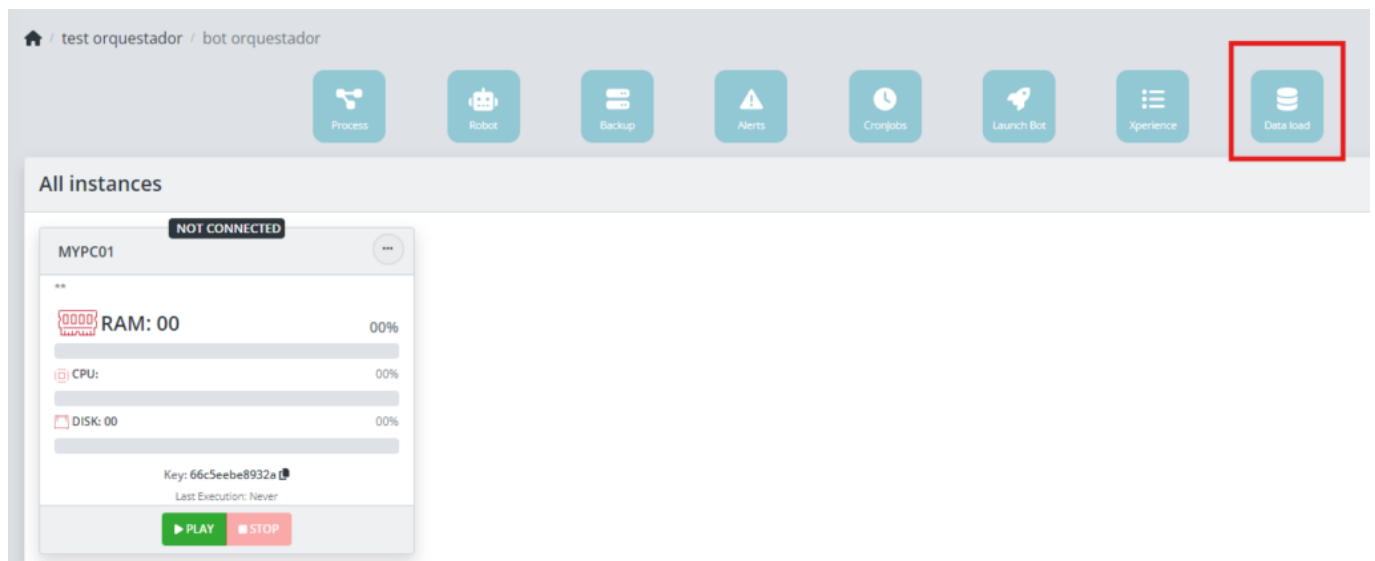


# Orquestador Rocketbot : Data Load

## Data Load

Es un sistema diseñado para crear tareas de proceso (Tasks), dentro de las cuales se añaden transacciones (Transactions) que facilitan la transferencia de información entre robots.

Para utilizar esta función, simplemente selecciona el proceso deseado y accede a la herramienta de proceso "Data Load".



Desde allí, serás redirigido a una lista con todas las tareas de proceso disponibles.

The screenshot shows a table titled 'Process data' with the following columns: Key, Transactions, Progress, Errors, Loaded at, Status, and Priority. The table contains five rows of data. At the bottom, there is a pagination control showing 'First', 'Prev', '1', '2', 'Next', and 'Last', with '1' selected. A dropdown menu shows '5' items.

Key	Transactions	Progress	Errors	Loaded at	Status	Priority
9hwfflysh	5	0%	0%	2024-08-22 11:20:26	Pending	Very Low
4hjt11Hn	0	0%	0%	2024-08-22 11:24:49	Finished	Very Low
PgY9131I	5	0%	0%	2024-08-22 11:36:50	Pending	Very Low
HdBCp3Uk	0	0%	0%	2024-08-22 11:37:19	Pending	Very Low
19sdxkLW	0	0%	0%	2024-08-22 15:33:51	Pending	Very Low

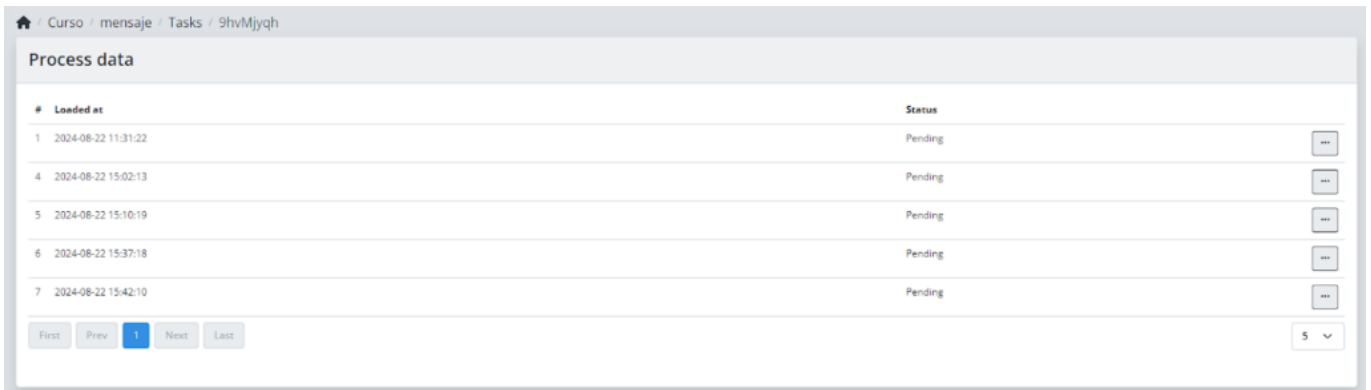
En esta lista encontraremos las siguientes columnas:

- **Key:** Identificador único de la tarea.
- **Transactions:** Número de transacciones asociadas a la tarea.
- **Loaded at:** Fecha y hora en que la tarea fue cargada.
- **Status:** Estado actual de la tarea (Pendiente o Procesada).
- **Priority:** Nivel de prioridad asignado a la tarea, que determinará el orden en que se ejecutará.

Además, se contará con las siguientes herramientas para gestionar las tareas:

- **Details:** Muestra el detalle de la tarea y redirige a una tabla con sus transacciones.
- **Restart:** Cambia el estado de la tarea de 'Procesada' a 'Pendiente'.
- **Finished:** Cambia el estado de la tarea de 'Pendiente' a 'Procesada'.

La vista "Details" nos presenta una tabla con las transacciones de la tarea, donde se muestra la siguiente información:



The screenshot shows a web interface with a breadcrumb trail: 'Curso / mensaje / Tasks / 9hvMjyqh'. Below it is a table titled 'Process data'. The table has two main columns: 'Loaded at' and 'Status'. There are 7 rows of data, all with a status of 'Pending'. Each row has a small menu icon (three dots) on the right. At the bottom of the table, there are navigation buttons: 'First', 'Prev', 'Next', and 'Last'. The 'Next' button is highlighted in blue. To the right of these buttons is a dropdown menu showing the number '5'.

#	Loaded at	Status
1	2024-08-22 11:31:22	Pending
4	2024-08-22 15:02:13	Pending
5	2024-08-22 15:10:19	Pending
6	2024-08-22 15:37:18	Pending
7	2024-08-22 15:42:10	Pending

- **Loaded at:** Fecha y hora de creación de la transacción.
- **Status:** Estado actual de la transacción (Pending, Running, Finished).

Además, disponemos de las siguientes herramientas para gestionar cada transacción:

- **View transaction:** Permite visualizar los detalles de la transacción abriendo un modal con la información enviada.
- **Restart:** Cambia el estado de la transacción de 'Procesada' a 'Pendiente'.
- **Finished:** Cambia el estado de la transacción de 'Pendiente' a 'Procesada'.

## Creación de una tarea y asignación de transacciones

Para crear una tarea y asignarle transacciones, se debe utilizar el módulo "Orchestrator Framework" de Rocketbot Studio.

Utilizando el comando "Crate Task" o "Crear Tarea".



Studio v.2024.05.28

Nombre del Robot ?

bot\_orquestador\_tarea



Modo en vivo  ?

Filtrar comando

Orchestrator Framework ∨

Login NOC +

Obtener procesos +

Obtener Tareas +

Crear Tarea +

Establecer Prioridad +

Agregar Transacción +

Agregar multiples Transacciones

Obtener transacciones sin proces

Establecer estado +

Enviar alerta +

Enviar log personalizado +

Detener Framework +

¿Debe detenerse el Framework?

Debemos ingresar el Token del proceso al que se le asignará la tarea, y es recomendable guardar la key de la tarea en una variable, como por ejemplo "Task".

Robot - data-bot1

## Create Task

Adds a new task Help

Properties Config About

Process Token

Task Key

Assign result to a Variable

Variable name without []

Accept Cancel

## Add transaction

En este paso, se ingresará nuevamente el Token del proceso, y en el campo "task key", se colocará la variable que contiene la key de la tarea creada, permitiendo comenzar a añadir transacciones a la misma.

Además, se puede configurar la información que se desea enviar, y opcionalmente agregar las cabeceras correspondientes a cada dato.

Robot - data-bot1

## Add Transaction

Add one new transaction Help

Properties Config About

Process Token

Task Key

Transaction

Assign result to a Variable

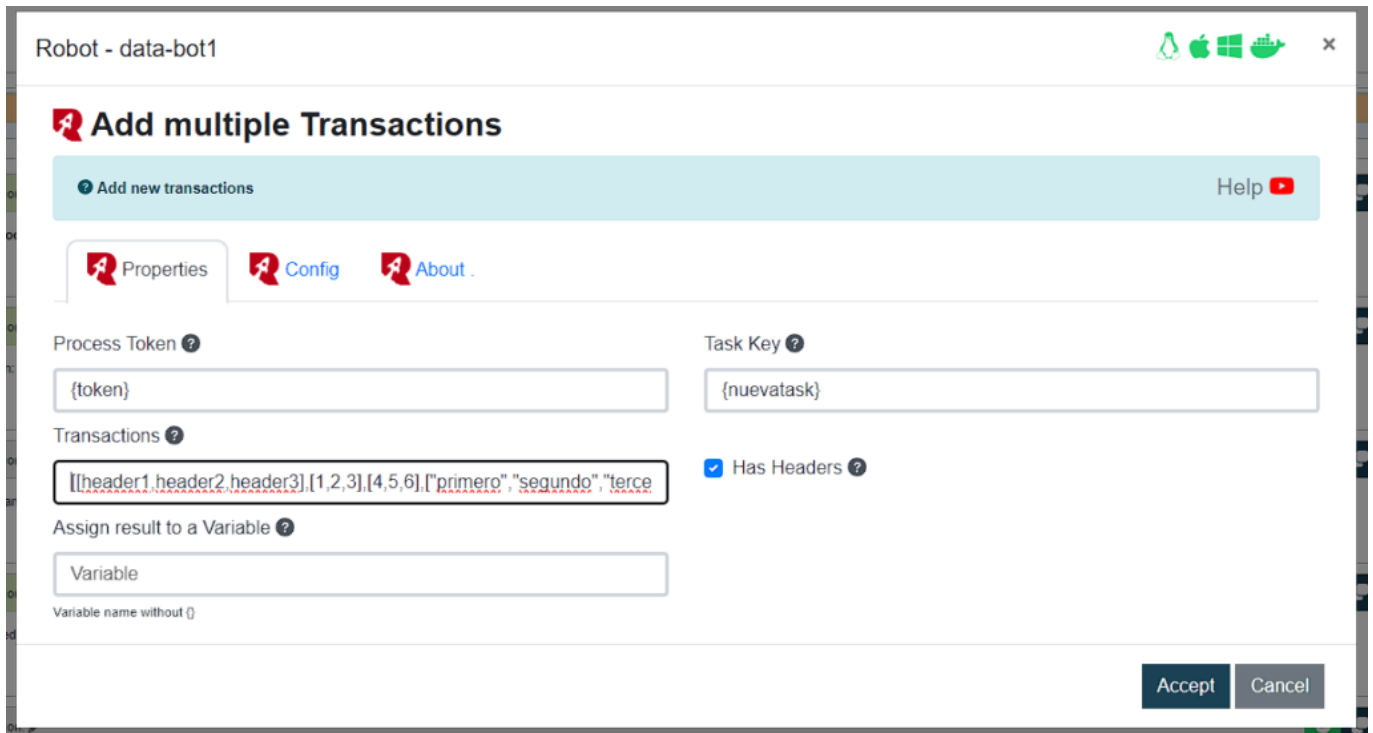
Variable name without []

Has Headers

Accept Cancel

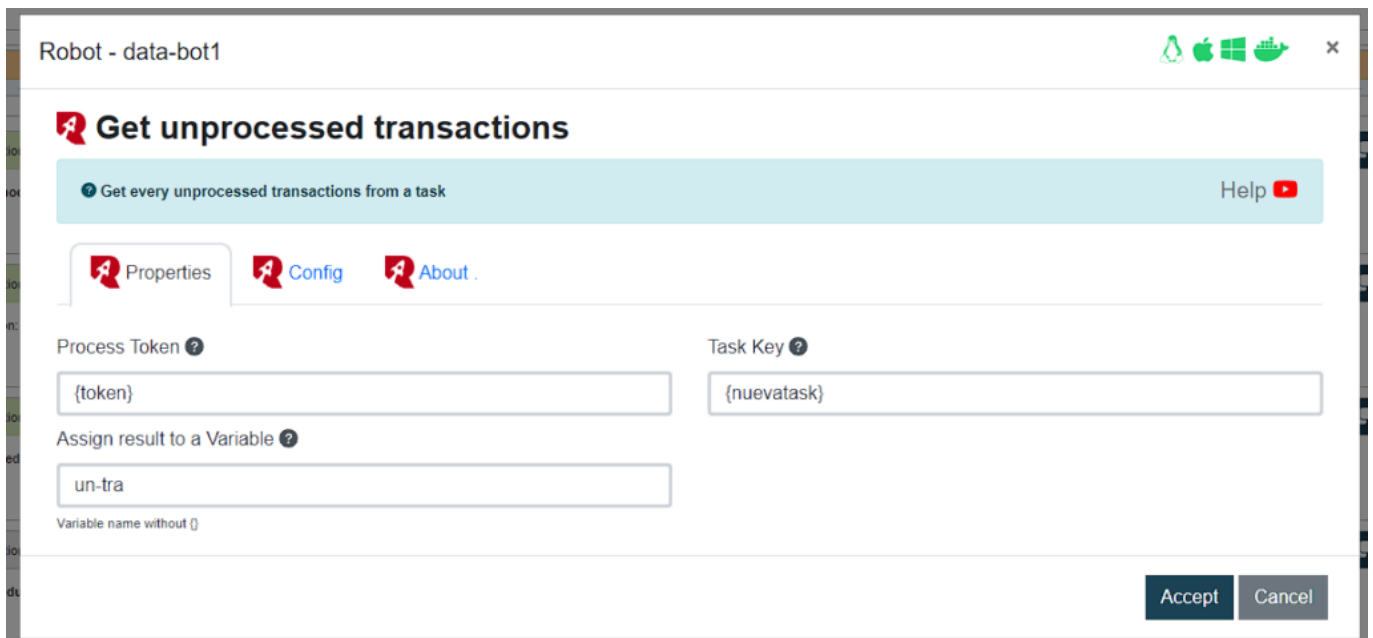
## Add multiple transactions

Para asignar múltiples listas de datos.



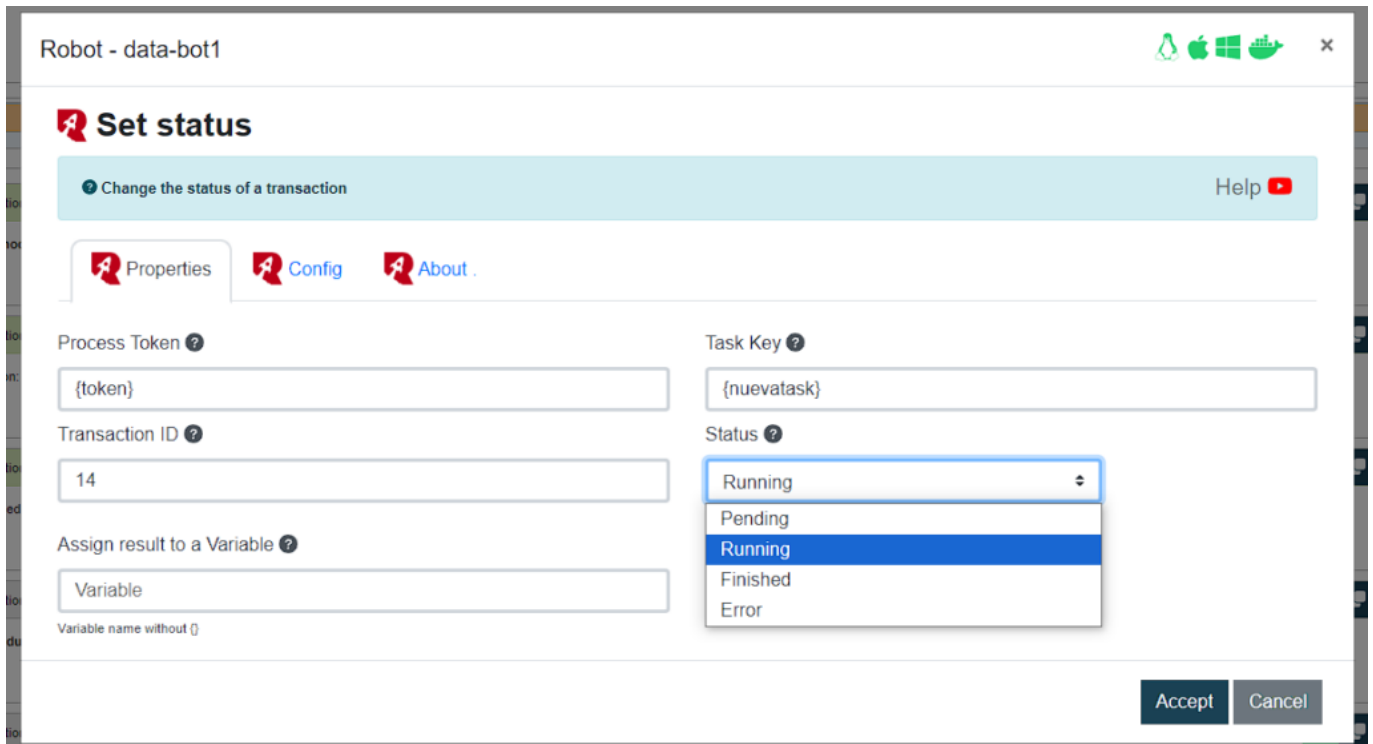
## Get Unprocessed Transactions

A continuación, al emplear un robot receptor para recibir la información enviada, es posible procesar los datos y almacenarlos utilizando el comando **"Get Unprocessed Transactions,"** el cual recupera la primera transacción no procesada de la tarea solicitada.



## Set Status

Asimismo, podemos asignar un estado a una transacción para indicar si se encuentra pendiente, en uso o procesada



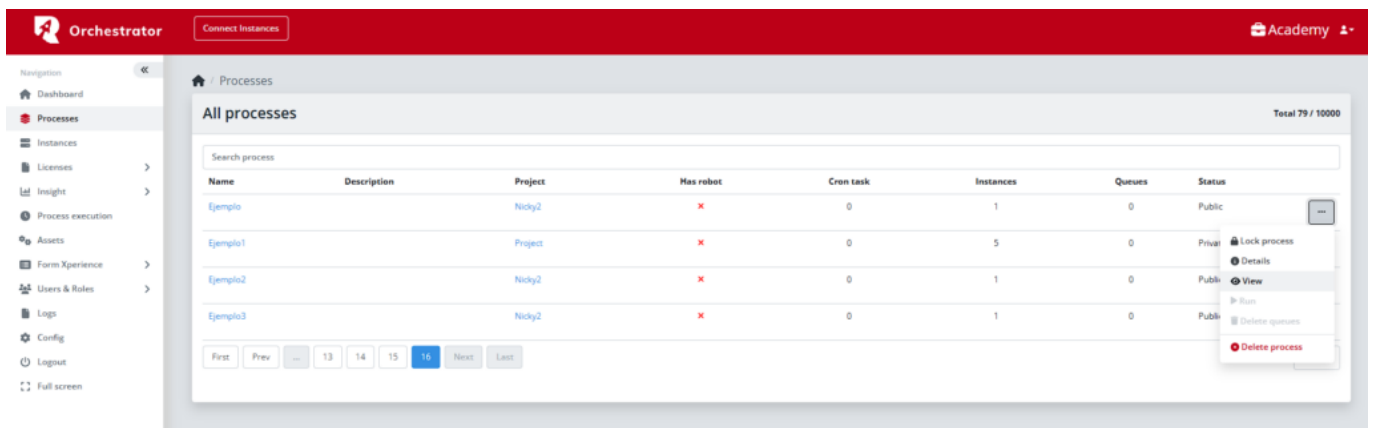
## Orquestador Rocketbot : Instancias

### Definición de instancia

Una instancia representa el enlace entre el ordenador del usuario y el servidor del Orquestador. Su función es gestionar la ejecución de los robots, ya sea de forma manual o automatizada.

### Procedimiento para asignar una instancia

Para asignar una instancia, es indispensable contar previamente con un proceso dentro de un proyecto seleccionado y acceder a su menú de configuración ("View" del proceso).

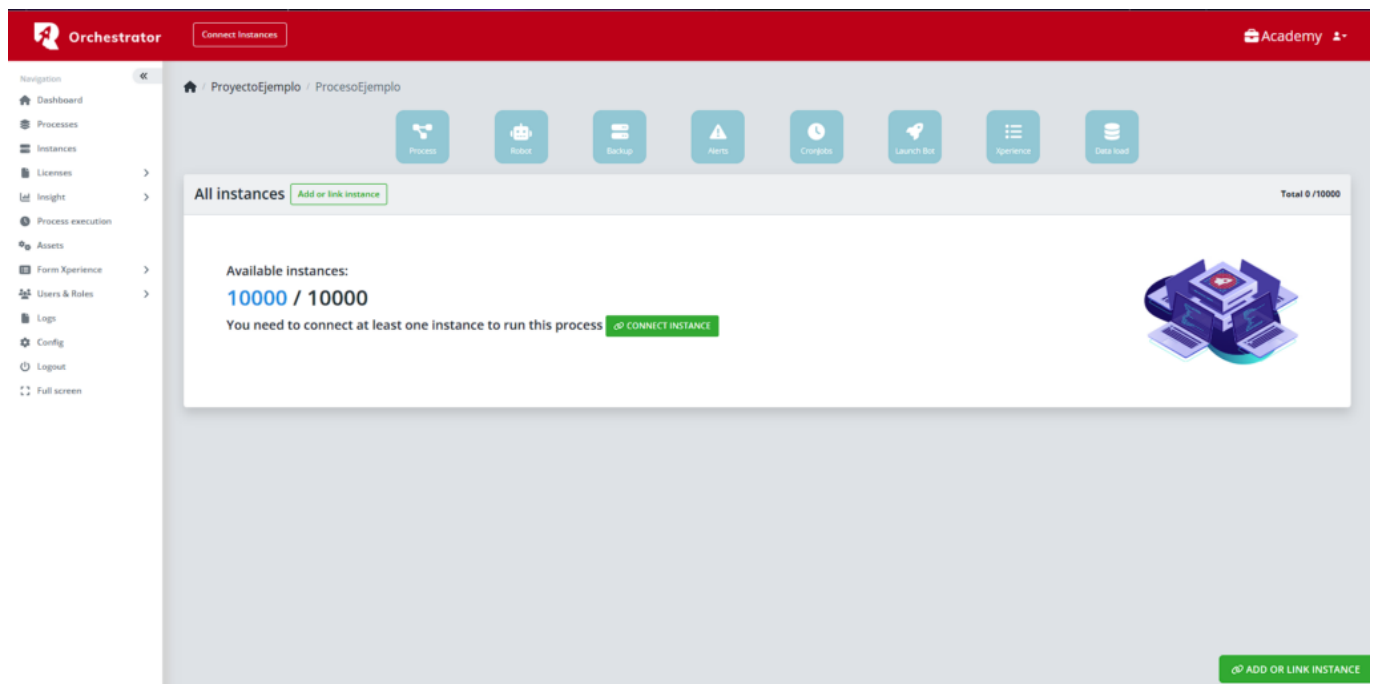


Si aún no has creado tu primer proyecto o proceso, se recomienda consultar el

siguiente documento: [¿Cómo crear mi primer proceso? .](#)

## Crear o vincular una instancia

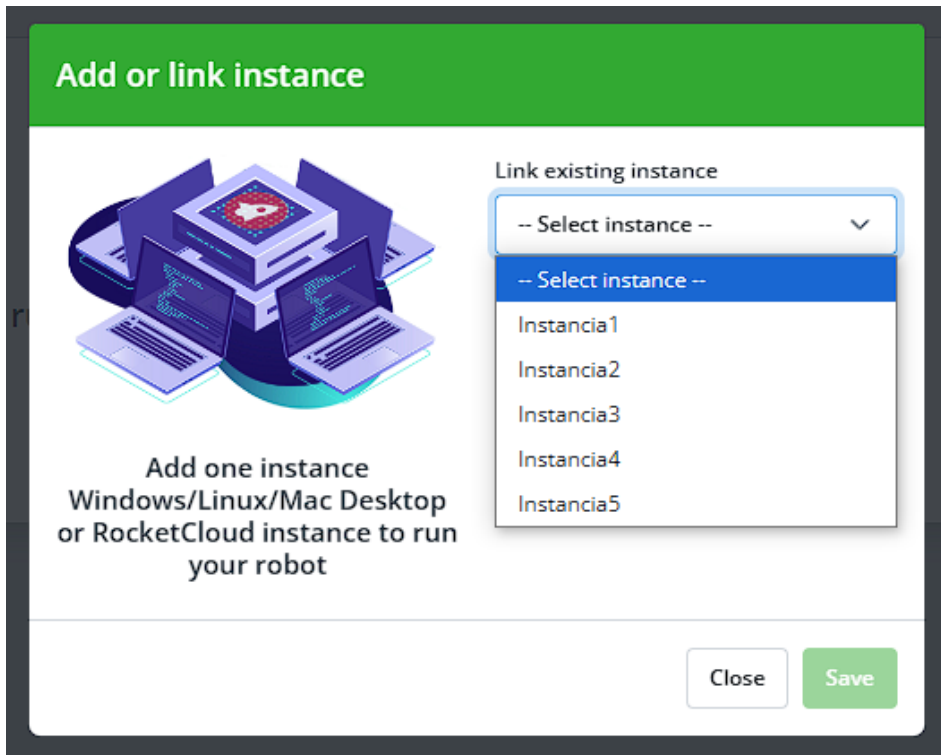
Una vez dentro del “view” del proceso, se nos mostrará esta vista:



Para crear o seleccionar una instancia, hacemos clic en el botón **[Add or Link Instance]**, aunque también está el botón **[Connect Instance]**. Y se nos abrirá un modal, en el cual tendremos dos opciones.

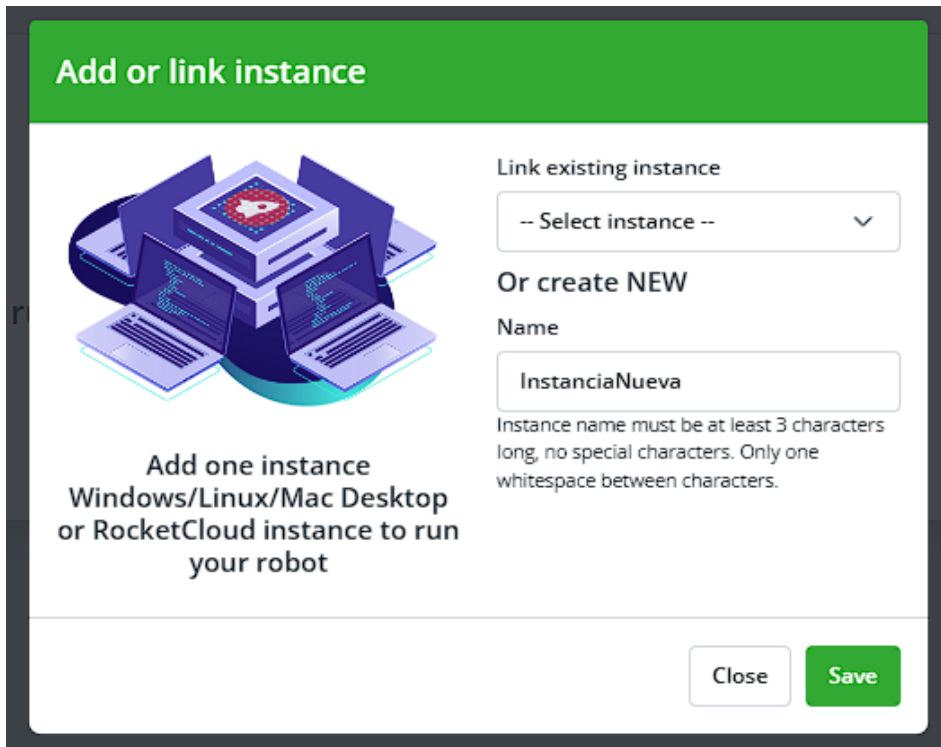
### Vincular una instancia existente

“**Link existing instance**” es un *select* donde podremos ver una lista de todas las instancias que poseemos, y deberemos seleccionar una.



## Nueva Instancia

“Create New” es un *input* al cual le proporcionaremos un nombre para la instancia (un nombre que no exista aún), y al darle a “Save”, crearemos nuestra nueva instancia.



## Conectar instancia al cliente

De forma predeterminada, luego de ponerle nombre a la instancia, no estará vinculada con el equipo. Por lo tanto, hay que hacer clic en el menú de la instancia (los tres puntos) y allí seleccionar el botón **[Connect to Client]**.



## 2. Ejecutar el cliente

Dentro del archivo comprimido, se debe ejecutar el archivo **"OrchestratorClient.exe"**, el cual abrirá una ventana de acceso. En esta ventana, debemos ingresar la URL del servidor (copiar el **"server URL"** del punto 4).

The image shows two side-by-side screenshots. On the left is the 'Rocketbot Orchestrator Client' window. It features the 'ROCKETBOT' logo in red, a settings gear icon, and a text input field for 'Orchestrator URL' containing 'https://test-noc.myrb.io'. Below it is a 'Connection mode' dropdown menu with the text '-- Select an option --'. At the bottom is a grey 'CONNECT' button. On the right is a 'Your data' panel with a blue header. It contains three data entries: 'Server Url: https://test-noc.myrb.io', 'User apiKey: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ' (with a long redacted string below it), and 'Instance key: 6720ee3ac9387'. A red arrow points from the 'Server Url' field in the data panel to the 'Orchestrator URL' field in the client window. The number '4' is visible in the top right corner of the data panel area.

## 3. Configuración de credenciales para acceder al servidor

Existen dos opciones para configurar las credenciales:

Rocketbot Orchestrator Client

**ROCKETBOT**

Orchestrator URL:  
https://noc.test

Connection mode:  
Configuration File (noc.ini)

-- Select an option --

Credentials

API KEY

Configuration File (noc.ini)

**CONNECT**

User apiKey:

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ

ihQX-ItwlnM

Instance key: 6720ee3ac9387

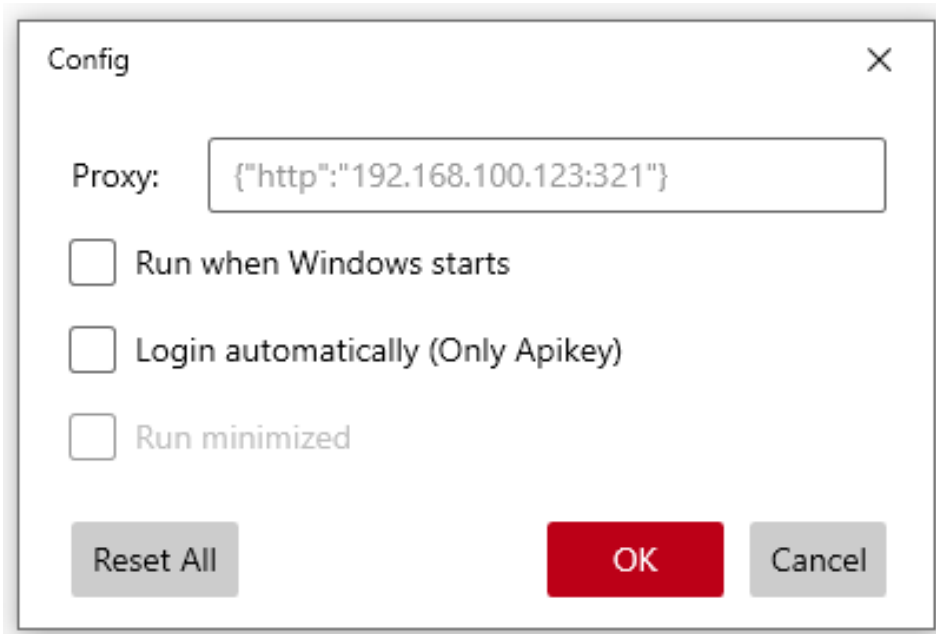
- **“Credentials”**: Debemos ingresar el **“email”** y la **“password”** de nuestra cuenta con la que normalmente accedemos al Orquestador.
- **“API KEY”**: Esta llave única se puede obtener en la información proporcionada en el punto 4 o en la información de nuestro perfil. Es la opción más segura, ya que no se utiliza la contraseña.  
**Importante: La “API KEY” tiene un periodo de vencimiento de 2 años; luego de este tiempo, será necesario generar una nueva por razones de seguridad.**
- **“File noc.ini”**: Opción para compatibilidad con versiones anteriores mediante el archivo **noc.ini** (Ver punto 6).

#### 4. Obtención de datos relevantes para configuración rápida



## 5. Configuración

En la parte superior derecha de la ventana, se encuentra el ícono para aplicar configuraciones secundarias:



Config

Proxy: {"http": "192.168.100.123:321"}

Run when Windows starts

Login automatically (Only Apikey)

Run minimized

Reset All OK Cancel

- **“Proxy”**: En caso de contar con un filtro de seguridad a la salida de internet, debe configurarse para habilitar la conexión entre el equipo y la instancia del servidor deseado. Se debe ingresar el protocolo **http**, seguido del pin de salida y el puerto utilizado.
- **“Run when Windows starts”**: Si está habilitado, el cliente se ejecutará automáticamente al iniciar Windows.
- **“Login automatically (Only API KEY)”**: Si está habilitada y se utiliza la opción **“API KEY”**, el cliente iniciará sesión automáticamente al abrirse.
- **“Run minimized”**: Si está habilitado, el cliente se ejecutará minimizado.
- **“Reset ALL”**: Restablece la configuración original.

## 6. Método File .ini

Cuando se utiliza este método, no se aplican las configuraciones del punto 5 y 8; las configuraciones se cargan directamente desde el archivo **.ini**.

```
noc: Bloc de notas
Archivo Edición Formato Ver Ayuda
[[USER]
# user access
user=
password=
#or API Key
apiKey=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwczpcL1wvdGVz
# Instance ID
key=6720ee3ac9387
[[NOC]
# you Url Orchestrator
# https://devnoc.myrb.io
# https://roc.myrb.io/s1
# https://roc.myrb.io/s2
# https://roc.myrb.io/s4
# https://roc.myrb.io/dev
server=https://test-noc.myrb.io
proxy =
#Send logs to Orchestrator?
logs = true
[[ROCKETBOT]
# Rocketbot bin
path = D:\Rocketbot\Rocketbot20201230a4\rocketbot.exe
# No debug on Rocketbot?
nodebug = false
#Screenshot available ?
screenshot = true
```

- **“User Access”**: Primera opción de acceso. Debemos ingresar el **“email”** y la **“password”** de nuestra cuenta con la que normalmente ingresamos al Orquestador.
- **“Or API Key”**: Segunda opción de acceso. Es la **API KEY** del usuario, la cual se obtiene en el punto 4 o en la información del perfil. Es la opción más privada, ya que no se utiliza la contraseña.
- **“Instance ID”**: La **“Key”** (llave única) de la instancia, que se obtiene en el punto 4 (instance key) o en la vista de la instancia (key).
- **“Your URL Orchestrator”**: Debemos ingresar la URL del servidor (copiar el **“server URL”** del punto 4).
- **“Proxy”**: Configuración del filtro de seguridad, tal como se explicó en el punto 5.
- **“Logs”**: Cuando está habilitado, permite enviar un registro de los comandos ejecutados por el robot. (Ver **“getlogs”** en herramientas de instancias).
- **“Path”**: Dirección del ejecutable de **Rocketbot Studio** en el equipo.
- **“Nodebug”**: Habilita (true) / deshabilita (false) los registros de

consola de **Studio**.

- **“Screenshot”**: Permite capturar la pantalla durante la ejecución del robot, para verificar si los programas intervinientes cumplieron su función y se cerraron correctamente. (Ver **screenshot** en herramientas de instancias).

## 7. Conectar instancia

Una vez configurado todo, procedemos a conectar la instancia.



## 8. Configuración de instancia y opciones adicionales

Select instance

-- Select an instance --

Allow take screenshot

Send logs to Orchestrator

Enable Studio nodebug

Rocketbot path:

C:\rocketbot\rocketbot.exe

Extra arguments:

e.g: --install\_modules --update\_drivers

Save & Continue

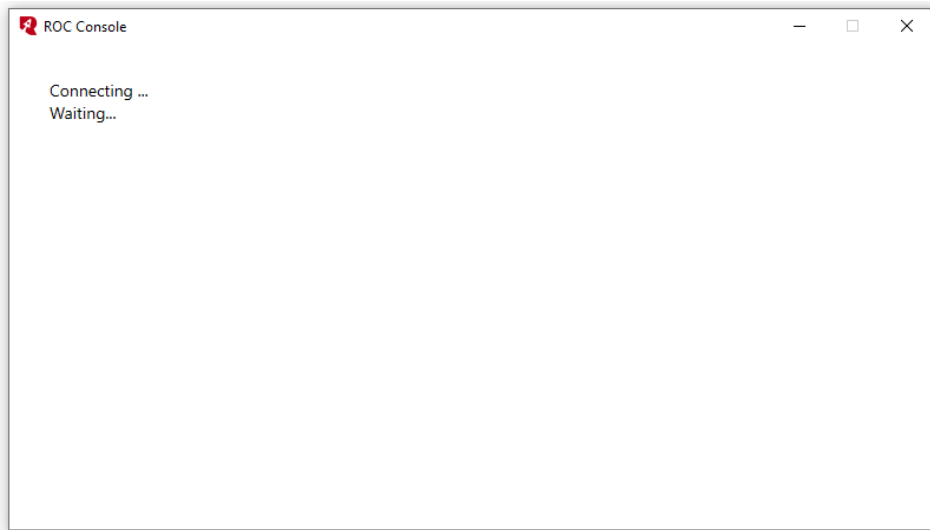
- **“Select instance”**: Se abrirá un *select* con la lista de nuestras instancias para elegir la que deseamos vincular.
- **“Allow take screenshot”**: Cuando está habilitado, permite capturar la pantalla durante la ejecución del robot, para verificar si los programas intervinientes cumplieron su función y se cerraron correctamente. ([Ver \*\*screenshot\*\* en herramientas de instancias](#)).
- **“Send logs to Orchestrator”**: Cuando está habilitado, permite enviar un registro de los comandos ejecutados por el robot. ([Ver \*\*“getlogs”\*\* en herramientas de instancias](#)).
- **“Enable Studio nodebug”**: Habilita/deshabilita los registros de consola de **Studio**.
- **“Rocketbot path”**: Dirección del ejecutable de **Rocketbot Studio** en el

equipo.

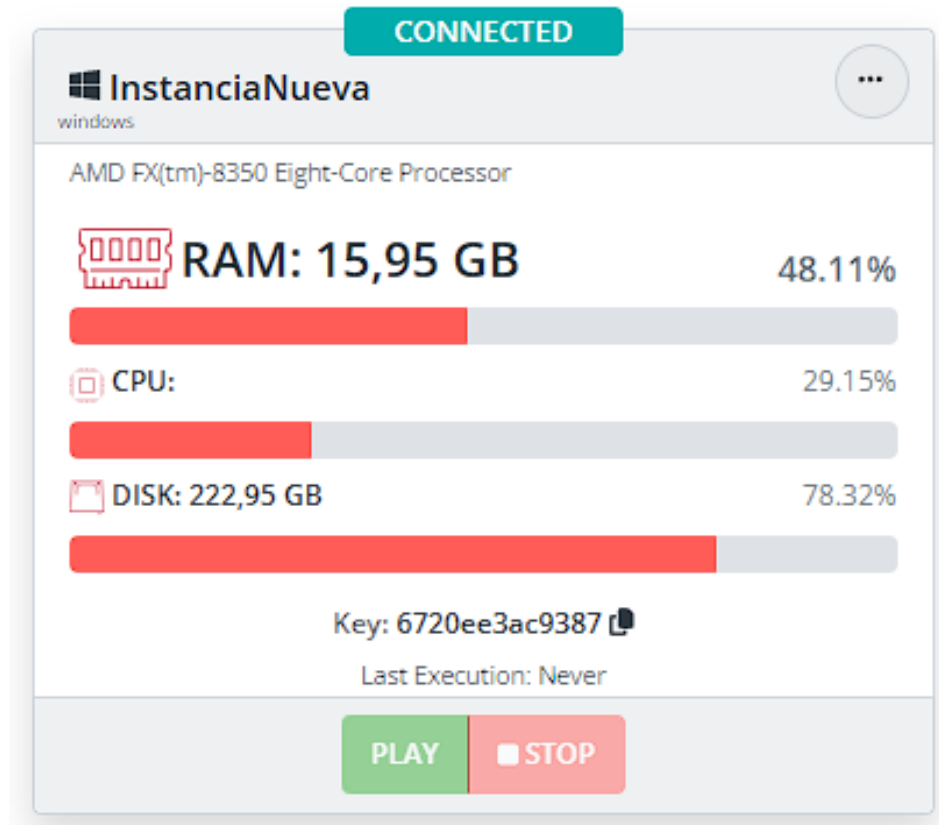
- **“Extra arguments”**: Permite instalar módulos o actualizar drivers.

## 9. Conexión de la instancia

Se abrirá la consola y, después de unos segundos, la instancia se conectará.



Podremos ver en la vista del proceso que nuestra instancia se conectó correctamente.

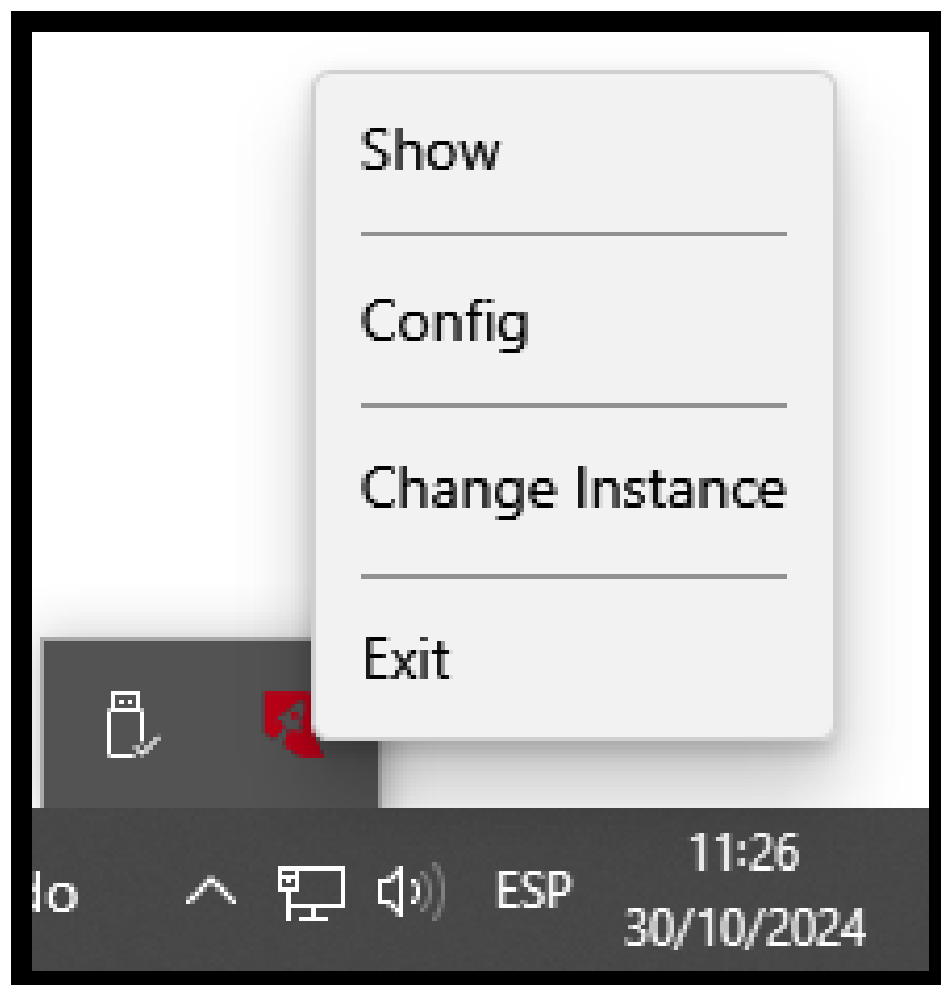


## 10. Interacción con la consola

Una vez que la consola esté en ejecución, se podrá minimizar y se “ocultará”.

Si deseamos volver a interactuar con ella, debemos ir a “iconos ocultos”.

**IMPORTANTE:** *No cerrar la consola.*



- “Show”: Abrir consola.
- “Config”: Abrir configuración del punto 5.
- “Change instance”: Cambiar instancia.
- “Exit”: Cerrar cliente Orquestador.

## Lista de instancias

Para acceder a la lista de instancias, debemos dirigirnos a la pestaña “Instances”, donde se desplegará la tabla con todas nuestras instancias.

Name	Key	Status	Processes
Instancia1	671fa59ce2627	Stopped	2
Instancia2	671fa5a4b73fc	Stopped	1
Instancia3	671fa5ac360b5	Not connected	1
Instancia4	671fa5b2edaa0	Not connected	1
Instancia5	671fa5bd3dcfd	Not connected	1

Dentro de este listado, podremos visualizar lo siguiente:

- El **nombre** de la instancia.
- Su **llave única** (Key).
- Su **estado** (Status).
- Los **procesos** en los cuales la instancia está involucrada.

En uno de los extremos de la tabla, encontraremos un menú desplegable (...) con diversas opciones para gestionar las instancias.

## Estado de una instancia

El estado de cada instancia puede ser uno de los tres siguientes:

### Status

Connected

Stopped

Not connected

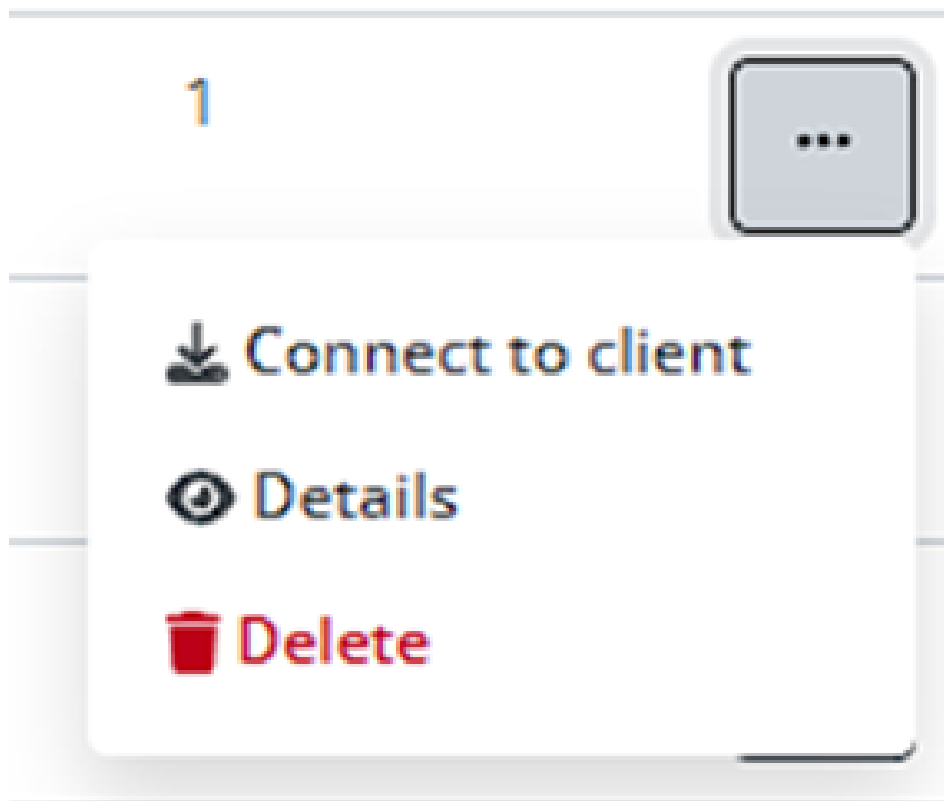
- **“Connected”**: (Conectado) Este estado indica que el proceso de configuración se completó exitosamente y que el puente de conexión entre la instancia y el equipo (Client) está funcionando correctamente. En este estado, podremos ver los datos en tiempo real de nuestro equipo.
- **“Stopped”**: (Interrumpido) Este estado muestra que el proceso de configuración fue exitoso, pero el puente de conexión entre la instancia y el equipo (Client) está cerrado o sin conexión. En este caso, solo podremos ver los datos registrados en la última conexión.

- **“Not Connected”**: (Desconectado) Este estado se muestra cuando el proceso de configuración no se completó correctamente. La instancia fue creada, pero nunca se estableció una conexión con un equipo (Client). Por lo tanto, no se pueden visualizar datos, ya que la instancia nunca estuvo vinculada a un equipo.

## Menú desplegable


Desde el menú desplegable (...) de cada instancia, podremos realizar las siguientes acciones:

## Processes




- **“Connect to client”**: [Conectar la instancia al cliente.](#)
- **“Details”**: (Detalles de la instancia) Al seleccionar esta opción, se abrirá un modal con información detallada sobre la instancia.


**Instance**


**Instancia1** 


**Connected**

AMD FX(tm)-8350 Eight-Core Processor

 **RAM: 15.95 GB** 48.4%

 **CPU:** 19.5%

 **DISK: 222.95 GB** 77%

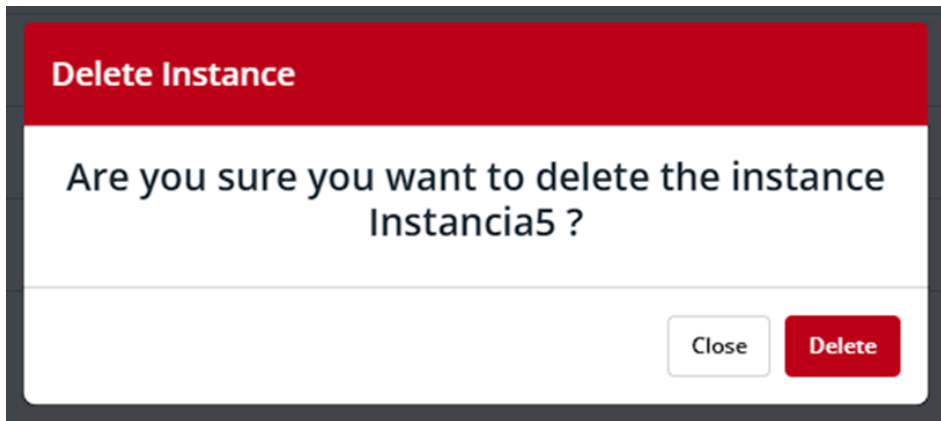
Key: 671fa59ce2627 

Process Name	Robot Name
Ejemplo1	
Ejemplo	
Ejemplo2	
Ejemplo3	

**Close**

- Aquí podremos encontrar:
  - Su **nombre**.
  - El **sistema operativo**.
  - Los **datos del equipo**:
    - Procesador
    - RAM
    - CPU
    - Disco
  - **Llave única (Key)**.
  - La **lista de los procesos** en los que la instancia interviene.
- **“Delete”**: (Eliminar) Elimina la instancia seleccionada. Por motivos de seguridad, se abrirá un modal para confirmar la eliminación de la

instancia.



**IMPORTANTE:** Si se elimina una instancia, se desvinculará automáticamente de todos los procesos en los cuales esté involucrada.

### **Herramientas de una instancia**

Si deseamos conocer las herramientas disponibles para cada instancia, es recomendable consultar el siguiente documento: [Herramientas de Instancia](#).

---

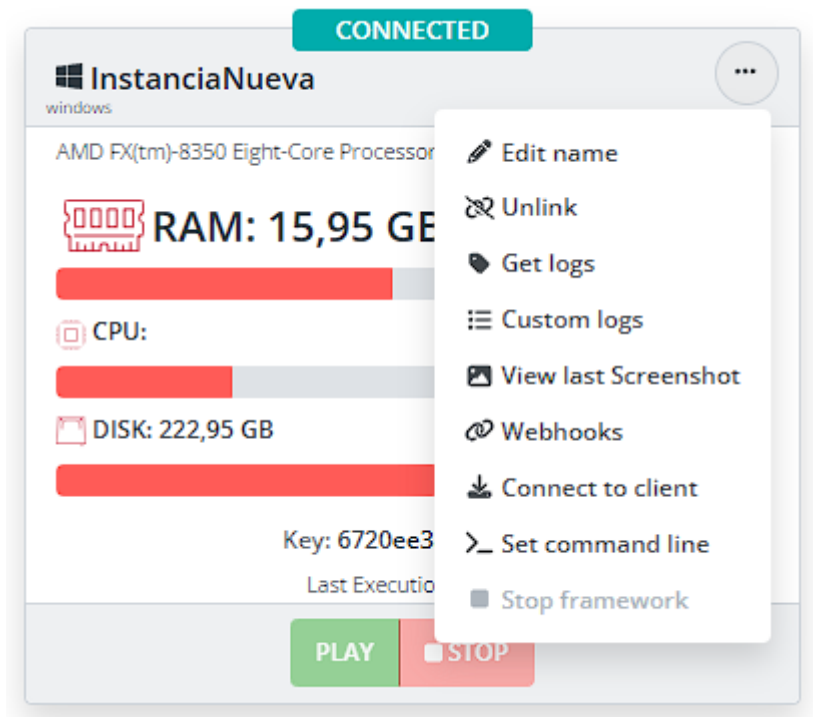
## **[Orquestador Rocketbot : Herramientas de una Instancia](#)**

### **Instancias**

Para la asignación, creación y más detalles sobre las instancias, se recomienda leer el siguiente documento: [Detalles sobre Instancias](#).

### **Herramientas de instancias**

### **View last Screenshot**



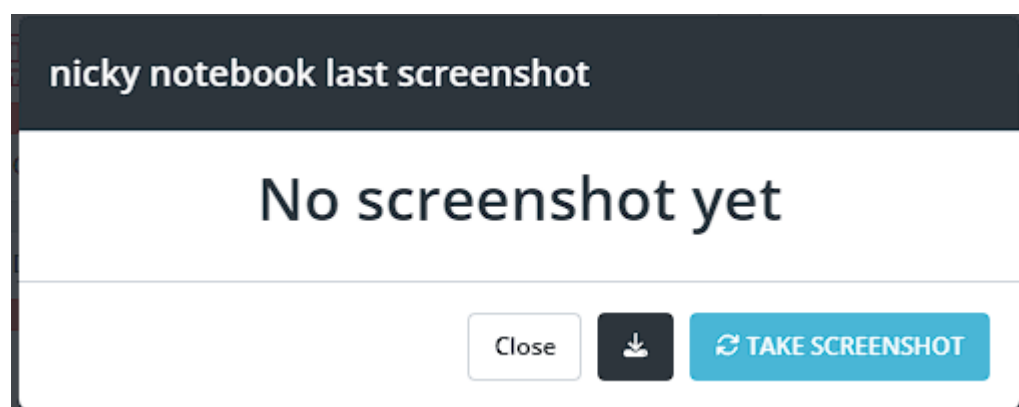
## Captura de pantalla

Para poder utilizar esta herramienta se debe configurar en el archivo “.ini” de nuestro cliente orquestador.

```
#Screenshot available ?  
screenshot = true
```

Una vez configurada podremos acceder a ella desde las herramientas de la instancia.

Si no tenemos ninguna captura de pantalla previa, nos saldrá el siguiente mensaje:

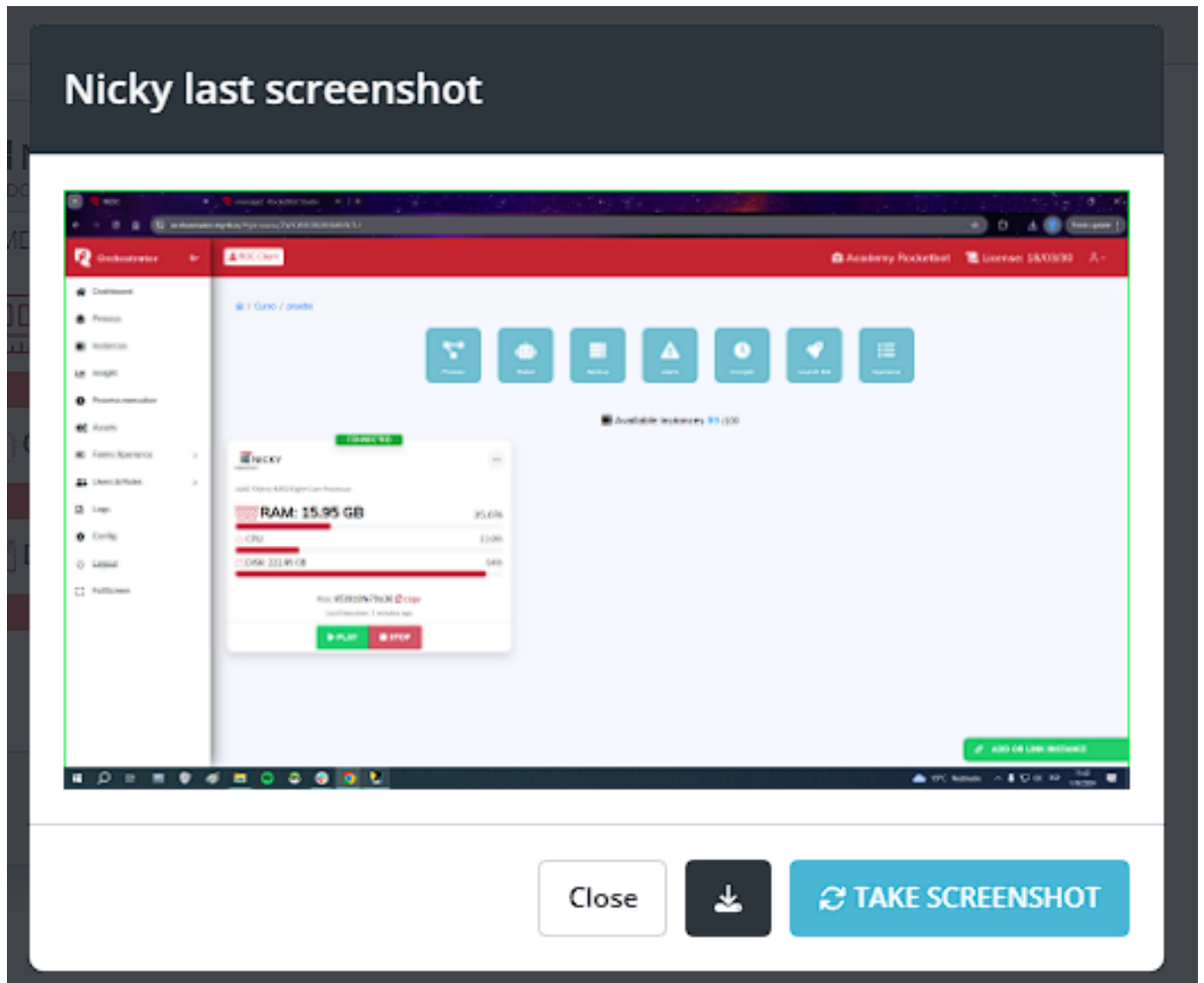


Para mandarle una petición al orquestador de una captura de pantalla nueva se debe seleccionar el botón “REFRESH”.

Dicho procedimiento no es automático y puede tardar algunos segundos.

Si la instancia está conectada al cliente orquestador obtendrá una captura de

pantalla y la mostrara en el mismo modal. Pero si dicha instancia no está conectada a ningún dispositivo/cliente orquestador, no tomara la captura.



Dicha captura de pantalla es general de la instancia y no del proceso. Por ende si tengo la misma instancia en otro proceso, se visualizara la misma captura de pantalla. Una vez que tengamos la captura, podremos descargarla con el icono que aparece en pantalla.

También podremos solicitar una nueva captura haciendo click en "TAKE SCREENSHOT" nuevamente.

## Webhooks

Son llamadas a APIs que realiza el orquestador bajo 3 condiciones siempre y cuando estén habilitadas y configuradas:

**Instance webhook**

**Webhook by play status**

 Enabled  
 Method  
  
 URL:  
  
 Headers:  
  
 Data:

**Webhook by stop status**

 Enabled  
 Method  
  
 URL:  
  
 Headers:  
  
 Data:

**Webhook by custom log**

 Enabled  
 Method  
  
 URL:  
  
 Headers:  
  
 Data:

The following variables can be used in the request  
{{instance\_key}} - {{instance\_name}} - {{process\_token}} - {{process\_name}}

1- Realiza la consulta a la API cuando se ejecuta un robot (No cuando inicia la instancia, sino recién cuando el orquestador encuentra el robot y lo ejecuta) (Cuando la barra de la instancia se pasa a estado amarilla).

Ej: Enviar robot en ejecución a una API para crear un listado de varios robots y sus instancias responsables.

2- Realiza la consulta a la API cuando se termina de ejecutar un robot.  
 Ej: Enviar robots a una API para crear un listado de robots ya ejecutados/procesados.

3- Realiza la consulta cuando llega un mensaje personalizado ("custom log") asociado a dicho proceso.

Ej: Enviar los mensajes personalizados que obtenemos en nuestro proceso a una API.

Configuración:

- "Enabled": Habilitar/Deshabilitar Webhook.
- "Method": Seleccionar el método de envío de información:
- "Get" (Se inhabilita el cuadro "Data", ya que se debe proporcionar la información en la URL)
- "Post"
- "URL": Se debe proporcionar la URL de la API a la cual se va a consultar.
- "Headers": Configuración de las cabeceras de dicha API.
- "Data": Información la cual se enviara a la API, esta debe enviarse en

formato Json. Algunas variables que se pueden usar: `{{instance_key}}`  
`/{{instance_name}}/{{process_token}}/{{process_name}}/{{data}}`

## Get Log

```
#Send logs to Orchestrator?  
logs = true
```

### Historial de comandos del robot:

Esta sección permite acceder al registro completo de todos los comandos que han sido enviados al robot, ya sea que estén en proceso de ejecución o que ya se hayan completado.

### Estado de la conexión:

- Sin conexión:

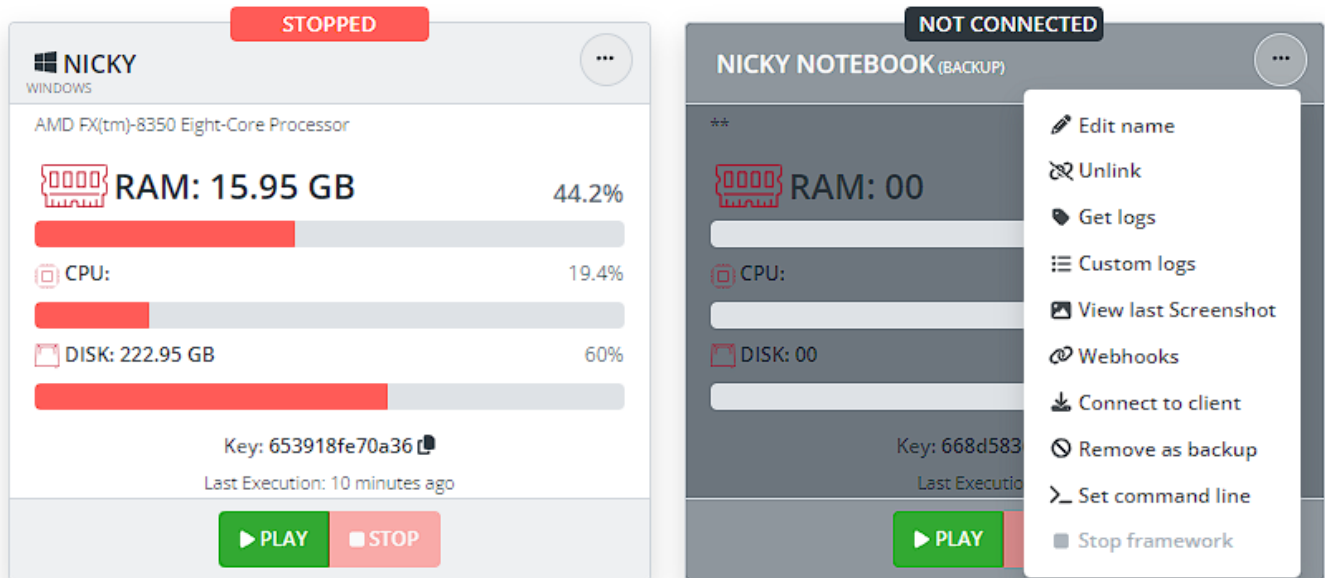
Si el robot no ha sido ejecutado previamente o si la última ejecución falló al intentar conectarse, se mostrará un modal que indica la falta de conexión.



- Con conexión:

Una vez establecida la conexión, se desplegará información detallada sobre el estado actual del robot y los comandos ejecutados.

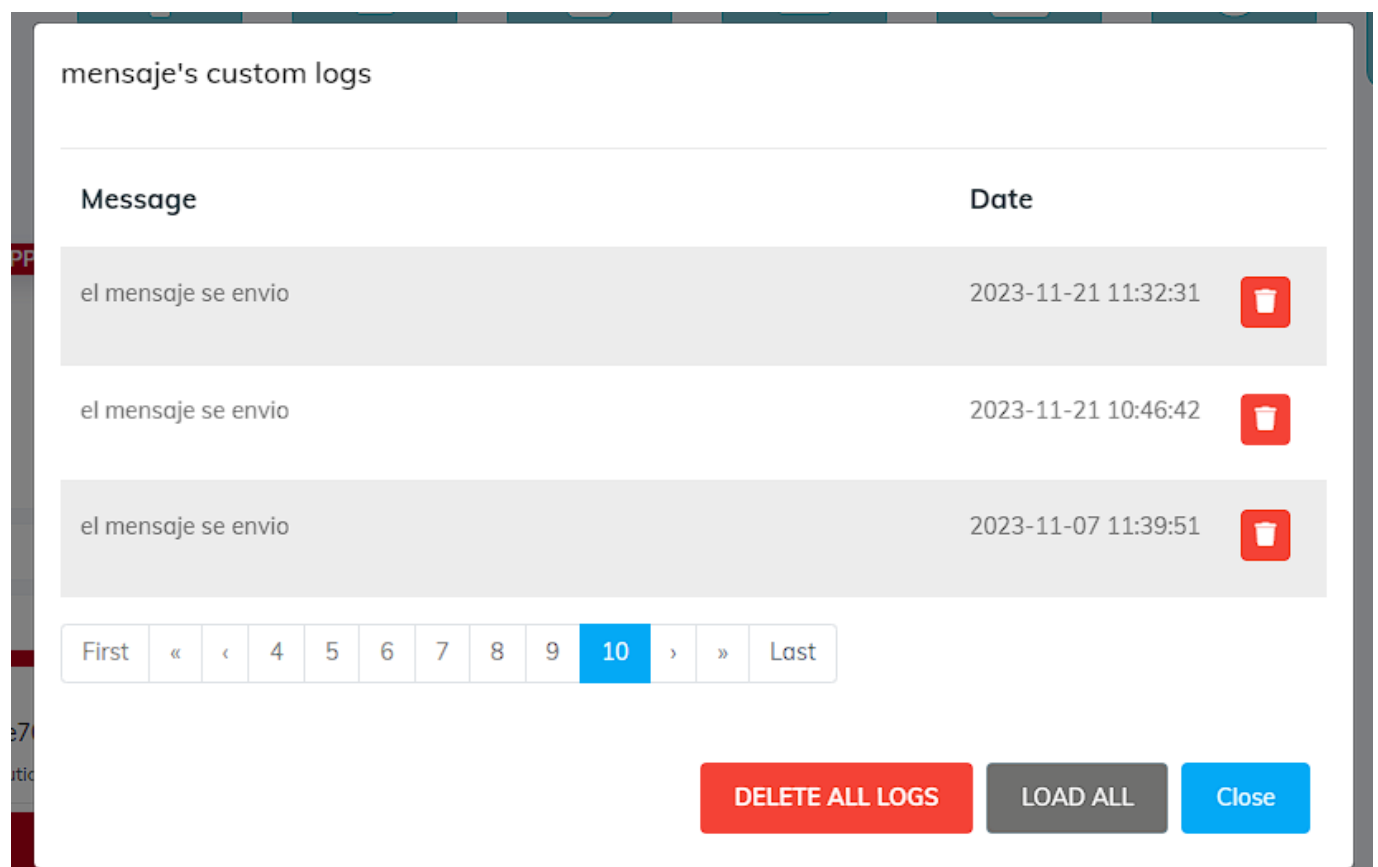




## Custom logs

Utilizando el módulo “Rocket Logs”, podemos enviar mensajes tanto internos al orquestador como a correos electrónicos predefinidos. Estos mensajes son útiles para recibir notificaciones y tener control sobre los comandos de nuestros robots.

Por ejemplo, un comando “Custom Log” con los mensajes “El robot comenzó” y “El robot finalizó” nos indicará el estado del robot. Además de enviar estos mensajes al orquestador, también es posible enviarlos por correo electrónico.

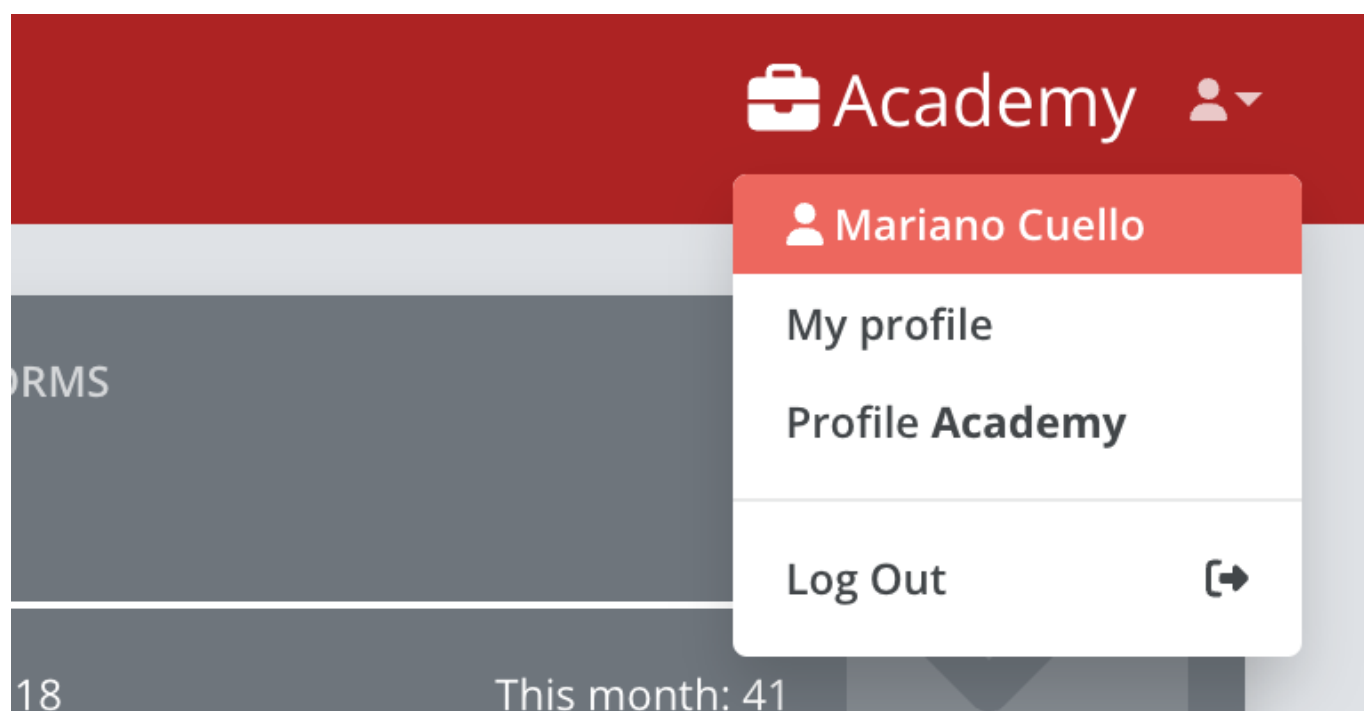


# Orquestador Rocketbot : Generacion de Api Key

## ¿Cómo generar una API Key?

### Primer método

Necesitamos dirigirnos a la sección del perfil.



Desde allí, hacemos clic en el botón "My Profile".

**2 Factor authentication:** Disabled

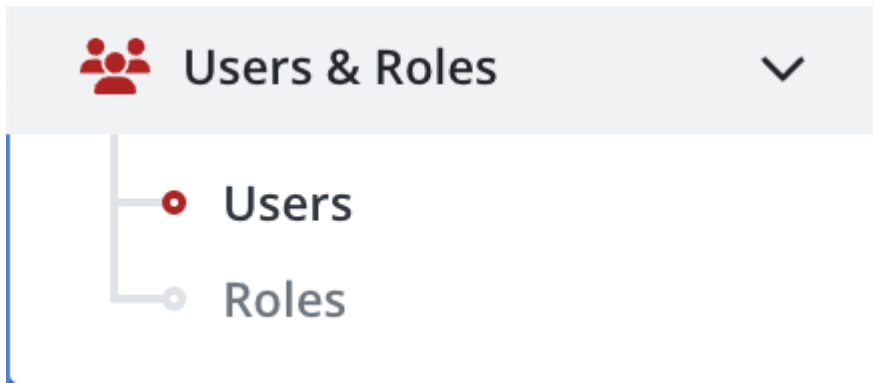
**API key:** eyJ0eXAiOiJKV1QiLCJhbGciOi ... [Copy](#) / [Refresh](#)

**API key expiration:** 8/7/2026, 10:44:06 AM

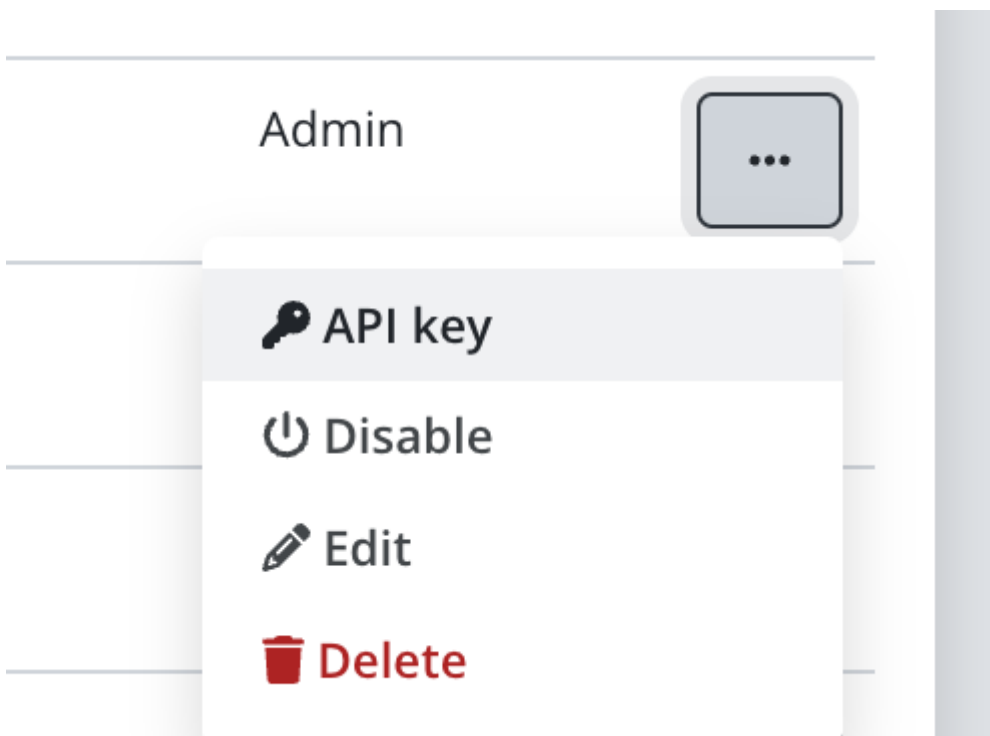
En el apartado de API Key, podemos refrescar o copiar. Al hacer clic en "Refresh", generamos una nueva API Key.

## Segundo método

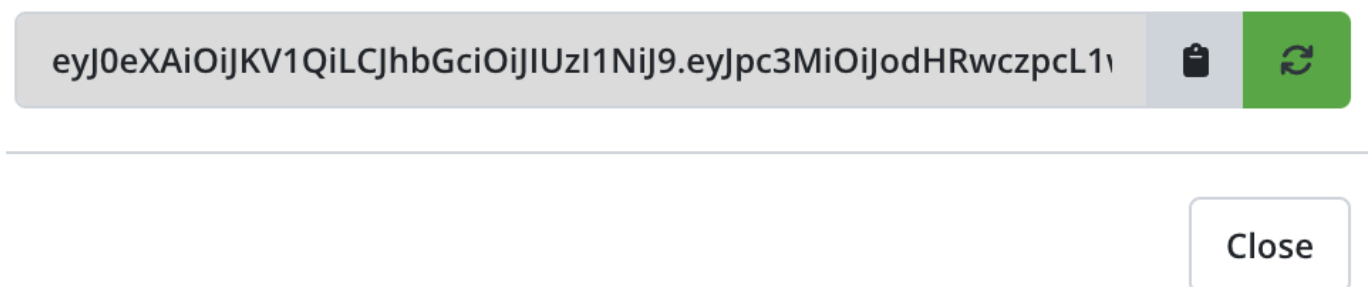
Nos dirigimos al menú del orquestador, al módulo de “User y Roles”, y allí hacemos clic en “Users”.



Luego, en usuarios, encontraremos el menú lateral (tres puntos) y podremos seleccionar la opción de API Key.



Una vez ahí, podremos copiar o generar una API Key.



Si está interesado en saber cuándo expira su API Key, se recomienda leer el siguiente documento: [Documentación sobre la expiración de API Keys en](#)

[Rocketbot](#).

---

## Orquestador Rocketbot : Procesos

### ¿Qué es un proceso?

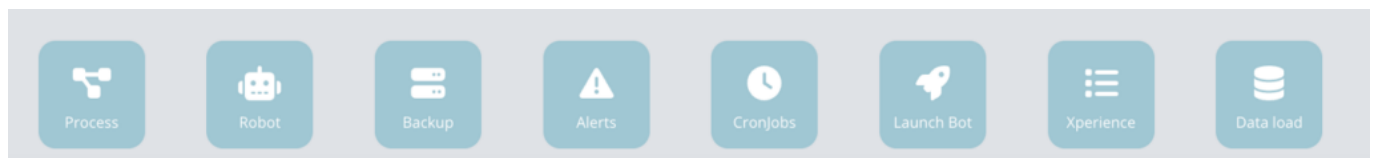
Un proceso es donde se configurará el robot y la automatización.

### ¿Cómo se crea o se asigna un proceso?

Para crear un proceso sigue el paso a paso del siguiente documento: [Creación de tu primer proceso](#).

Una vez el proceso este creado, al ingresar al mismo encontraremos lo siguiente :

### Interfaz



### Process

Aquí se puede definir el nombre y la descripción, y cuenta con el *token*, el cual se puede copiar con acceso rápido. En la parte inferior encontramos los botones para borrar proceso, editar y cerrar.

#### Edit process

Name

Description

Token

[Delete process](#) [Close](#) [Save](#)


## Robot

Aquí se debe asignar el nombre para identificar al robot dentro del orquestador. En *Start robot name* se debe asignar el nombre del robot a ejecutar, que debe coincidir exactamente con el de la base de datos; de lo contrario, no se encontrará el robot a automatizar.

También es necesario asignarle la base de datos del robot.

### Edit robot

<b>Robot name</b> <input type="text" value="prueba orquestador"/> <small>Robot name must be at least 3 characters long, alphanumeric and whitespaces only.</small>	<b>Start robot name</b> <input type="text" value="bot_orquestador"/>	<b>Uplad DB</b> <input type="button" value="Choose File"/> <input type="text" value="No file chosen"/>
--	---	---



Add a process and load a robot to run in one or more instances.  
You can load a robot by exporting to DB production or a DB project.

---

## Backup Instance

Aquí podemos seleccionar una instancia de respaldo, en caso de que la principal se caiga, y los minutos a partir de los cuales deberá activarse si la principal no se reconecta.

## Backup instance

Select instance

-- Select --



Minutes to activate

0

Close

Save

## Alerts

En esta sección, se puede configurar una notificación ante una desconexión de la instancia principal. Aquí se deben poner los correos electrónicos donde se enviarán las alertas y los minutos posteriores a la eventualidad (sin que haya reconexión) en los que se disparará la alerta.

## Edit process alerts email

Email

Add tag...



Email separated by comma: `user1@mail.com,user2@mail.com`

Minutes

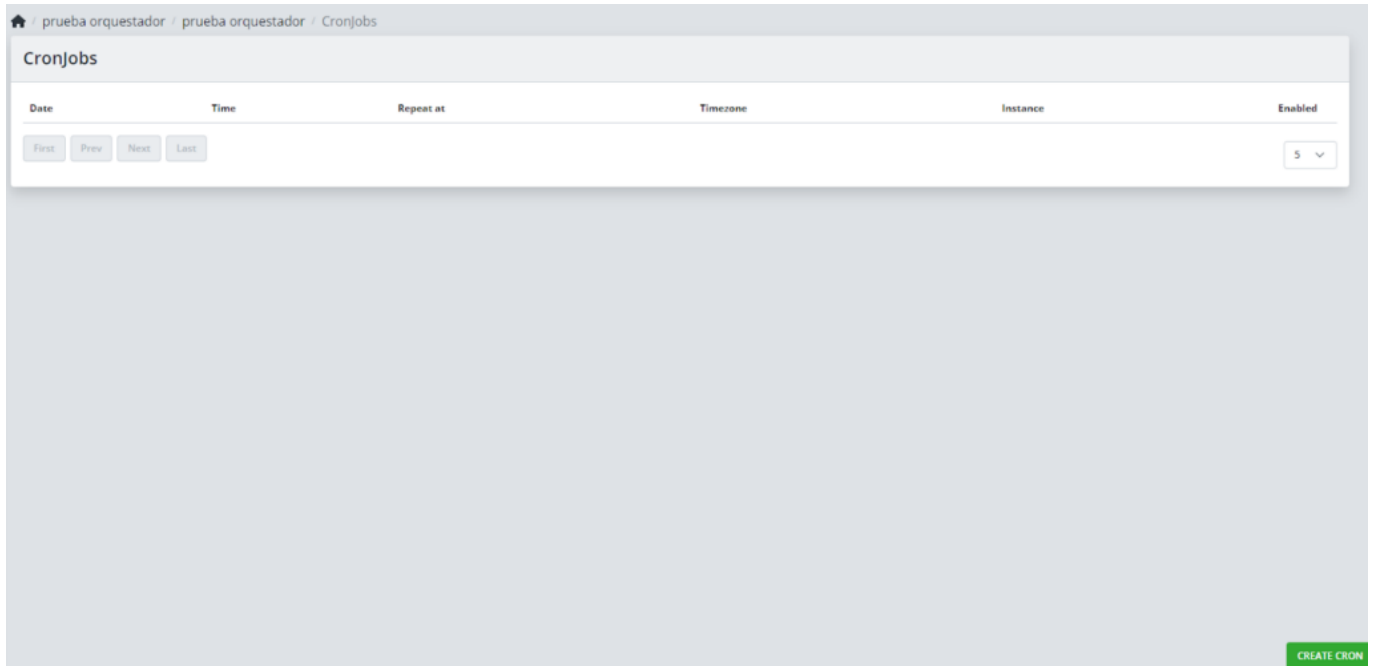
0

Close

Save

## Cronjobs

En esta sección se podrán ver las ejecuciones programadas para cada proceso y configurar nuevas realizando click en [\[ Create Cron \]](#).



Para mayor detalle, puede revisar el siguiente documento: [Trabajar con Cronjobs](#)

### Launch Bot (*trigger*)

Esta opción permite seleccionar otro proceso para que inicie, o se dispare, cuando finalice el proceso que se esta configurando.

## Set trigger bot

### Process



Select another process to start when the instances finish.

Close

Save

## Xperience

Esta opción permite seleccionar un formulario Xperience para que lance la ejecución del proceso cuando se envíe dicho formulario, ya sea de manera manual o mediante API.

### Set Xperience form

Form

No trigger



Select a form to trigger the process when you receive some raw queue.

Close

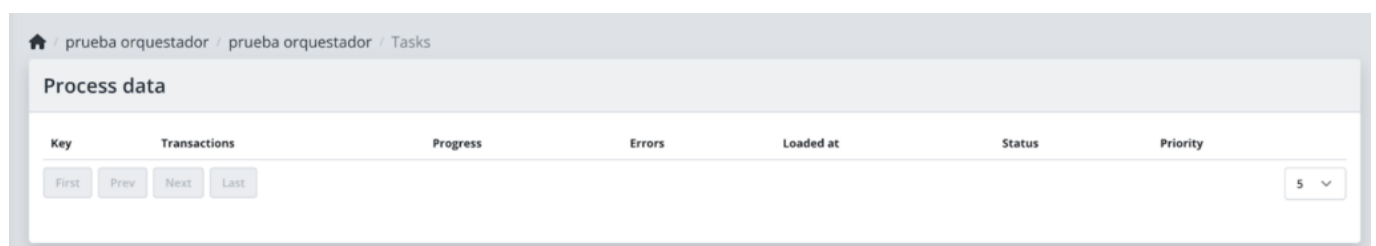
Save

## Data Load

Aquí encontraremos las tareas y transacciones asignadas al proceso, utilizando el módulo Orchestrator Framework desde otro robot, y su estado actual.

Si necesita mas informacion sobre Data Load puede revisar el siguiente documento :

[Orquestador Rocketbot : Data Load](#)



prueba orquestador / prueba orquestador / Tasks

Process data

Key	Transactions	Progress	Errors	Loaded at	Status	Priority
First	Prev	Next	Last			5

## Asignar una instancia

Para agregar una o mas instancias donde ejecutar tu proceso sigue el paso a paso del siguiente documento: [Agregar Instancia](#).

## ¿Cómo paralelizar?

Cuando hablamos de paralelizar debemos distinguir 2 escenarios:

1. Paralelizar un mismo proceso en diferentes instancias
2. Paralelizar diferentes procesos en una misma instancia

El robot de un proceso se puede ejecutar en múltiples instancias, el orquestador enviará la señal de ejecución a todas las instancias asignadas y conectadas al proceso (Salvo las de backup). Así mismo una misma instancia puede estar asignada a múltiples procesos y en la misma ejecutar el robot de cada uno de ellos, en momentos diferentes o al mismo tiempo, según el proceso lo permita o no (LINK). A su vez, cabe mencionar que no se puede ejecutar un robot varias veces al mismo tiempo en una instancia.

Para saber mas acerca de que procesos pueden paralelizarse:

[¿Qué considerar para ejecución en paralelo?](#)

## Queues de un proceso

Las queues que figuran en el dashboard de procesos representan ejecuciones pendientes del mismo que se generan cuando un proceso diferente ha disparado al que contiene estas colas. Las colas o queues del proceso se generan cuando el mismo no tiene una instancia disponible para ejecutar el robot al momento en que el se produjo el “disparo”.

En otras palabras, sólo se producen cuando un proceso tiene configurado un trigger a otro, al finalizar el primero, se ejecuta el trigger y este aguarda a tener una instancia disponible.

---

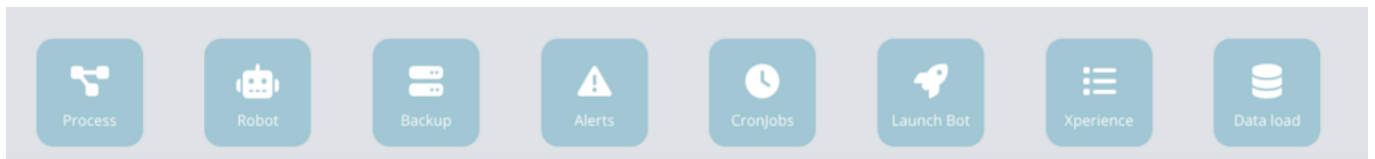
## [Orquestador Rocketbot : Xperience](#)

En el orquestador podemos usar los formularios de Xperience, podremos crear

formularios, saber de su creador y cuando se creó .

## Xperience en procesos

En la creacion de procesos encontraremos ya en su interfaz el siguiente menú:



En el apartado Xperience podremos seleccionar el formulario para que lance (trigger) la ejecución del proceso cuando se realice un envío del mismo.

### Set Xperience form

Form



Select a form to trigger the process when you receive some raw queue.

Close Save

Para revisar los formularios de Xperience, debemos seleccionar en el menú lateral *Form Xperience*.

En la barra lateral nos aparecerán las tres opciones:

## Form Xperience



- Forms
- Queues
- Forms Users

## Forms Xperience :

### Forms :

Aquí podemos crear formularios de Xperience y también observar sus detalles en el orquestador, como su nombre, estatus, si es público, los envíos (*submits*), la fecha de la última modificación y el último envío (*last submit*).

Al lado, en los tres puntos, encontramos el menú donde podemos ver el formulario, editarlo o borrarlo.

The screenshot shows the 'Forms' management interface. At the top right, it indicates 'Total 110 /10000'. Below this is a search bar labeled 'Search form'. The main content is a table with the following columns: Name, Enabled, Public, Submits, Last Modification, and Last Submit. The table lists five forms: testQA, seguimientoAuditoria, PruebaForm, testform, and Mobilbox. Each row has a three-dot menu icon on the right. At the bottom of the table, there are pagination controls: 'First', 'Prev', '...', '19', '20', '21' (highlighted), '22', 'Next', 'Last', and a dropdown menu showing '5'. In the bottom right corner, there is a green button labeled 'CREATE FORM'.

Name	Enabled	Public	Submits	Last Modification	Last Submit	
testQA	✓	✓	0	2 months ago	--	...
seguimientoAuditoria	✓	✓	29	2 months ago	1 month ago	...
PruebaForm	✓	✓	8	2 months ago	2 months ago	...
testform	✓	✓	0	1 month ago	--	...
Mobilbox	✓	✓	2	1 month ago	1 month ago	...

Si deseamos crear un formulario, hacemos clic en [**Create Form**]. Se nos pedirá el nombre del formulario y tendremos dos opciones: comenzar con el formulario desde cero o importarlo desde un archivo.

## Create a new Form

### Create an amazing form to collect data easily.

Form name

Form must be at least 3 characters long, alphanumeric only

- Start from scratch
- Import form from file

Close

Create

Para la creación de formularios o para importarlos, se recomienda ver el siguiente curso de Xperience: [Curso de Xperience](#).

### Form Queues :

Aquí podemos observar todas las *form queues*, utilizando un filtro para buscarlas. Abajo, veremos el nombre del formulario, su ID, el estatus, *queue by*, *locked by*, la fecha de la *queue* y la fecha del proceso.

En el menú lateral nos da la opción *restart* , finalizar y mostrar más o extra .

🏠 / Queues

Form queues: 11

From: 06/08/2024 To: 13/08/2024 [Filter by dates](#)

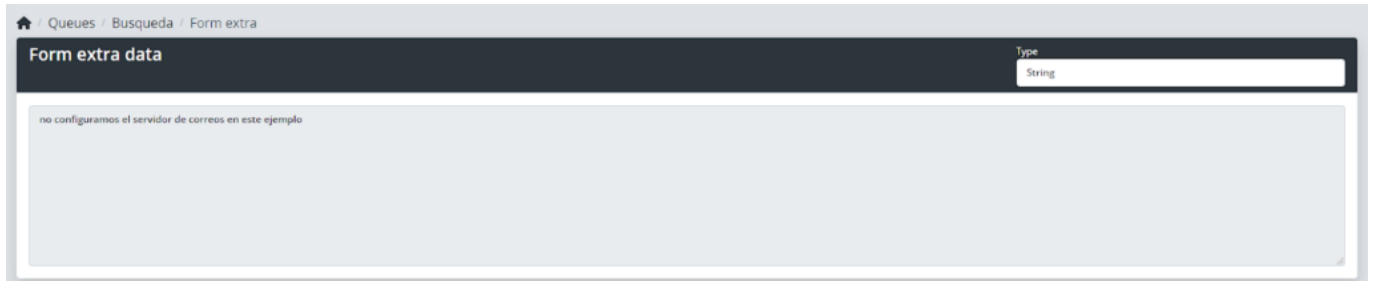
Search queue:  Status: -- All -- Form: -- All -- Queued by: -- All --

Form name	Id	Status	Queued by	Locked by	Queue date	Process date	
exportaCertificaciones	8456	✓ Processed	N/A	Robot Interno	2024-08-07 12:40:13	2024-08-07 12:42:05	⋮
exportaCertificaciones	8460	✓ Processed	N/A	Robot Interno	2024-08-09 09:24:23	2024-08-09 09:26:02	⋮
Saludo	8461	✓ Processed	N/A	Marcela	2024-08-09 11:26:26	N/A	⋮
Clientify	8462	✓ Processed	N/A	Robot Interno	2024-08-09 11:39:03	2024-08-09 11:40:30	⋮
Saludo	8463	✓ Processed	N/A	Marcela	2024-08-09 11:40:50	N/A	⋮
Saludo	8464	✓ Processed	N/A	Marcela	2024-08-09 11:46:53	N/A	⋮
Saludo	8465	✓ Processed	N/A	Marcela	2024-08-09 11:47:29	N/A	⋮
Clientify	8466	✓ Processed	N/A	Robot Interno	2024-08-09 12:31:51	2024-08-09 12:33:18	⋮
exportaCertificaciones	8467	✓ Processed	N/A	Robot Interno	2024-08-12 10:33:58	2024-08-12 10:34:58	⋮

⋮ menu: Restart, Finished, Show extra

En “show extra” podremos visualizar la información que se envía a través de xperience.

Con el modulo “Rocketbot Xperience” podremos utilizar comandos que interactúen con los formularios xperience, por ejemplo con el comando “Return Message to Xperience” que podremos enviar un mensaje y poder visualizarlo en la sección “show extra”:



## Procesamiento de las Queues (Xperience):

Estas son las entradas de datos realizadas desde los formularios. Si un proceso tiene configurado un formulario, el orquestador realiza un chequeo cada un minuto, si hay datos sin procesar de ese formulario; en caso de encontrar datos (una queue pendiente), si instancias disponibles, se ejecutará el robot de ese proceso en las instancias asignadas disponibles. Si todas están ocupadas, el orquestador repetirá la comprobación al minuto siguiente.

## Form Users :

Aquí se encuentran los usuarios y sus correos electrónicos, junto con la cantidad de formularios que poseen. También podemos modificar a los usuarios.

---

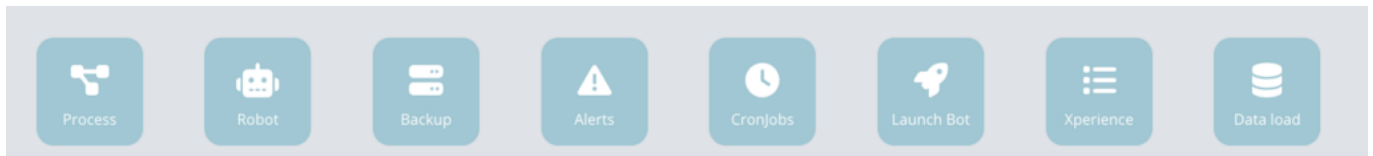
# Orquestador Rocketbot : Trabajar con cronjobs

## ¿Qué es un cronjob?

El cronjob de un proceso es una o más reglas que determinan la fecha, hora y periodicidad de ejecución del mismo.

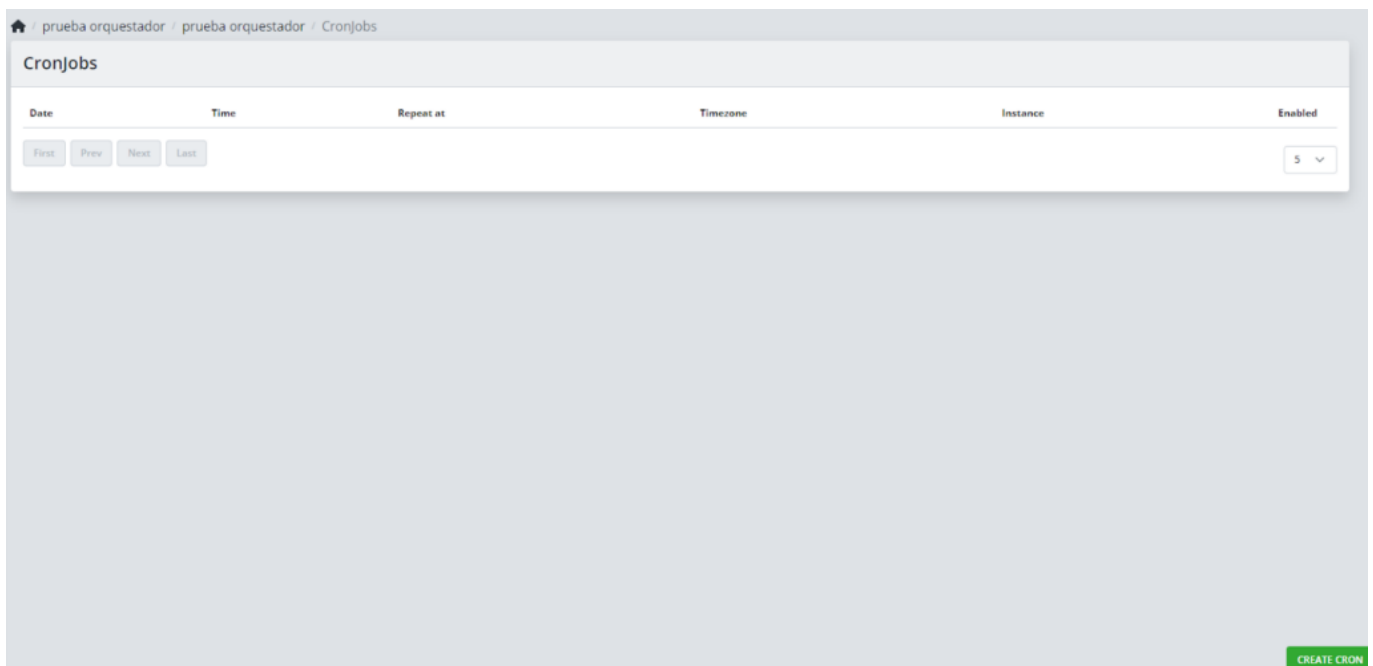
## ¿Cómo crearlo?

Al ingresar al proceso en el orquestador, encontraremos la sección de cronjobs ubicada en el menú superior.



## Cronjobs:

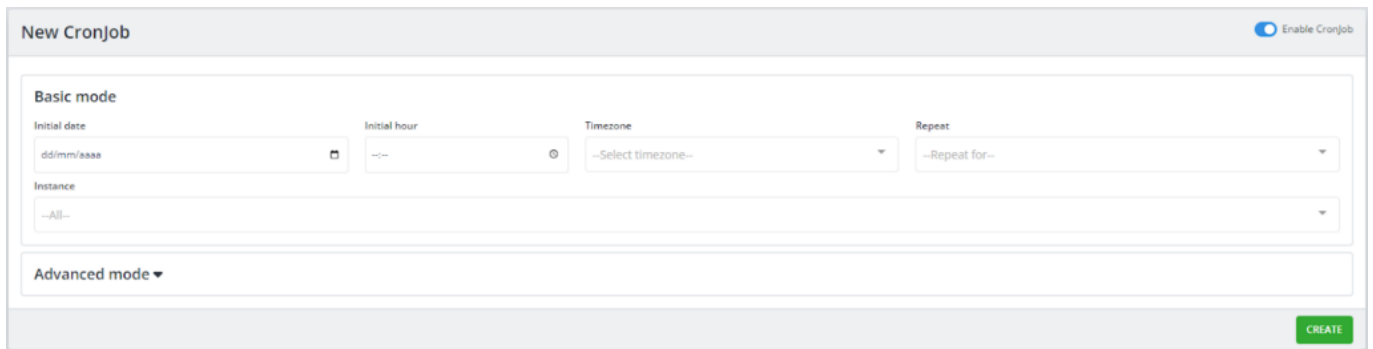
En esta sección se encontrará un tablero con el listado de cronjobs definidos para el proceso y, en la esquina inferior derecha, estará el botón para crear nuevos.



## Crear Cronjob:

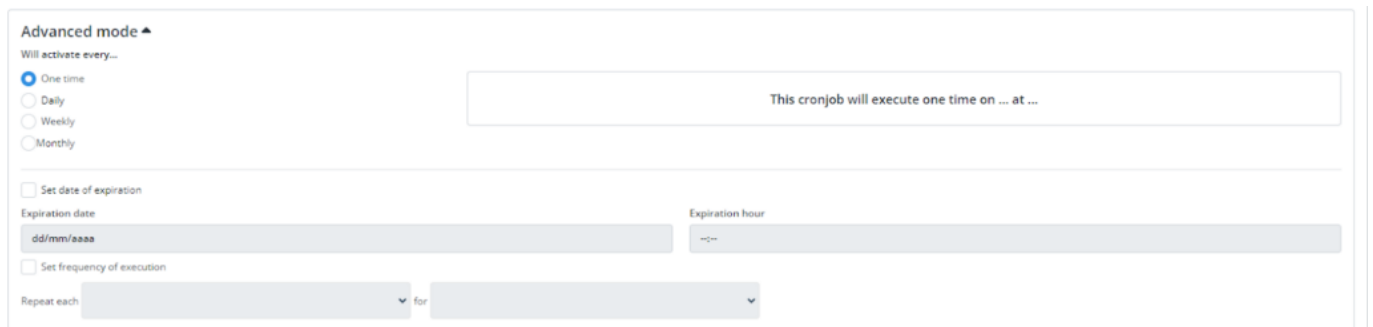
## Opciones básicas:

- Date: Fecha de implementación
- Time: Hora de ejecución
- Timezone: Zona horaria del ambiente de ejecución
- Repeat at: Periodicidad de ejecución
- Instance: Instancia asignada de las disponibles en el proceso
- Enabled: Opción que determina si el cronjob esta habilitado o no



The screenshot shows the 'New Cronjob' form in 'Basic mode'. It includes fields for 'Initial date' (dd/mm/yyyy), 'Initial hour' (24-hour format), 'Timezone' (dropdown menu), and 'Repeat' (dropdown menu). There is also an 'Instance' dropdown menu. A 'CREATE' button is visible at the bottom right.

## Opciones avanzadas:



The screenshot shows the 'Advanced mode' form. It includes radio buttons for 'One time', 'Daily', 'Weekly', and 'Monthly'. A text box indicates 'This cronjob will execute one time on ... at ...'. There are checkboxes for 'Set date of expiration' and 'Set frequency of execution'. The 'Expiration date' field is dd/mm/yyyy and the 'Expiration hour' field is 24-hour format. The 'Repeat each' field is a dropdown menu followed by 'for' and another dropdown menu.

Estas opciones nos permiten definir más específicamente:

- Cada cuanto repetir el proceso (Una vez, diariamente, semanalmente, mensualmente). A su vez, es se podra especificar en detalle cuando, por ejemplo, al seleccionar semanalmente se puede seleccionar los días de la semana en los que se deberá ejecutar.
- Durante cuánto tiempo, permite definir una fecha de expiración de la regla.
- Con que frecuencia se ejecutará, por ejemplo, ejecutar cada una hora durante 12 horas.

## ¿Cómo funciona un Cronjob?

Cuando llega la fecha y hora configurada en un cronjob, el orquestador comprueba si existen instancias disponibles para ejecutar el proceso. En caso afirmativo, ejecutará el robot del proceso en todas las que estén disponibles, de las asignadas a dicho proceso.

**IMPORTANTE:** Si no hay instancias disponibles al momento de ejecución del Cronjob, el orquestador repetirá la comprobación cuando vuelva a coincidir el día y hora, es decir, se omite.

## Opciones Cronologicas de un Cronjob

### Frecuencia de ejecución del proceso:

**Únicamente:** El cronjob se ejecuta una sola vez en la fecha y hora definidas, y no se vuelve a ejecutar.

**Diariamente:** Se puede configurar para repetirse cada “X” días.

#### Advanced mode ▲

Will activate every...

- One time  
 Daily  
 Weekly  
 Monthly

Repeat every  days

**Semanalmente:** Se puede configurar para repetirse cada “X” semanas. Además, permite seleccionar los días de la semana en los que se desea que se ejecute el robot.

#### Advanced mode ▲

Will activate every...

- One time  
 Daily  
 Weekly  
 Monthly

Repeat every  weeks on  
 Sunday  Monday  Tuesday  Wednesday  Thursday  Friday  Saturday

**Mensualmente:** Se puede configurar para ejecutarse en uno o varios meses específicos, con dos tipos de configuraciones:

- **Días específicos del mes:** Por ejemplo, el 1 de enero, el 3 de marzo, etc.

#### Advanced mode ▲

Will activate every...

- One time  
 Daily  
 Weekly  
 Monthly

Months

Days

On  
--Weeks--  --Days--

- **Semanas específicas del mes:** Permite elegir en qué semana o semanas del mes se ejecutará el robot, y además seleccionar los días de la semana para esa ejecución. Por ejemplo, que se ejecute los lunes y miércoles de la primera y tercera semana de enero y marzo.

#### Advanced mode ▲

Will activate every...

- One time
- Daily
- Weekly
- Monthly

The screenshot shows a configuration window for a cron job. It has two main sections: 'Months' and 'Days'. The 'Months' section has a dropdown menu with options for January, March, May, July, August, September, October, November, and December. The 'Days' section has a dropdown menu with options for First, Third, and Last. Below these, there is a radio button for 'On' which is selected, and a dropdown menu with options for Monday, Wednesday, and Friday.

## Fecha de expiración del cronjob

Esta opción permite definir una fecha y hora en la que el cronjob dejará de ejecutarse y, por lo tanto, el robot dejará de activarse.

Set date of expiration

Expiration date

06/05/2025

Expiration hour

01:00 PM

## Bucle

Permite asignar un bucle para que el cronjob active el robot cada "X" tiempo y lo haga durante un periodo de "Y" tiempo.

Ejemplo: Que el robot se ejecute cada 1 hora durante un día completo.

Set frequency of execution

Repeat each

One hour

for

One day

# [Orquestador Rocketbot : Expiración User ApiKey](#)

## ¿Por qué expira?

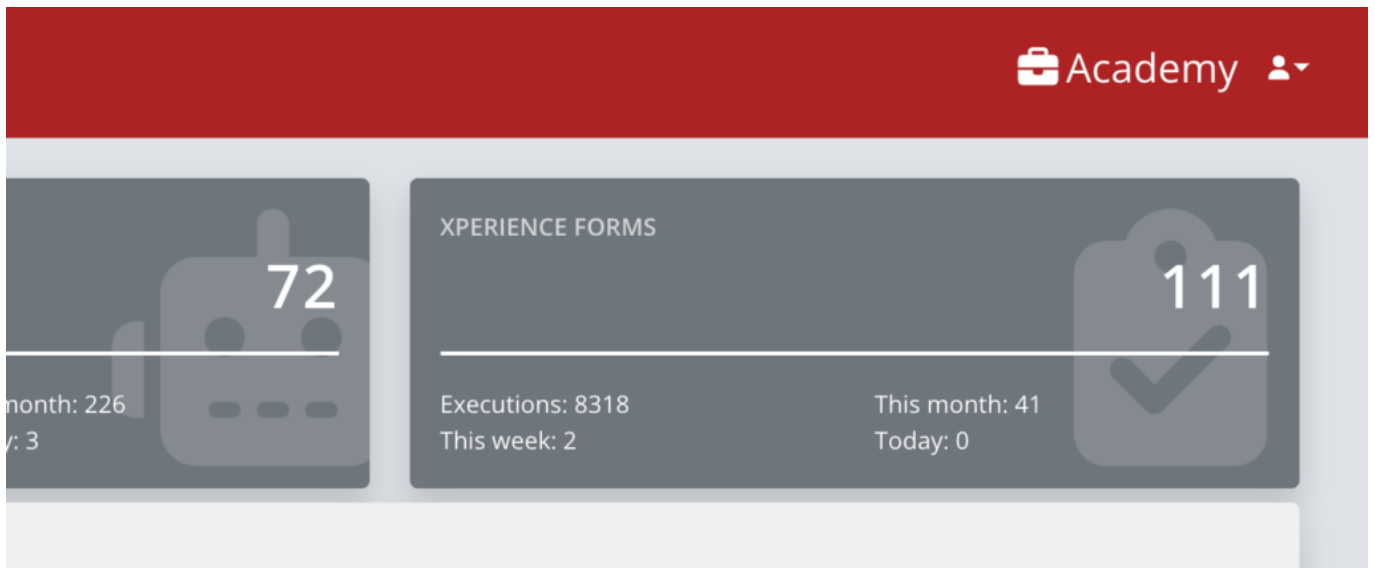
La API Key del Orquestador de Rocketbot expira en dos años como una medida de seguridad.

Este límite de tiempo ayuda a proteger la integridad del sistema al asegurarse de que las claves de acceso no se utilicen indefinidamente, lo que podría aumentar el riesgo de exposición o uso indebido.

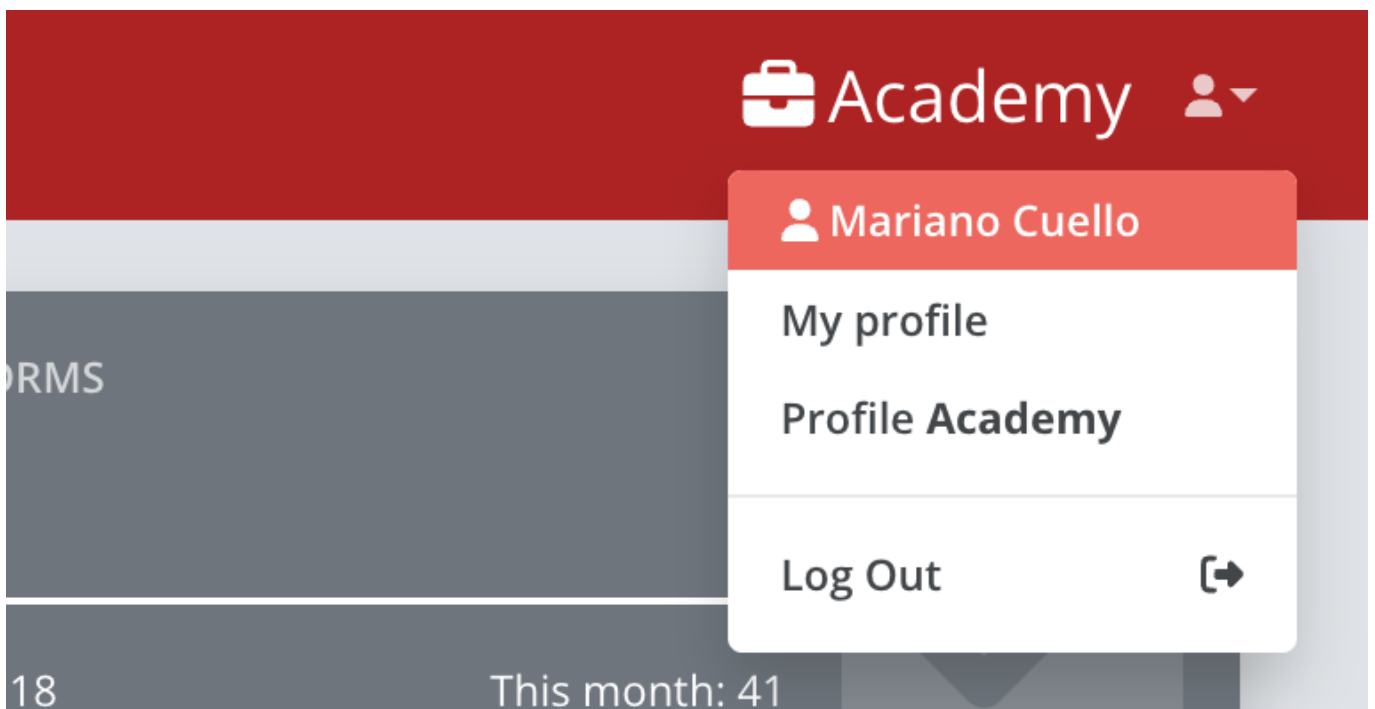
Al forzar la renovación periódica de las API Keys, se obliga a los administradores a revisar y actualizar las claves, asegurando que solo las personas y sistemas autorizados tengan acceso continuo. Además, esto permite implementar mejoras en la seguridad y cambios en las políticas de acceso de manera regular.

## ¿Dónde la encuentro?

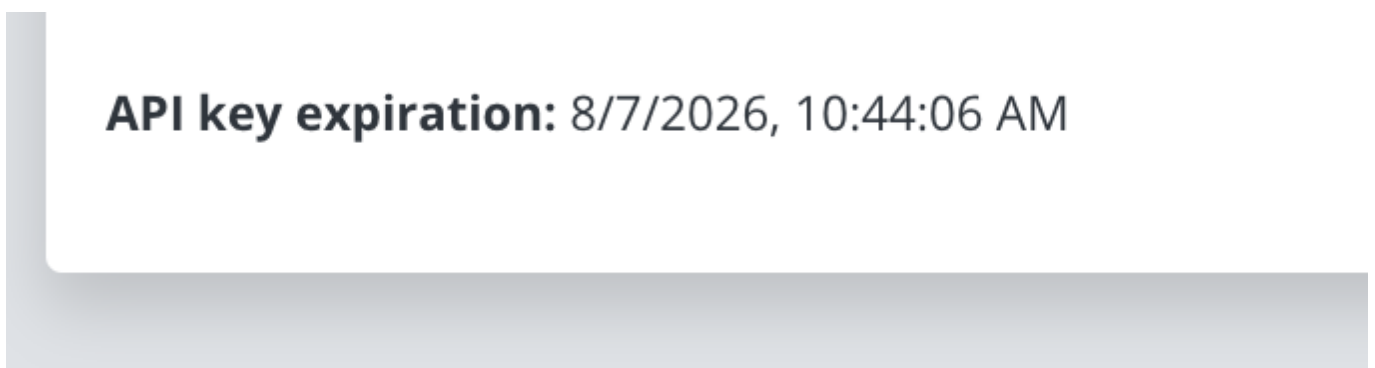
Para saber esta información debemos acceder al perfil del usuario.



Hacer clic en *My Profile*.



En la última casilla se encontrará la fecha de expiración de su API Key.

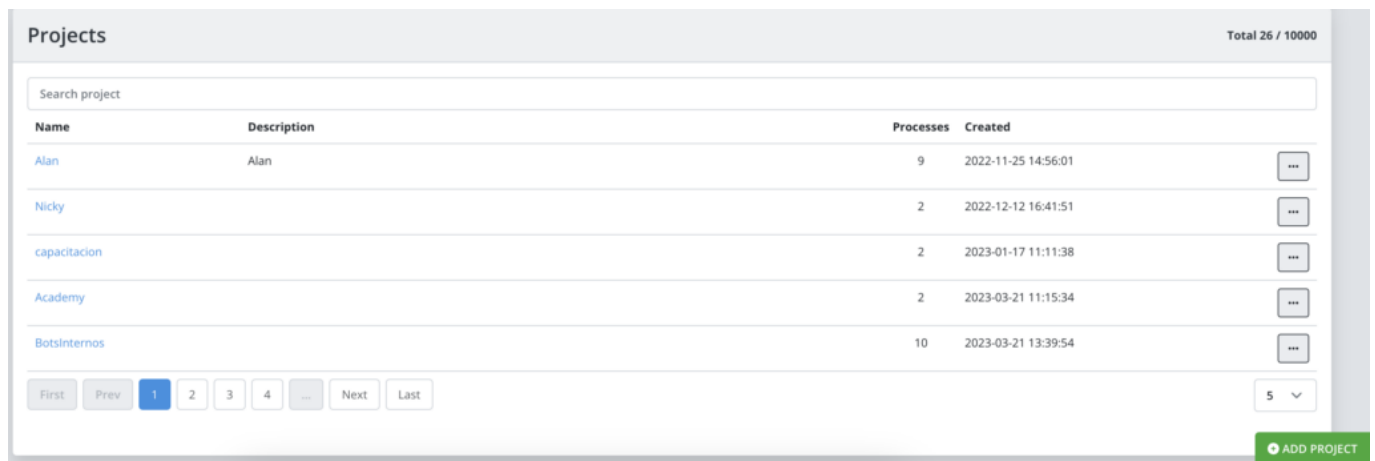


Si está interesado en la generación de API Keys, se recomienda leer el siguiente documento: [Generación de API Key en Rocketbot](#).

# Orquestador Rocketbot : Como crear mi primer proceso

## Paso 1: Proyecto

Un proyecto es una carpeta donde estarán los procesos automatizados. Su función es ordenar y agrupar los procesos.



The screenshot shows the 'Projects' dashboard in Rocketbot. At the top right, it says 'Total 26 / 10000'. Below the title is a search bar labeled 'Search project'. The main content is a table with the following columns: 'Name', 'Description', 'Processes', and 'Created'. There are five rows of project data, each with a three-dot menu icon on the right. At the bottom left, there are pagination controls: 'First', 'Prev', '1', '2', '3', '4', '...', 'Next', 'Last'. At the bottom right, there is a dropdown menu showing '5' and a green button labeled 'ADD PROJECT'.

Name	Description	Processes	Created
Alan	Alan	9	2022-11-25 14:56:01
Nicky		2	2022-12-12 16:41:51
capacitacion		2	2023-01-17 11:11:38
Academy		2	2023-03-21 11:15:34
Botsinternos		10	2023-03-21 13:39:54

## Crear proyecto

El primer paso que se debe realizar es crear un proyecto o seleccionar uno existente, al cual se le asignara el nuevo proceso. En el dashboard de inicio podrá ver los proyectos existentes para seleccionar.

Si desea crear uno nuevo haga click en el botón **[ Add Project ]** ubicado abajo a la derecha.

## New project



A project will allow you to have control of one or more robots and add instances of various types of systems so that your robots run in different architectures.

To create a project, enter the name and press create.

Name

Only alphanumeric characters and at least 3 characters long, project names cannot be repeated.

Close

Create

Asigne un nombre al proyecto que contendrá sus procesos. por ej: "QA", "Produccion", "DEV", etc.

Una vez creado, ábralo desde el la lista de proyectos para poder agregar procesos al mismo.

## Paso 2: Proceso y Robot

Un proceso es donde se subirá y configurará el robot, triggers y demás puntos correspondientes a la automatización.

A screenshot of a web interface for managing processes. The header shows 'ProyectoEjemplo processes' with an 'Add process' button and a 'Total 7 / 10000' indicator. Below the header is a search bar and a table with columns: Name, Description, Has robot, Cron task, Instances, Queues, and Status. The table contains two rows: 'Ejemplo2' and 'Ejemplo3'. Below the table are navigation buttons: 'First', 'Prev', '1', 'Next', 'Last', and a dropdown menu showing '5'. At the bottom right, there is a green 'ADD PROCESS' button.

Name	Description	Has robot	Cron task	Instances	Queues	Status
Ejemplo2		x	0	1	0	Public
Ejemplo3		x	0	1	0	Public

Para subir un robot, se debe crear o seleccionar un proceso existente.

## Crear proceso


Para crear un proceso debe hacer click en el botón **[ Add Process ]** ubicado abajo a la derecha.

Al nuevo proceso, se le debe ingresar:

- Nombre de proceso
- Nombre del robot principal o robot Padre que inicia el proceso
- Ruta del archivo exportado a producción desde Rocketbot Studio (extensión .db)

### Add process

<b>Process name</b> <input type="text" value="orquestador"/> <small>Process name must be at least 3 characters long, no special characters. Only one whitespace between characters.</small>	<b>Start robot name</b> <input type="text" value="bot_orquestador"/>	<b>Uplad DB</b> <input type="button" value="Choose File"/> <input type="text" value="bot_orquestador.db"/>
---	---	---



Add a process and load a robot to run in one or more instances You can load a robot by exporting to DB production or a DB project

Una vez creado, se puede seleccionar desde la lista de procesos para realizar las configuraciones siguientes.

## Paso 3: Instancia donde corre el robot


En caso de querer asignar una nueva instancia al proceso, primero debemos agregarla y luego vincularla, por defecto no estará conectada.

ProyectoEjemplo / Ejemplo3

Process Robot Backup Alerts Cronjobs Launch Bot Xperience Data load

All instances [Add or link instance](#) Total 0 / 10000

Available instances:  
**85 / 100**  
 You need to connect at least one instance to run this process [CONNECT INSTANCE](#)




[ADD OR LINK INSTANCE](#)

[¿Cómo vincular/crear instancia y conectarla?](#)

## Orquestador Rocketbot – Xperience form editor

Aquí se crean y editan los formularios de Xperience.

### Form settings (propiedades del formulario)

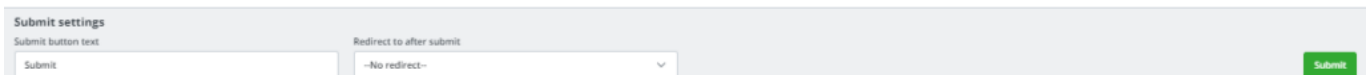
Form settings	
Form name: formExample	Public <input type="checkbox"/> Enabled <input checked="" type="checkbox"/> reCaptcha <input type="checkbox"/> send API <input type="checkbox"/> Form token: JCQF5GGF14FB85IV 
<b>Form name</b>	Permite editar el nombre del formulario. Debe contener más de dos caracteres, solo caracteres alfanuméricos y espacios, y no debe estar compuesto solo por espacios. No pueden existir dos formularios con el mismo nombre.
<b>Public</b>	Si está marcado como público, cualquiera que tenga el enlace del formulario podrá completarlo. Si está marcado como privado, solo los usuarios registrados con el formulario asignado podrán completarlo.
<b>Enabled</b>	Permite editar el nombre del formulario. Debe contener más de dos caracteres, solo caracteres alfanuméricos y espacios, y no debe estar compuesto solo por espacios. No pueden existir dos formularios con el mismo nombre.
<b>reCaptcha</b>	Cuando está activado, al intentar enviar un formulario, el usuario deberá completar un captcha.
<b>send API</b>	Si está activado, permite completar el formulario mediante una API, sin necesidad de una interfaz gráfica.
<b>Form token</b>	Indica el token asignado al formulario, el cual no puede ser editado. Al hacer clic en el ícono de copiado, este token se copiará en el portapapeles.

## Botones de Visualización / Guardado:



- Visualización** Redirige al usuario a la vista de completado del formulario, mostrando lo que verá el usuario final al completar el formulario.
- Guardado** Permite guardar el formulario. Si ocurre un error durante el guardado, se mostrará un mensaje de error y el formulario no podrá ser guardado.

## Submit settings (propiedades del completado del formulario):



- Submit button text** Aquí se indica el texto que tiene que tener el botón de completado de formulario. Por defecto es Submit.
- Redirect to after submit** Permite al usuario elegir qué ocurrirá después de completar el formulario. Por defecto, no hay redirección. El usuario puede seleccionar la opción URL para redirigir a otra dirección.
- Target** Esta opción aparece si el usuario eligió URL en "Redirigir luego de completar". Aquí se ingresa la URL a la que el usuario será redirigido después de completar el formulario.

## Botón de agregar elemento, guardar



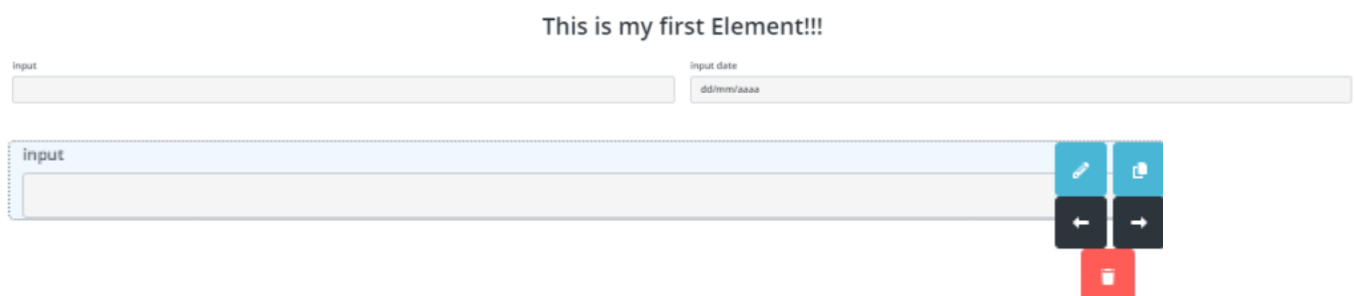
- Add element** Abre el modal de creación de elemento (ver sección de crear/editar elemento).
- Save** Guarda el formulario. Si surge algún error durante el guardado, se mostrará un mensaje de error y el formulario no podrá ser guardado.

## Pestaña de Editor



### Visor de elementos:

Aquí se pueden ver y modificar todos los elementos del formulario.



Pasando el mouse por sobre cada elemento aparecerá su menú de opciones.

Desde este visor las distintas acciones que se pueden realizar sobre los elementos son:

**Mover elemento:** Arrastrándolo con el mouse o utilizando las flechas del menú de opciones. “->” mueve el elemento una vez hacia la derecha, “<-” mueve el elemento una vez hacia la izquierda.



**Editar elemento:** Al hacer clic en el ícono del lápiz en el menú de opciones. Abre el modal de edición del elemento (ver sección de crear/editar elemento).



**Clonar elemento:** Al hacer clic en el ícono de copiar en el menú de opciones, se clonará el elemento (ver sección de clonar elemento).



**Borrar elemento:** Al hacer clic en el botón rojo con el ícono de cesto de basura en el menú de opciones, el elemento será eliminado.

### Crear / Editar formulario

Los formularios se crearán/editarán mediante un modal.

### New element

Element type (*)	Element id (*)	Bootstrap col class (*)
--Select form element type--	<input type="text"/>	--Select col class--
Element title		
<input type="text"/>		
Element style		
<input type="text"/>		
Element required	Css class	
No	<input type="text"/>	

(\*) Required fields

Al editar un elemento, se mostrará un modal similar, pero de color azul y con los campos del elemento completados con sus valores correspondientes. Tiene exactamente los mismos campos que un modal de crear elemento.

**Edit element**

Element type (*) <input type="text" value="Title"/>	Element id (*) <input type="text" value="12"/>	Bootstrap col class (*) <input type="text" value="col-md-1"/>
Element title <input type="text" value="1"/>		
Element style <input type="text"/>		
Element required <input type="text" value="No"/>	Css class <input type="text"/>	

(\*) Required fields

Al clonar un elemento, se abrirá un modal similar al de creación, con todos los campos llenos con los valores del objeto original, excepto el ID, que estará vacío. No se puede repetir el ID del elemento original. Al clonarlo, el clon aparecerá al lado (o abajo, si no cabe en la fila) del elemento original.

Si el elemento original tenía custom functions asignadas a eventos, su clon las tendrá asignadas de forma similar.

#### Atributos obligatorios:

Están marcados con un (\*); el elemento no puede ser creado/editado si no están completados.

- Tipo** *(ver tipos de elemento disponibles)*  
No puede repetirse entre elementos. Es obligatorio ingresar manualmente un id en todos los elementos salvo en Title, Paragraph, Image y Space. En estos casos si se crea/edita el elemento sin un id, se le asignará uno creado aleatoriamente.
- Id**
- Bootstrap col class** Indica el ancho que tendrá el elemento en el formulario, variando de 1 a 12 unidades. Una fila en un formulario puede contener hasta 12 unidades.

#### Atributos comunes a todos los elementos, no obligatorios

- css class** La clase css que tendrá el elemento en el formulario.
- style** El estilo inline que tendrá el elemento en el formulario.
- required** Si el elemento es requerido para poder completar el formulario.

#### Atributos específicos a ciertos elementos

- Element title** La etiqueta del elemento. Disponible en todos menos en Paragraph.

<b>Content</b>	Disponible en Paragraph. Contenido del párrafo.
<b>Pattern data</b>	Disponible en Entrada y Contraseña. Indica el patrón que debe tener el elemento. Si no se cumple, el formulario no puede completarse.
<b>File size y file type</b>	Disponible en File. File size indica el tamaño máximo en bytes que puede tener el archivo subido, y file type es un selector con el tipo de archivo requerido. Si se indica un tipo no se podrán subir otro tipo de archivos que no sea el tipo indicado.
<b>Max y Min value</b>	Disponible en Range. Indica el valor mínimo y máximo.
<b>Options (Image)</b>	Disponible en Image, Aquí se ingresa la url o el código base64 de la imagen que se desea mostrar.
<b>Options (Select)</b>	<p>Disponible en Select. Hay dos opciones, agregar las opciones manualmente o traerlas desde una API.</p> <p>Para agregarlas manualmente, ingrese un Text (Texto) y un Value (Valor) en los campos correspondientes, luego presione el botón verde "+" para añadir el par Texto-Valor. Ambos campos deben estar completos para que la adición sea exitosa.</p> <p>En la tabla se mostrarán todas las opciones ingresadas. Cada opción tiene botones para mover hacia arriba, mover hacia abajo y borrar. Los botones de flecha permiten cambiar la posición de las opciones, mientras que el botón rojo elimina la opción seleccionada.</p> <p>También puede traer valores a través de una API. Ingrese la URL donde se hará la petición, el nombre de la llave que trae el texto de cada opción en "Text Key" y el nombre de la llave que trae el valor de cada opción en "Value Key".</p>

## Events (eventos)

En los elementos tipo Input, Date, Number, Password, Textarea, Checkbox, File, Range el footer del modal tendrá un botón Show Events. Clickearlo expandirá un grupo de selectores, volverlo a clicar los esconderá.

Estos eventos ocurren cuando se realiza una determinada acción en cada elemento HTML del form. Estos son: *onclick*, *onchange*, *onblur*, *onmouseover*, *oninput*. Para más información sobre estos eventos ver [https://www.w3schools.com/tags/ref\\_eventattributes.asp](https://www.w3schools.com/tags/ref_eventattributes.asp).

Event	Custom function
On click	-- No function --
On change	-- No function --
On blur	-- No function --
On mouse over	-- No function --
On input	-- No function --

Cada evento tendrá seleccionado un selector, en el se podrá elegir una custom function (función customizada) creada por el usuario, para asignarla a ese evento. No es obligatorio asignarle funciones a cada evento.

Para ver cómo funcionan los eventos, dirigirse a *Events y custom functions*.

## Tipos de elemento disponibles

<b>Title (título)</b>	Heading de página.
<b>Paragraph (párrafo)</b>	Un bloque de texto.
<b>Input</b>	Un input para ingresar texto.
<b>Date (fecha)</b>	Un input para ingresar una fecha.
<b>Number (número)</b>	Un input para ingresar valores numéricos.
<b>Password</b>	Un input para ingresar una contraseña.
<b>Textarea (Área de texto)</b>	Un input para ingresar área de texto.
<b>Checkbox</b>	Un checkbox individual.
<b>File (archivo)</b>	Un input para seleccionar un archivo en el dispositivo desde el que se ve el formulario.
<b>Range (rango)</b>	Una barra en la que se seleccionan valores ubicados entre un valor mínimo y un valor máximo.
<b>Select</b>	Un selector de valores ingresados manualmente o traídos desde una api.
<b>Space (espacio)</b>	Un espacio vacío. En el editor se ve como una caja de líneas punteadas, pero en el visor del formulario se verá como un espacio vacío.
<b>Image (imagen)</b>	Una imagen, se puede traer mediante una url o mediante un código de base64.
<b>Signature (Firma)</b>	Un canvas donde el usuario puede dibujar su firma. En el editor se visualizará como un rectángulo de bordes sólidos.
<b>QR Reader (Lector de QR)</b>	Un lector de QR. En el editor se verá como una imagen estática.
<b>Barcode EAN Reader (Lector de código de barras)</b>	Un lector de código de barras. En el editor se verá como una imagen estática.
<b>Takephoto (Foto)</b>	Un capturador de foto. En el editor se verá como una imagen estática.

## Pestaña de Javascript



Aquí se puede customizar la experiencia del usuario del formulario agregándole código Javascript.


### Librerías CDN

Se puede agregar código Javascript de terceros mediante [librerías CDN](#).

En *URL to CDN* (Dirección a CDN) se puede ingresar la dirección a la librería deseada. Con el botón Add (Agregar) se agrega a la lista de CDNs agregadas.

No se puede agregar la misma dirección dos veces. Si se lo intenta aparecerá un mensaje de error y no se permitirá agregar la librería.

Add Javascript libraries to give more fascinating options to your Rocketbot Xperience Form View

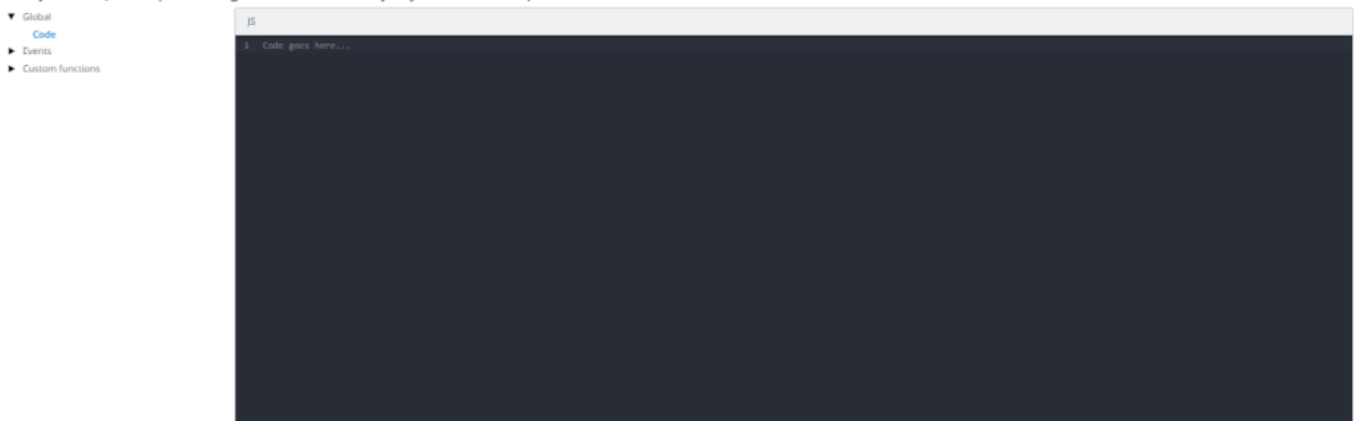
URL to CDN		Add
<a href="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js">https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js</a>		
<small>Library already added</small>		
#	Javascript Lib CDN	
1	<a href="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js">https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.30.1/moment.min.js</a>	

En la tabla se listarán todas las librerías asignadas, con su dirección y su número de orden en que fueron ingresadas, de primera a última ingresada. El botón rojo con el cesto de basura eliminará la librería seleccionada de la lista.

## Código manual

Aquí se puede ingresar manualmente código personalizado. A la izquierda se encuentra un menú desplegable con Global, Events y Custom functions. A la derecha se encuentra un editor de código asociado a la sección elegida en el menú. El área del menú que este en ese momento seleccionada estará resaltada en color azul en el menú.

Add your own Javascript code to give more functionality to your Rocketbot Xperience



## Global

Es código Javascript que afecta a todo el formulario. Cuando se ingresa a la pestaña de Javascript está expandido y seleccionado por defecto. En caso que se quiera volver a acceder, en el menú clicar *Global* para expandirlo y luego clicar en *Code*.

## Events (Eventos)

Bajo este menú se listarán, primero los eventos particulares del formulario, y luego los eventos asociados a los elementos del formulario.

### Form events (Eventos del formulario):

Son los eventos ligados a todo formulario.

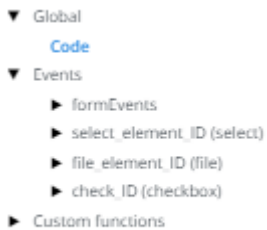
**load** Cuando un formulario se termina de cargar en el navegador.

**submit** Cuando un formulario es completado luego de clicar el botón de enviado.

**data\_received** Si el formulario está asociado a un robot, se ejecuta cuando llegan datos desde este a través del comando send data to Xperience.

Clickeando en cada uno de ellos se puede acceder a su editor de código.

## Events (eventos)

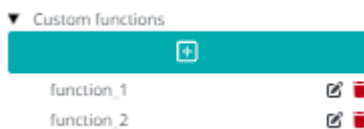


En este menú se listan todos los elementos creados del form cuyo tipo tenga eventos asociados (Input, Date, Number, Password, Textarea, Checkbox, File, Range). En el menú se listarán por su ID, con el tipo de elemento entre paréntesis.



Clickear cada uno de ellos los expandirá y se podrán ver listados todos los eventos asociados. Clickear cada evento accederá a su editor de código.

## Custom functions (Funciones customizadas)



Aquí se listarán por nombre todas las funciones creadas. Clickeando el botón de "+" al principio de la lista abrirá el modal de creación de función.

Cada función posee un nombre, ningún o varios parámetros y su código. Las funciones no pueden tener nombres repetidos y éstas y los nombres de parámetros deben seguir las reglas de nomenclatura de Javascript.

- Deben empezar con una letra, \_ o \$
- No pueden contener espacios en blanco
- No pueden ser [palabras reservadas](#).

Clickear una función en la lista accederá a su editor de código. Hacerlo en el botón de edición abrirá el modal de edición, donde se podrá editar el nombre y los parámetros de la función seleccionada. En el botón de borrado permitirá, previa confirmación, borrar la función seleccionada.

## Crear/editar función

**Function name (nombre de función):** No puede estar vacío, debe seguir las reglas de nomenclatura de Javascript y no puede estar repetido. Si el nombre es repetido o inválido aparecerá un mensaje de error y no dejará crear/editar la función.

**Function parameters (parámetros de función):** Es optativo, una función puede no tener parámetros. Siguen las reglas de nomenclatura Javascript y una función no puede tener dos parámetros con el mismo nombre. Para agregar un parámetro, se escribe su nombre y luego se ingresa presionando la tecla Enter, la tecla Space o la tecla ,. Para borrar un parámetro, se clickea la x al lado de su nombre.

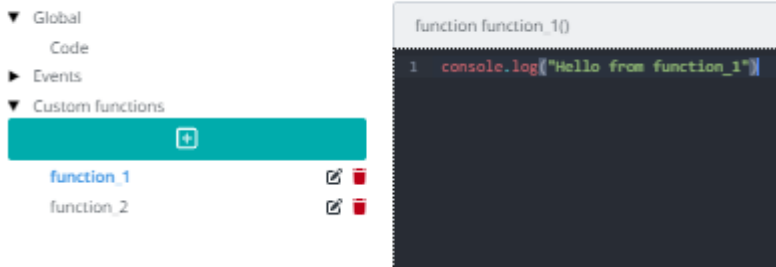
Si se intenta ingresar un parámetro inválido o repetido, aparecerá un mensaje de error y no se podrá agregar.

### Asignación y uso de custom functions

Declarando la función en el editor de código deseado la llamará para que se ejecute.

Por ejemplo se crea una función `function_1()` y en su código ingresamos `console.log("hello from function_1")`.

## Add your own Javascript code to give more functionality to your Ro



Se va hasta Events -> formEvents -> load y en su editor se ingresa `function_1()`.

## Add your own Javascript code to give more functi



Cuando un usuario abra un formulario, cada vez que este cargue, se llamará a la función `function_1` y se ejecutará el código asociado a esta, en este caso un `console.log`.

Hay dos formas de asignar custom functions. Una es ingresarlas manualmente, como se vio en el ejemplo anterior, y funciona tanto para código global, eventos del formulario y eventos de los elementos. La otra forma, que solo sirve para eventos de elementos, es asignar las funciones a través del modal de crear/editar elemento (*ver sección crear/editar elemento*).

Por ejemplo, en el modal de edición de un elemento tipo File, le asigno `function_1` al evento `onclick`.

**Edit element**

Element type (\*) Element id (\*) Bootstrap col class (\*)

File file\_element ID col-md-12

File size (in bytes) File type

all

Element title

Element style

Element required Css class

No

(\*) Required fields

---

Event	Custom function
On click	<input type="text" value="function_1"/>
On change	<input type="text" value="-- No function --"/>
On blur	<input type="text" value="-- No function --"/>
On mouse over	<input type="text" value="-- No function --"/>
On input	<input type="text" value="-- No function --"/>

Cuando se vinculan funciones de esta manera, al ir hacia el editor de código del evento, la función se declara al final del código previamente existente y se bloquea el editor. Solo se puede ver el código, sin posibilidad de edición.

▲ The code cannot be edited because a custom function has been assigned to it.
 Enable code editor

```

1 console.log("hello")
2 //previous code here
3
4 function_1()//Function added automatically
5
```

Si se quiere volver a editar manualmente el código, se puede hacer clic en *Enable code editor* (Habilitar editor de código). El editor volverá a estar activo, pero si se regresa al modal de edición del elemento, se puede ver que la función fue desvinculada del evento.

## Pestaña de CSS style (estilos de CSS)



Aquí se puede customizar el estilo y la experiencia de usuario del formulario agregándole estilos de CSS.

## Librerías CDN

Se pueden agregar hojas de estilo de terceros mediante [librerías CDN](#).

En *URL to CDN* (Dirección a CDN) se puede ingresar la dirección a la librería deseada. Con el botón Add (Agregar) se agrega a la lista de CDNs agregadas.

No se puede agregar la misma dirección dos veces. Si se lo intenta aparecerá un mensaje de error y no se permitirá agregar la librería.

Add css libraries to improve your forms and give a spectacular user experience

URL to CDN	<input type="text" value="https://cdn.jsdelivr.net/npm/bootstrap@3.4.1/dist/css/bootstrap.min.css"/>	Add
<small>Library already added</small>		
#	Css Lib CDN	
1	https://cdn.jsdelivr.net/npm/bootstrap@3.4.1/dist/css/bootstrap.min.css	

En la tabla se listarán todas las librerías asignadas, con su dirección y su número de orden en que fueron ingresadas, de primera a última ingresada. El botón rojo con el cesto de basura eliminará la librería seleccionada de la lista.

## Editor de estilo de CSS

En este editor se pueden definir manualmente los estilos para la vista del formulario, especificando los selectores cuyos estilos de desean modificar.



Add css libraries to improve your forms and give a spectacular user experience

URL to CDN	<input type="text" value="https://cdnjs.cloudflare.com/ajax/lib.min.css"/>
#	Css Lib CDN
1	https://cdn.jsdelivr.net/npm/bootstrap@3.4.1/dist/css/bootstrap.min.css

```
1 h1 {
2   font-size: 25px;
3 }
4
5 .form-label {
6   color: red;
7 }
8
9 #first-name-input {
10  border: 1px solid black;
11 }
```

En el ejemplo de la imagen superior, se definió para todos los elementos `<h1>` del form un tamaño de fuente de 25 píxeles, para todos los elementos con clase `form-label` un color rojo y para los elementos con ID `first-name-input` un borde sólido negro de 1 píxel de ancho.

## Pestaña de code (código)



Aquí se puede ver y editar el código de los elementos del form, descargar el código `.json` de las variables, descargar un robot prefabricado que manipule el formulario, y descargar el código `.json` del formulario.

### Visor/editor de código

En el visor se puede ver el código `.json` de todos los elementos del form en orden de primero a último, con todos sus atributos.



Presionar el botón verde a la derecha habilitará el modo de edición. En el código se formateará de forma que su visualización y edición sea sencilla, como un elemento por bloque. Volver a clicar el botón cerrará el modo de edición, cambiará el formato a texto plano y guarda todos los cambios realizados.

Si se introduce código incorrecto, aparecerá un mensaje de error y al cerrarse el editor los cambios realizados desde la última vez que se habilitó el editor de código no se aplicarán.

### Botón de Download vars (descargar variables)

Crea todas las variables necesarias para que el robot autocomplete con los datos enviados al queue, genera y descarga un archivo `.json`, con cada uno de los elementos del form, siendo su atributo `name` el ID del elemento.

### Botón de Download robot prefab (Descargar robot prefabricado)

Crea y descarga un robot en formato `.json` para cargarlo en Rocketbot Studio.

### Botón de Export form (exportar formulario)

Crea un archivo `.json` con el formulario y lo descarga, con sus elementos, librerías y código Javascript/CSS, eventos y funciones personalizadas.