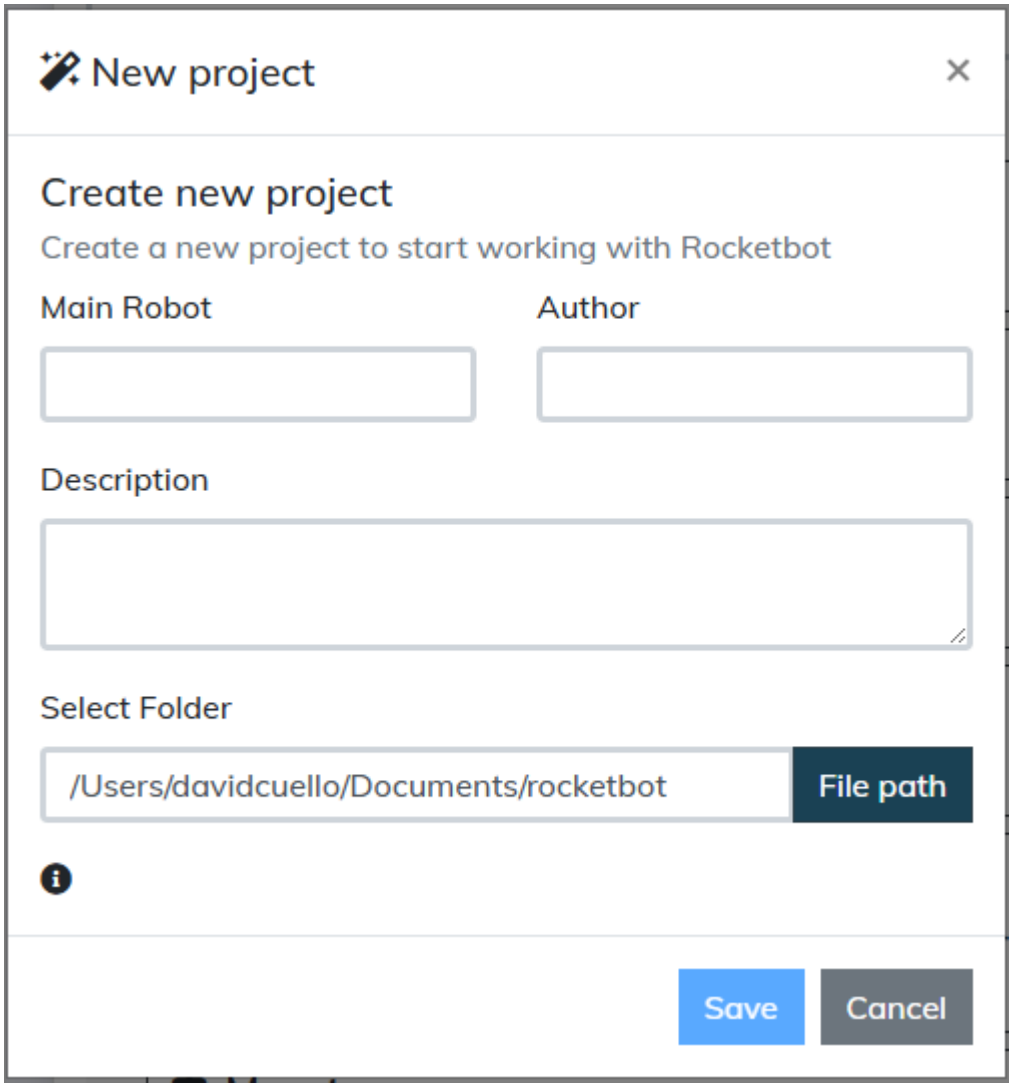
Studio Rocketbot — Iniciar un proyecto

Este documento explica como crear un proyecto en Studio que pueda ser ejecutado en cualquier ambiente que cuente con una licencia de Rocketbot Studio.

Nuevo proyecto

Para crear un nuevo proyecto, desde Studio ve a Inicio y haz click en el botón Nuevo Proyecto y verás el siguiente modal.



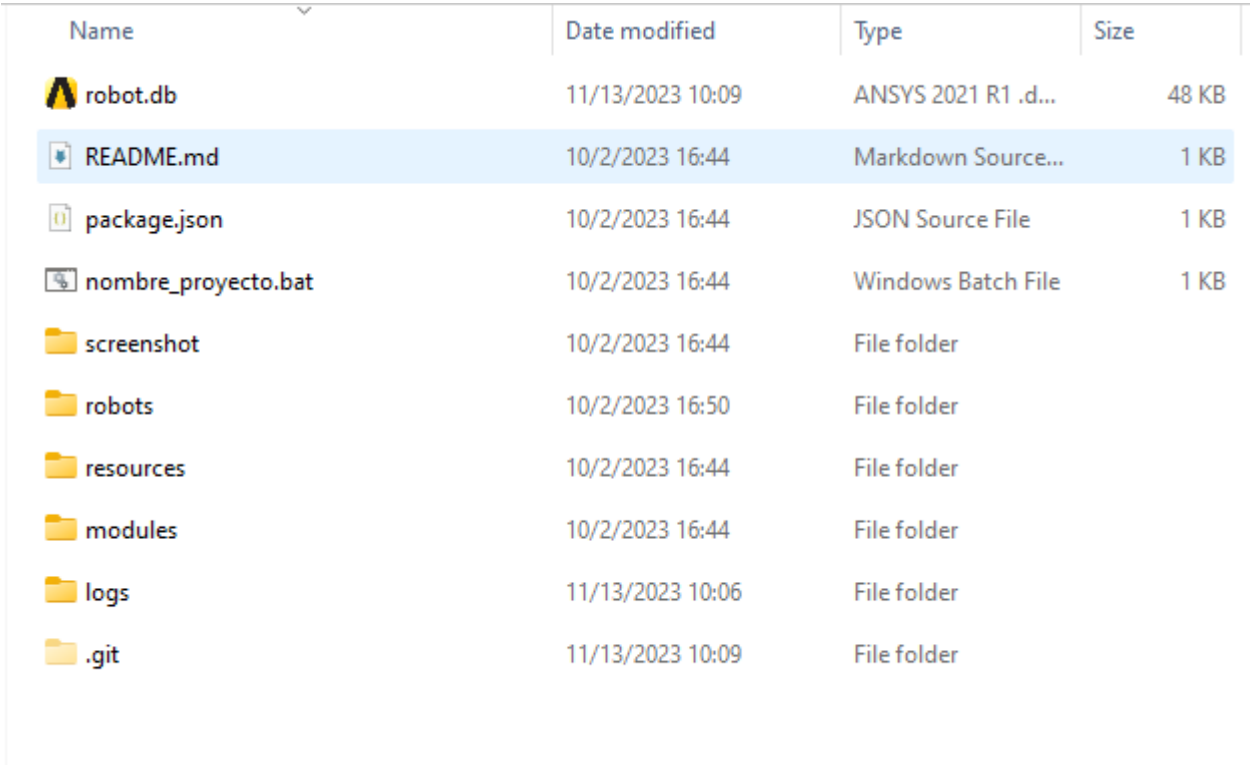
En el modal de nuevo proyecto, completa los siguientes campos:

- Robot principal: Especifica un nombre para tu robot principal y tu proyecto. Este nombre debe ser único para cada proyecto, ya que se utilizará para crear una carpeta con este nombre
- Descripción: Especifica una descripción que resuma lo que quieres hacer con este proyecto.
- Author: Indica el nombre del autor del proyecto
- Seleccionar carpeta: Selecciona la ubicación en la que quieres crear el

proyecto. La hubicación predeterminada donde se crean los proyectos es en %USERPROFILE%\Documents\rocketbot

Estructura de carpetas y archivos

Al completar los datos y crear el proyecto, se crearán los siguientes archivos y carpetas:



- robot.db: Base de datos SQLite creada por defecto donde se almacenará tu robot. En esta base de datos contendrá todos los subrobots que sean creados y podrás llamarlos desde el comando Ejecuta otro script Rocketbot

- README.md: Archivo markdown con el nombre del proyecto y la descripción. Puedes editarlo para agregar una documentación acerca de como ejecutar tu robot

- package.json: Contiene información acerca de tu robot. Este archivo contiene datos como: nombre, descripción, versión, autor, licencia, nombre del robot principal.

 Puedes incluir además las versiones de los módulos que utilice tu robot.

```
{
  "name": "Rocket",
  "description": "",
  "version": "1.0.0",
  "main": "run_main",
  "author": "",
  "license": "MIT",
  "modules": [
    {"name": "AdvancedExcel", "version": "13.2.1"}
  ]
}
```

- Archivo batch: Este archivo contiene la instrucción de consola para ejecutar tu robot. Está asociado al ambiente y rutas donde se creó el proyecto.

```
cd C:\rocketbot
rocketbot.exe -start=Rocket -
db="%USERPROFILE%\Documents\rocketbot\Rocket\robot.db"
```

- screenshot: Carpeta donde podrás almacenar las capturas de pantallas que realiza tu robot

- robots: Carpeta con las base de datos de robots como servicio que utiliza tu robot. En esta carpeta puedes agregar robots de otros proyectos para ser utilizados en tu robot

- Resources: Carpeta donde deberás alojar los archivos que tu robot necesita para trabajar y los archivos que genera.

 Puedes agregar subcarpetas si lo necesitas, como carpeta para descargas, plantillas, archivos de configuración, etc

- modules: Carpeta con los módulos que utilizará tu robot. Esta carpeta te permite tener diferentes versiones de los módulos para diferentes proyectos.

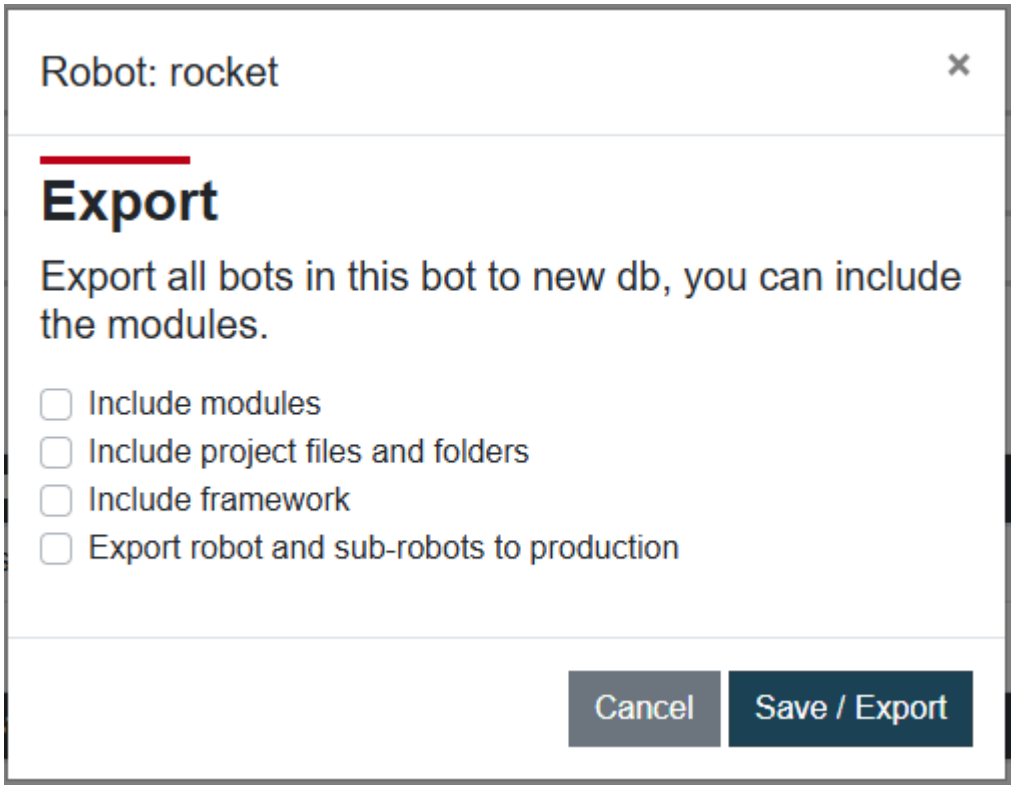
- logs: Carpeta donde se almacenarán los logs generados por la ejecución de cada comando de tu robot. Para que los robots se almacenen en esta carpeta, debes indicarlo en la configuración de tu robot, en caso

contrario, los almacenará en la carpeta de rocketbot.

- .git: Carpeta generada por git al generar un nuevo repositorio. Esto te permitirá usar desde studio los comandos de git para el control de versiones

Exportar un proyecto

Para exportar un proyecto debes ir a la pestaña Robot del ambiente de desarrollo y dar click en el botón ***Exportar robots a Db***. Dependiendo de lo que necesites, deberás marcar alguno de las casillas:



- Sin marcar casillas: Si no se marca ninguna casilla, se creará solo una base de datos de tu robot con la última versión de cada uno de los subrobots.
- Incluir módulos: Al marcar esta casilla, se creará una carpeta comprimida con todos los módulos que utiliza tu robot
- Incluir archivos y carpetas del projecto: Al marcar esta casilla, se creará una copia de la carpeta de tu proyecto en la ruta indicada luego de dar click en **Guardar/Exportar**
- Incluir Framework: Al marcar esta casilla, se exportará ademas una versión reducida de rocketbot que solo permite ejecutar los robots. Con esta versión reducida, puedes ejecutar tu robot sin permitir entrar a Studio
- Exportar robots a producción: Al marcar esta casilla, la base de datos creada se encriptará y no se podrá ver el contenido del robot desde

Studio.

⚠ Si marcas la casilla Exportar robots a producción, no selecciones una base de datos existente. Al encriptar los datos, no podrás volver a ver ni editar el contenido y podría perder tu robot

Diferencia con versiones anteriores

Esta documentación está basada en las funcionalidades de Rocketbot Studio 2024.05.28 o superior. Versiones tienen diferencias en la creación de proyectos y se deben configurar manualmente para replicar a las ultimas versiones

Versión 2023

La estructura de carpetas al crear un proyecto fue integrada en la versión Rocketbot Studio 2023.03.30 pero con ligeras diferencias

Carpetas y archivos

package.json	Este archivo no es generado en la versión 2023
modules	Esta carpeta se utiliza para almacenar las versiones de los módulo, pero el robot no los utiliza en la ejecución
logs	Esta carpeta era utilizada para almacenar los logs generados intencionalmente con el robot. No se almacenaban los logs generados por la actividad del robot
robots	Esta carpeta era utilizada para almacenar copias de seguridad de la base de datos original. No se podían utilizar los robots como servicios

Menú de exportar

La versión 2023 no cuenta con la opción de exportar incluyendo archivos y carpetas del proyecto ni incluir el framework

Versión 2020 e inferior

Las versiones inferior a la 2023 no cuentan con la funcionalidad de crear proyecto. Para estas versiones, se debe crear una base de datos y luego generar la estructura de carpetas de forma manual.

Para exportar una base de datos, puedes elegir entre dos botones en la pestaña robot. Exportar robots y subrobots, y exportar robots y subrobots a producción

Usando XPath para Elementos Dinámicos

Cuando tratamos con elementos dinámicos en una página web, como aquellos cuyos identificadores XPath o Full XPath pueden cambiar, XPath ofrece una flexibilidad que resulta valiosa. Puedes crear expresiones que se centren en atributos específicos para asegurar la identificación precisa de los elementos, incluso si otros aspectos de la estructura de la página cambian.

Estrategias XPath para Elementos Dinámicos

Identificación por Atributo

Especificar un atributo específico de un elemento, como una clase, ID o cualquier otro atributo único.

```
//tagname[@class='claseEspecifica']
```

Esto asegura que, aunque la posición del elemento o la estructura de la página cambien, aún puedes dirigirte al elemento deseado basándote en un atributo constante.

XPath Relativo

Construir rutas XPath que son relativas al contexto del elemento, evitando dependencias absolutas que podrían volverse obsoletas con cambios en la estructura

```
//div[@id='contenedor']/p[@class='miClase']
```

Utilizando rutas relativas, tu XPath se adapta mejor a cambios en la jerarquía de la página.

Funciones de XPath

Utilizar funciones XPath para seleccionar elementos basados en contenido, posición, u otras características dinámicas

```
//div[contains(text(), 'TextoDinamico')]
```

Las funciones XPath, como contains(), permiten seleccionar elementos basados en el contenido variable que pueden tener.

Ventajas de Utilizar XPath para Elementos Dinámicos

- **Adaptabilidad:** XPath permite construir expresiones que se adaptan a cambios en la estructura de la página.
- **Precisión:** Al basarse en atributos específicos, XPath proporciona una identificación más precisa de los elementos.
- **Mantenibilidad:** Estrategias como XPath relativo facilitan la adaptación

del scraper a modificaciones en la jerarquía de la página.

En resumen, al emplear XPath para elementos dinámicos y centrarte en atributos específicos, puedes construir expresiones más robustas y resistentes a cambios en la estructura de la página, asegurando una extracción de datos más fiable y duradera.

Tipos de Identificadores en Web Scraping

En este documento revisaremos algunos identificadores esenciales en el web scraping, tanto su significado, como ventajas y desventajas.

ID (Identificador Único)

- **Significado:** Cada elemento en una página web puede tener un atributo ID único. Es un identificador exclusivo asignado al elemento, permitiendo su identificación sin ambigüedad.
- **Ejemplo:** Contenido
- **Ventajas:** Rápida identificación y acceso directo al elemento. Ideal cuando se necesita seleccionar un elemento específico.
- **Desventajas:** No todos los elementos tienen un ID único. Los IDs pueden cambiar, afectando la estabilidad.

Clase (Class)

- **Significado:** Los elementos pueden tener uno o más atributos de clase. Múltiples elementos pueden compartir la misma clase.
- **Ejemplo:** Texto
- **Ventajas:** Útil para seleccionar grupos de elementos similares
- **Desventajas:** No garantiza la unicidad. Puede seleccionar más elementos de los deseados.

Selector de Etiqueta (Tag Selector)

- **Significado:** Se refiere al nombre de la etiqueta HTML (por ejemplo, div, p, a). Selecciona todos los elementos de esa etiqueta en la página.
- **Ejemplo:** <h2>Título</h2>
- **Ventajas:** Útil para seleccionar todos los elementos de un tipo específico en una página.
- **Desventajas:** Puede seleccionar demasiados elementos si la etiqueta es común en la página.

XPath (Ruta de Acceso XML)

- **Significado:** Es una expresión que define la ubicación de un elemento en un documento XML o HTML. Permite seleccionar nodos de manera precisa.
- **Ejemplo:** //div[@id='miElementoUnico']
- **Ventajas:** Proporciona flexibilidad y precisión. Es útil en situaciones donde otros identificadores no son suficientes.
- **Desventajas:** Puede ser largo y vulnerable a cambios en la estructura de la página.

Full XPath (Ruta de Acceso XML Completa)

- **Significado:** Especifica la ruta completa desde el nodo raíz hasta el elemento deseado.
- **Ejemplo:** /html/body/div[1]/p[2]
- **Ventajas:** Especificidad máxima en la selección.
- **Desventajas:** Altamente susceptible a cambios estructurales. Menos legible y mantenible.

> Cuando los elementos son dinámicos podemos crear un Xpath con un atributo específico para hacerlo más robusto, más información acá [Xpath para elementos dinámicos]

Name

- **Significado:** Se refiere al atributo “name” de un elemento HTML.
- **Ejemplo:** <input type="text" name="usuario">
- **Ventajas:** Útil en formularios y elementos interactivos.
- **Desventajas:** No todos los elementos tienen un atributo “name”. Puede no ser único.

Elección del Identificador:

La elección del identificador dependerá del contexto y los requisitos específicos del proyecto. Algunos aspectos a considerar:

- **Unicidad:** Si hay un identificador único disponible (ID), es preferible, ya que facilita la selección precisa del elemento.
- **Flexibilidad:** Las clases permiten seleccionar grupos de elementos similares, lo que puede ser útil si se requiere la extracción de múltiples elementos.
- **Estabilidad:** Si los identificadores cambian con frecuencia, es preferible utilizar estrategias más estables, como selectores de etiquetas o XPath.
- **Eficiencia:** Seleccionar identificadores más específicos puede mejorar la

eficiencia del scraper al reducir la cantidad de datos a procesar.

En resumen, la elección del identificador dependerá de la situación específica, y es recomendable adaptar la estrategia de selección a las necesidades del proyecto.

Cómo obtener los identificadores web

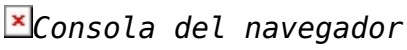
Para poder controlar las páginas web, necesitamos acceder a los identificadores de los elementos para así especificar en cuál de ellos queremos realizar acciones. En este documento veremos el paso a paso para lograrlo.

Tomaremos como ejemplo la página RPA Challenge

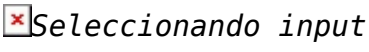
Inspeccionar la página

Primero debemos inspeccionar la web para visualizar el código fuente, para esto podemos dar click derecho en cualquier lugar y seleccionar la opción **Inspeccionar**, también podemos utilizar los atajos de teclado CTRL + SHIFT + C o F12 para Windows o Command + SHIFT + C para macOS.

Veremos una ventana como esta


Consola del navegador

Clickeamos la flecha que se encuentra en la esquina superior izquierda y una vez que esté de color azul, significa que ya podemos marcar cualquier elemento de la web. Seleccionamos el elemento deseado, y automáticamente se resaltará en azul en el código, abarcando todo el contenido del elemento elegido.


Seleccionando input

Los elementos pueden tener diferentes tipos de identificadores (id, class, name, etc.), para más detalles, revisa la siguiente entrada Tipos de Identificadores Web.

Elemento Dinámico

Podemos ver que el elemento marcado tiene **id**, **name** y **class**

```
<input _ngcontent-c2="" ng-reflect-name="labelCompanyName" id="jnPCa" name="jnPCa" class="ng-untouched ng-pristine ng-invalid">
```

Algo bastante común es que algunos de éstos sean dinámicos. Si recargamos la página y revisamos nuevamente, veremos que, tanto el id como el name

cambiaron

<input _ngcontent-c2="" ng-reflect-name="labelCompanyName" id="UDONR" name="UDONR" class="ng-untouched ng-pristine ng-invalid">

Esto sucederá cada vez que la página se recargue o se vuelva a abrir, por lo que nuestro robot fallaría si indicamos uno de estos datos, el **Xpath** tampoco nos serviría ya que como el id es dinámico, también sería diferente

//*[@id="jnPCa"]

//*[@id="UDONR"]

El **fullXpath** tampoco serviría, ya que si nos fijamos, los elementos cambian de posición

*Posición inicial del elemento*

*Posición del elemento cambiada*

Una forma óptima de controlar esto, es crear un Xpath centrándose en algún atributo específico, más detalles acá: Xpath para elementos dinámicos.

Si nos fijamos en el elemento, veremos que lo único que no cambia es el siguiente atributo

ng-reflect-name="labelCompanyName"

Podemos entonces crear un Xpath utilizando esa información, de esta forma

//input[@ng-reflect-name="labelCompanyName"]

Para saber si está correcto, podemos probarlo en el mismo navegador, donde inicialmente abrimos la consola, seguimos en **Elements** y presionamos CTRL + F si estamos en Windows o Command + F si estamos macOS, esto habilitará la casilla de búsqueda (1) y pegaremos el xpath previamente mencionado, al hacerlo nos marcará el elemento tanto en el código (2) como en la vista web (3)

*(1) Input de búsqueda*
(2) Elemento marcado en código
(3) Elemento marcado en la vista web

Si está todo correcto, entonces ese identificador es el que debemos utilizar en los comandos de Rocketbot para indicar el elemento con el que queremos trabajar.

Elemento No Dinámico

Ahora veremos el caso de un elemento donde su posición e identificadores no cambian, en este ejemplo tomaremos el botón para descargar el archivo Excel

*Botón descarga*

El elemento es el siguiente, si recargamos y lo volvemos a revisar, seguirá teniendo la misma información

<a _ngcontent-c1="" class=" col s12 m12 l12 btn waves-effect waves-light uiColorPrimary center" target="_blank" href="./assets/downloadFiles/challenge.xlsx"> Download Excel <i _ngcontent-c1="" class="material-icons right">cloud_download</i>

Obtenemos el fullXpath y lo utilizamos en el comando web de Rocketbot donde necesitemos indicar el elemento

/html/body/app-root/div[2]/app-rpa1/div/div[1]/div[6]/a

Cómo importar un Robot

Cuando nos comparten un archivo .bd o un .json o cuando queremos mover un robot de un ambiente a otro, debemos cargarlo a Rocketbot Studio, para eso debemos realizar lo siguiente:

Cargar BD

Nos vamos a la pantalla de Inicio y clickeamos el botón **Cargar DB de Proyecto**, esto nos abrirá el explorador, buscamos el archivo y lo cargamos.


Nos mostrará una lista con todos los robots dentro de esa Base de datos, seleccionamos el robot que necesitamos y ésto nos redirigirá al Dashboard donde visualizaremos el flujo.



Cargar Json

Nos vamos a la pantalla de Inicio y clickeamos el botón **Robot Nuevo**, ingresamos un nombre y lo creamos.



Una vez en el Dashboard podemos arrastrar nuestro archivo **.json** o clickear donde aparece el mensaje ***"Arrastre aquí un archivo de Rocketbot Studio (json) para editar"***, esto nos abrirá el explorador, lo buscamos y cargamos.

Nos mostrará un mensaje indicando que el nombre del robot actual cambiará, si clickeamos **OK**, dejará el nombre del robot que se está cargando, si clickeamos **Cancel**, conservará el nombre que indicamos al crear el robot nuevo.



Cómo exportar un Robot

Cuando tenemos un proyecto y queremos exportarlo para compartirlo o llevarlo al ambiente productivo, debemos realizar lo siguiente:

Ir a la pestaña **Robot** donde tendremos 2 opciones:

1. **Exportar robots a DB:** Nos mostrará un pop-up donde podremos elegir si queremos que nos descargue un **.zip** con los módulos utilizados en nuestro robot y si queremos exportar la Base de datos a Producción.

   Si tenemos robot padre y robot hijos, para exportar todo el proyecto siempre debemos realizarlo desde el robot principal

   - **Check seleccionado en "Incluir módulos":** Descargará un .zip con la carpeta modules, donde dentro se encontrarán todos los módulos que contiene el robot o proyecto.
   - **Check seleccionado en "Exportar robots a producción":** Base de datos encriptada, no permite volver a cargar a Rocketbot Studio ni ver y/o modificar sus comandos. Sólo permite la ejecución del robot.
   - **Check no seleccionado en "Exportar robots a producción":** Base de datos normal que se puede volver a cargar en Rocketbot Studio, ver su información y modificar. Permite descargar el Proyecto completo (varios bots que corresponden a un mismo proceso, robot padre e hijos).
2. **Guardar en archivo Json:** Sólo descarga un Robot, el que se tiene cargado actualmente en Rocketbot Studio.

Para cargar estos archivos a Rocketbot Studio, puedes revisar la siguiente entrada:

Cómo importar un Robot