1c1","values"...</pre> <p>Expand (53 characters)</p>	   
	<input type="checkbox"/>	delete_data_store	true	   

Get Records

Este comando permitirá obtener los registros almacenados. Para ello, se seleccionará el **Data Store** y luego se aplicarán los filtros necesarios.

Los filtros permiten realizar búsquedas específicas dentro de los registros. Por ejemplo, se puede establecer un filtro que busque registros en los que el campo “nombres” contenga “John”.

Get Records ✕

Add a description

Get records from Data Store {x} Test Command

Data Store

Test Data Store ▼

Filter

Object	OPERATOR	Comparison	
Object		Comparison	Add

Use a variable or a direct value. Example: 'name' contains 'John'

Assign result to variable

result_get_records ▼

Cancel Ok

Para configurar el filtro:

1. Se elige el objeto.
2. Se selecciona la acción a realizar.
3. Se agrega una comparación.

Finalmente, se asignará una variable que almacenará el **ID** y la acción realizada (por ejemplo, "contiene").

Add

Name:

Type :

Disable

Group:

Data: [RAW view](#) [JSON view](#) [Table view](#)

```
[{"id":"f33b33b2f15266917ddc3e42e4d1670f","values":"this test is for:data store"}]
```

Edit

Cancel

Si otro robot utiliza este comando (**Get Records**), podrá acceder a los datos de la variable asignada en el robot anterior.

Update Record

Este comando permite actualizar o editar un dato almacenado.

Para ello, se debe:

1. Seleccionar el dato almacenado.
2. Ingresar el **ID**, obtenido previamente al crear el registro.
3. Editar el registro según sea necesario.
4. Asignar una variable para almacenar el resultado, que incluirá el **ID** y los valores actualizados del registro.

Update Record ✕

Add a description 

 Update a record in Data Store

{ x }

 Test Command

Data Store 

Test Data Store




Record ID 

f33b33b2f15266917ddc3e42e4d1670f

Record 

name:jose

Assign result to variable 

update_record



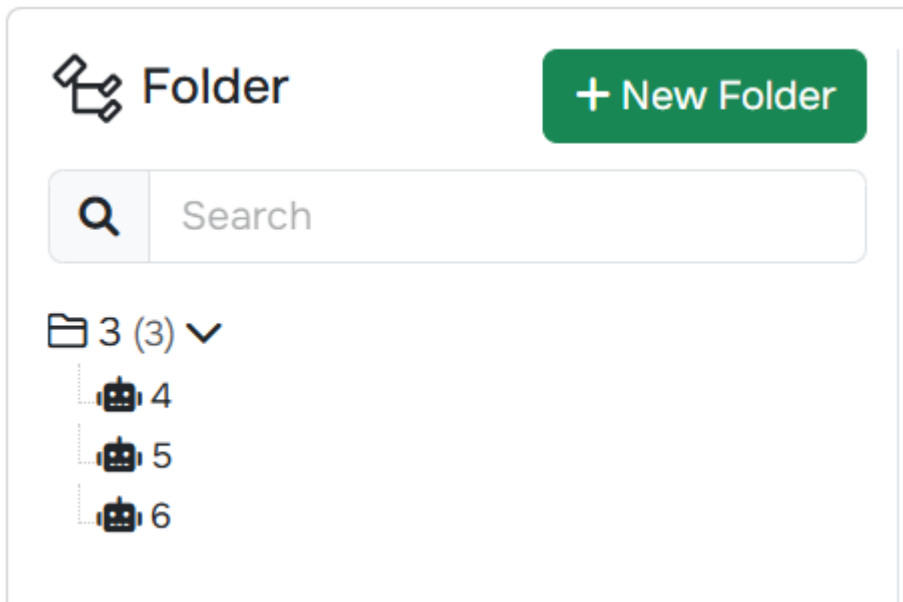
Cancel

Ok

Saturn Studio/Rocketbot – Folder and Bots

Folder (Carpetas)

Una carpeta (folder) es un proyecto que puede contener uno o más robots.



Todos las carpetas del usuario estarán en una lista, junto con el nombre de todos sus robots. Hay un filtro para buscar carpetas por su nombre.

Crear carpeta

Cada carpeta debe tener un nombre, la descripción es opcional. Si posee una descripción, se puede utilizar la IA para generar un documento con el paso a paso del proyecto.

Folder name

read emails

Assign a name to your folder, can use letters, numbers and spaces (azAZ09_)

Folder description

will have robots that read mails and can write and send messages within mails.

Generate documentation with the necessary sub-processes with IA?

Generate with IA



Generating documentation..

Close

Save changes

Folder name

read emails

Assign a name to your folder, can use letters, numbers and spaces (azAZ09_)

Folder description

will have robots that read mails and can write and send messages within mails.

Generate documentation with the necessary sub-processes with IA?

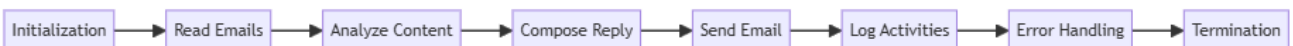
Generate with IA

Project Title:

Email Automation Robot: Reading, Writing, and Sending Messages

Project Sections Table:

#	Title	Description
1	Initialization	Start the automation and set up the environment.
2	Read Emails	Access the email account and read incoming messages.
3	Analyze Content	Analyze the content of the emails to determine actions (e.g., responses needed).
4	Compose Reply	Draft a response email based on the content and context of the original message.
5	Send Email	Send the composed reply or follow-up message back to the sender.
6	Log Activities	Log all actions taken for tracking and auditing purposes.
7	Error Handling	Implement error handling for issues that may arise during email processing.
8	Termination	Properly close connections and end the automation process.

BPM (Business Process Model) View of the Connected Sections:**Note:**

This workflow outlines the necessary steps to automate email interactions effectively using robotic process automation tools like Rocketbot. Each part can be implemented as independent modules for better management and debugging.

Close


Save changes

Y una vez se guarden los cambios aparecerá la carpeta nueva.

 Folder


[+ New Folder](#)

 3 (3) >

 read emails (0) v

Folder and Bots

Cuando se selecciona una carpeta, le podrá crear/editar/eliminar robots, ver la documentación (si es que se hizo) y editar, borrar e imprimir la descripción del proyecto. Se dispone una lista con todos los robots que la carpeta posea.


 Folder read mail

[+ New Robot](#)


Description:

will have robots that read mails and can write and send messages within mails.

 Robots:

Robot	Description	Actions
-------	-------------	---------

 Documentation:

Project Title:

Email Automation with Bots: Read and Send Messages

[View document](#)


Table of Project Sections:

#	Title	Description
1	Initialization	Start the automation and configure initial settings.

New robot

Cada robot debe tener un nombre, y opcionalmente una descripción. Se puede generar una descripción con ayuda de la

IA, que toma el nombre del robot.

 **Create new robot in read mail** ✕

Folder

read mail ▾

Assign a name to your robot

bot_test_mail

Can use letters, numbers and spaces (azAZ09_)

Description [✚ Generate description with IA](#)

Robot description

Close Save changes

Una vez se haya creado el robot tendremos la vista en dashboard de los robots, su descripción y las acciones que se pueden realizar (ejecutar, editar, borrar) También podemos dejarlo en modo activo y podremos observar cuántas veces se ejecutó el robot.

Folder read mail

+ New Robot

Description:

will have robots that read mails and can write and send messages within mails.



Robots:

Robot	Description	Actions
<p>bot_test_mail Version: 0 in a few seconds</p> <p>Edit robot</p>	<p>🔗 Start to add moons clicking in the button Edit Robot Trigger:</p>	<p> </p> <p><input type="checkbox"/> Active</p> <p>Runs: 0</p>

Documentation:

Project Title:

Email Automation with Bots: Read and Send Messages

[View document](#)

Table of Project Sections:

Rocketbot © 2024

En la descripción tendremos las “Moons” (comandos que posee el robot) y bajo el nombre del robot tendremos la versión de este (cuantas veces se modificó el robot) y el tiempo de creación.

Robot

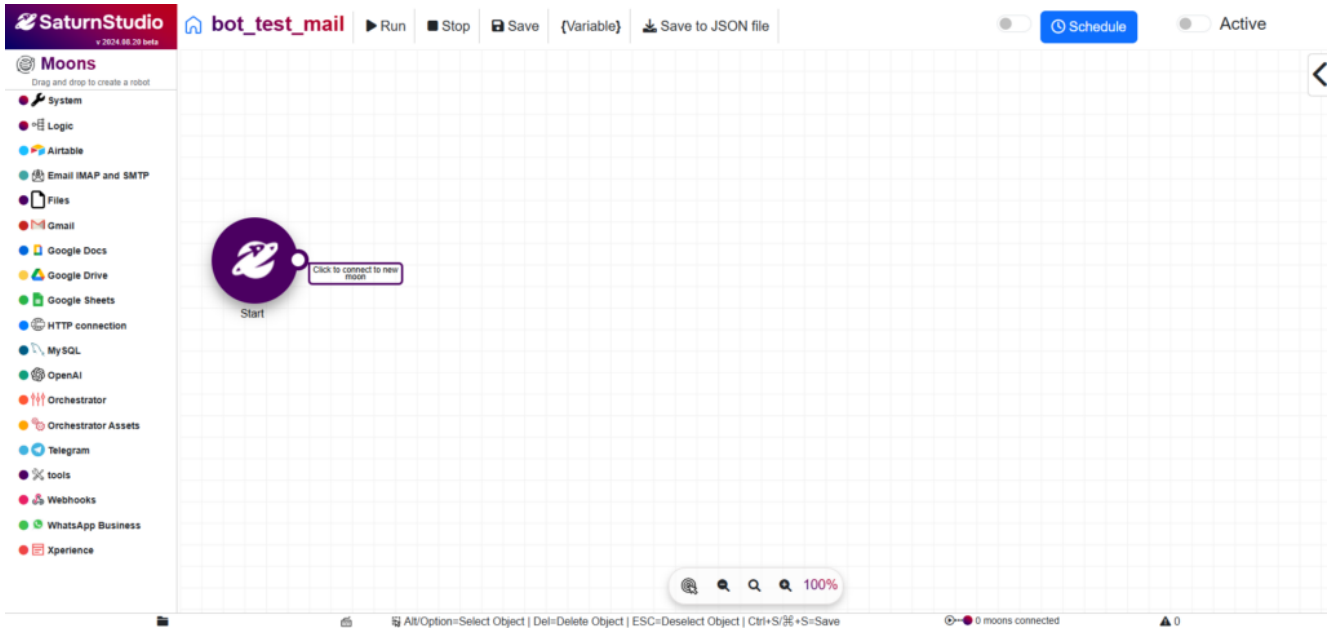
bot_test_mail

Version: 0

4 minutes ago

[Edit robot](#)

Para editar el robot y asignar comandos debe de entrar en la sección de Edit robot, una vez allí podrá asignarles todo los comandos que desee.



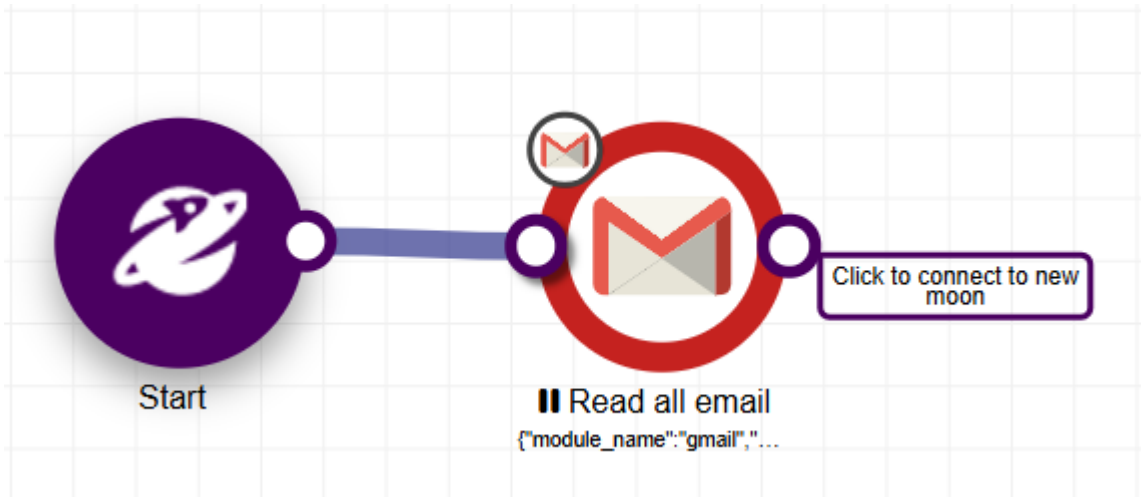
Saturn Studio/Rocketbot – Credentials

Las credenciales son las cuentas de servicio (como Google) que estarán vinculadas a los comandos de los módulos a utilizar (como el módulo de Gmail), modificando o permitiendo obtener la información de la cuenta que se desee usar.

Crear una credencial

Para ello, se debe ir a Saturn Studio y asignar un comando que requiera una credencial.

Un ejemplo sería usar un comando de Gmail.



Allí, dentro del comando, saldrá la opción para crear o seleccionar una credencial.

Read all email

Add a description

? Read all emails from a Gmail account {x} **Test Command**

Credential **?**

mariano.cuello@rocketbot.com ▼ **New**

Select the email credential you want to use

Inbox folder **?** CHAT ▼ **Email status** **?** UNREAD ▼ **Limit** **?** 2

Subject (optional) **?** BUY **Sent by (optional)** **?** test@email.com **Mark as read** **?**

Assign result to variable **?** Select one ▼

Error Handling: Run Robot **Error Handling: Stop all**

Cancel **Ok**

Podemos asignar una nueva credencial en el botón "New".

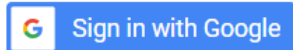


Gmail credentials


Config the email credentials to send and receive emails with the Gmail module

Credential name ?

Put a name that will help you identify the credential in the future





Cancel

 Save credentials

Allí podremos iniciar con Google y asignar un nombre a la credencial.

Una vez hecho esto, en el Dashboard tendremos las otras opciones.

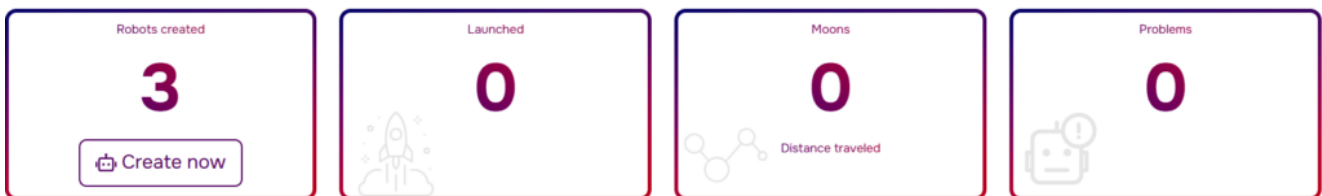
Service	Name	Actions
 Gmail	mariano.cuello@rocketbot.com	 Delete

- **Service:** el servicio que ocupa la credencial.
- **Name:** El nombre que se le asignó para identificar la credencial.
- **Delete:** Borrar la credencial. Cuando se borra la credencial, automáticamente se aplicará el cambio al comando al que se le asignó la credencial.

Saturn Studio/Rocketbot : Dashboard

En el dashboard se podrán ver las estadísticas del usuario:

Reportes generales: Se mostrarán datos sobre los robots lanzados, la distancia recorrida por las “lunas” (correspondiente a los comandos ejecutados por los robots) y los problemas detectados. Se puede crear un robot en la carpeta seleccionada desde aquí (botón **Create now**).



Se pueden visualizar las estadísticas del mes, semana o día corriente.

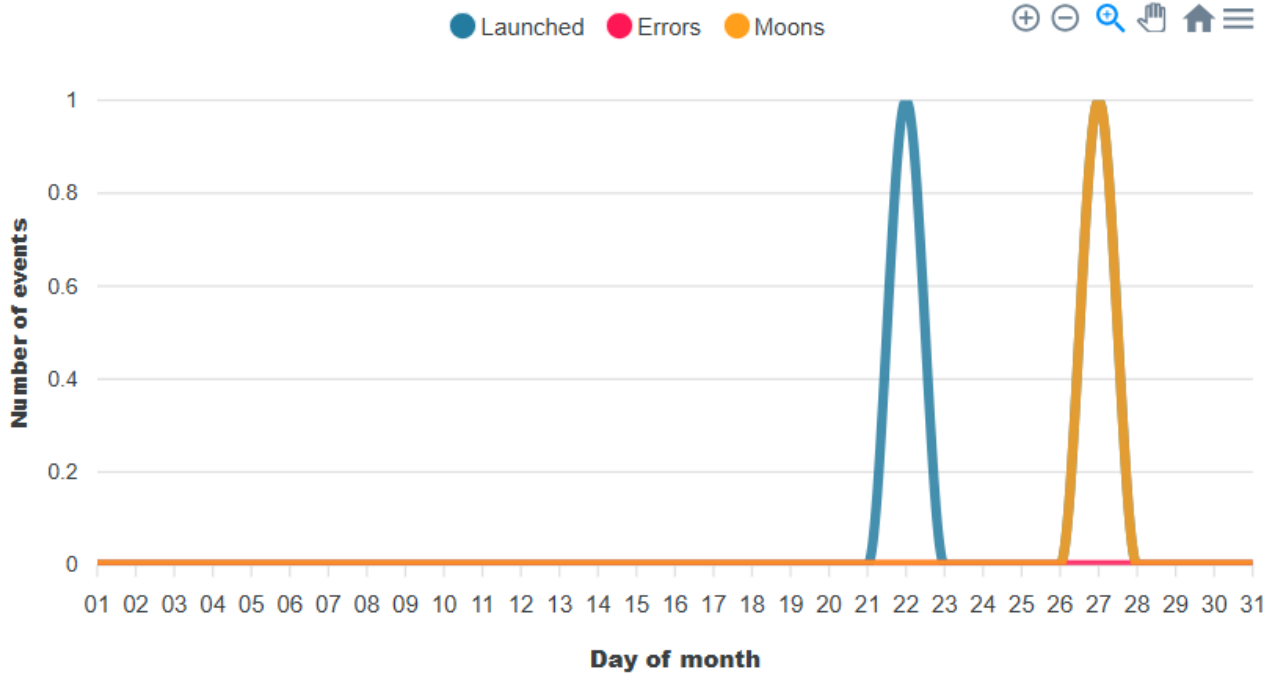


Gráfico de eventos

Se nos mostrará el siguiente gráfico, el cual nos indicará los números de eventos y en qué día del mes ocurrieron.

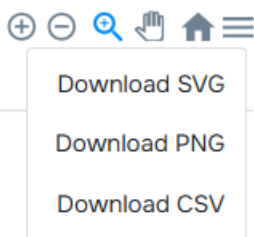
Podremos diferenciar:

- Los robots lanzados.
- Los errores.
- Las Lunas (comandos ejecutados del robot).



Tenemos las opciones del gráfico en su lateral, las cuales nos permiten:

- Hacer zoom.
- Reducir el zoom.
- Seleccionar el zoom.
- Activar el modo panorámico.
- Restaurar el zoom.
- Acceder al menú para descargar el gráfico.



Reporte de los últimos 10 robots ejecutados este mes

Tendremos en el reporte el nombre del robot y el tiempo.

Last 10 robot launches this month

Robot	Time
-------	------

Botón Refresh

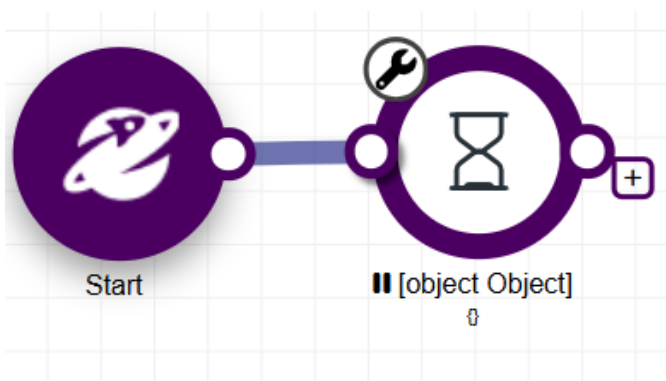
Este botón refrescará o actualizará el reporte del dashboard.



Refresh

Saturn Studio/Rocketbot – Moons

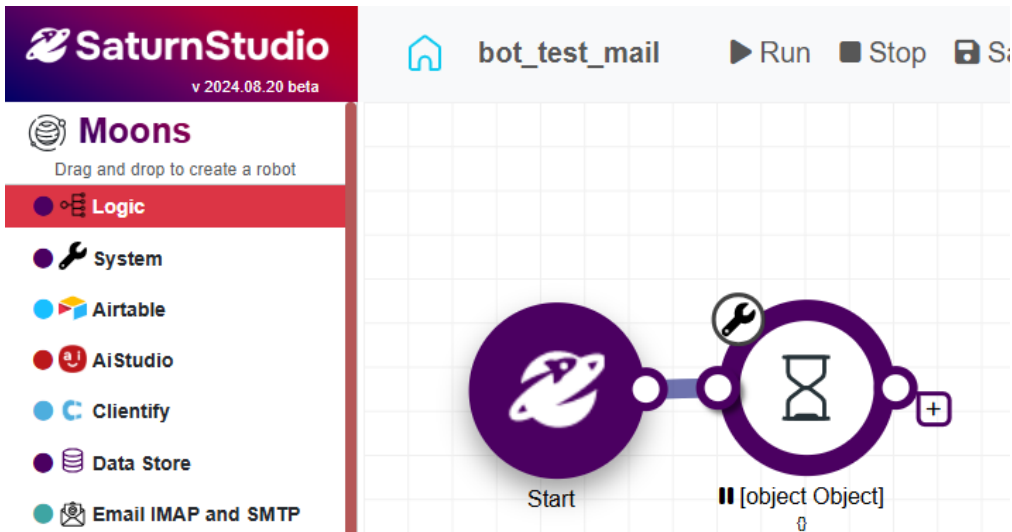
En **Saturn Studio**, cada comando es una **luna (moon)**. Para que los comandos funcionen correctamente, deben estar conectados de forma secuencial: cada comando debe enlazarse al anterior, y el primer comando debe estar conectado al comando inicial (**Start**).



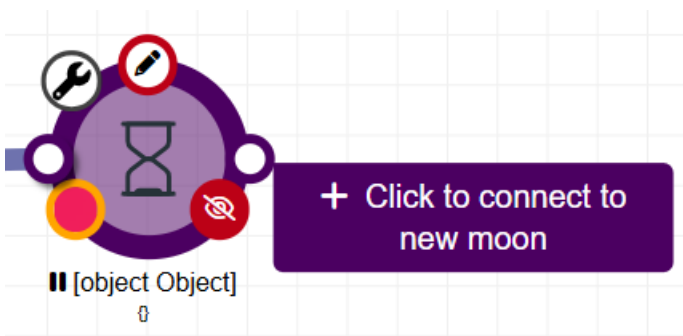
Agregar comandos

Hay dos maneras de agregar comandos al robot, desde el menú

lateral o como continuación de un comando existente.



Además, existe la opción de agregar comandos directamente como continuación de un comando existente. Al posicionar el cursor junto a una *moon*, aparecerá la opción “Click to connect to new moon”.



Al seleccionarla, se desplegará un menú donde podrá buscar o seleccionar el comando deseado. Este menú muestra una breve descripción del módulo y de los comandos disponibles dentro de él.

Search

Search...

Logic

Blocks that allow you to create logic in your automation.

System

Blocks that allow you to interact with the system.

Airtable

[Airtable](#) is an intuitive and powerful platform that combines the functionalities of a spreadsheet with the capabilities of a relational database

AiStudio

Module to interact with AiStudio

Clientify



IF

Evaluate a condition with JS logic



Switch Case

Switch between different cases evaluating the condition with javascript logic



While

Run an cyclic event evaluating the condition with python logic



Break

Break the loop



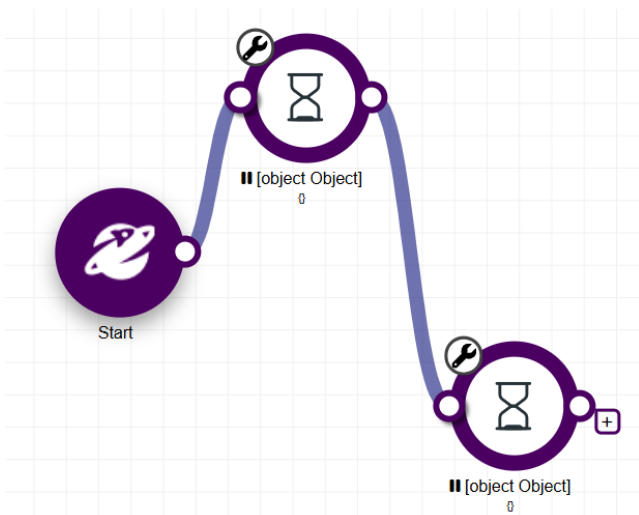
Iterate over elements

Loop through each item in a list or collection

Una vez seleccionado el comando, la nueva *moon* se agregará automáticamente al diagrama, y se abrirá el menú de configuración correspondiente para configurar comando.

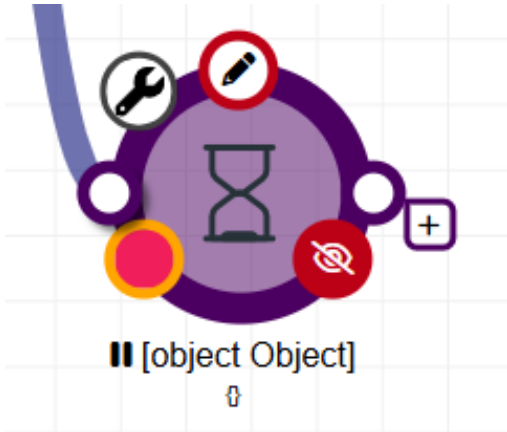
Interfaz

El entorno de **Saturn Studio** está diseñado para ser altamente visual e intuitivo. Los usuarios pueden arrastrar y soltar las *moons* libremente, organizándolas según sus preferencias.



Opciones

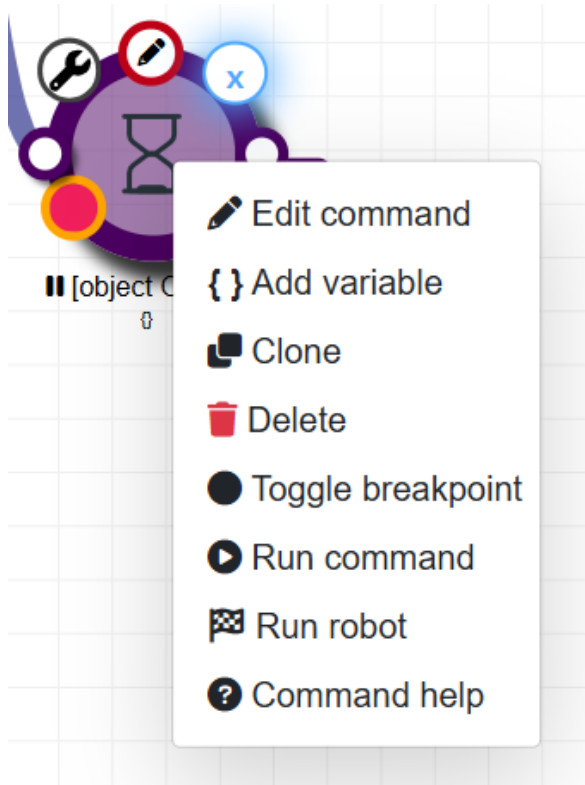
Cuando se posiciona el cursor sobre un comando en el diagrama, se presentan las siguientes opciones rápidas:



- Editar el comando.
- Alternar visibilidad (visible o no).
- Configurar punto de interrupción.

Opciones Avanzadas

Al hacer clic derecho sobre un comando, se accede a opciones avanzadas:



- **Editar comando.**
- **Agregar variable.**
- **Clonar comando.**
- **Eliminar comando.**
- **Alternar punto de interrupción.**
- **Ejecutar comando.**
- **Ejecutar robot.**
- **Acceder a la ayuda del comando.**