

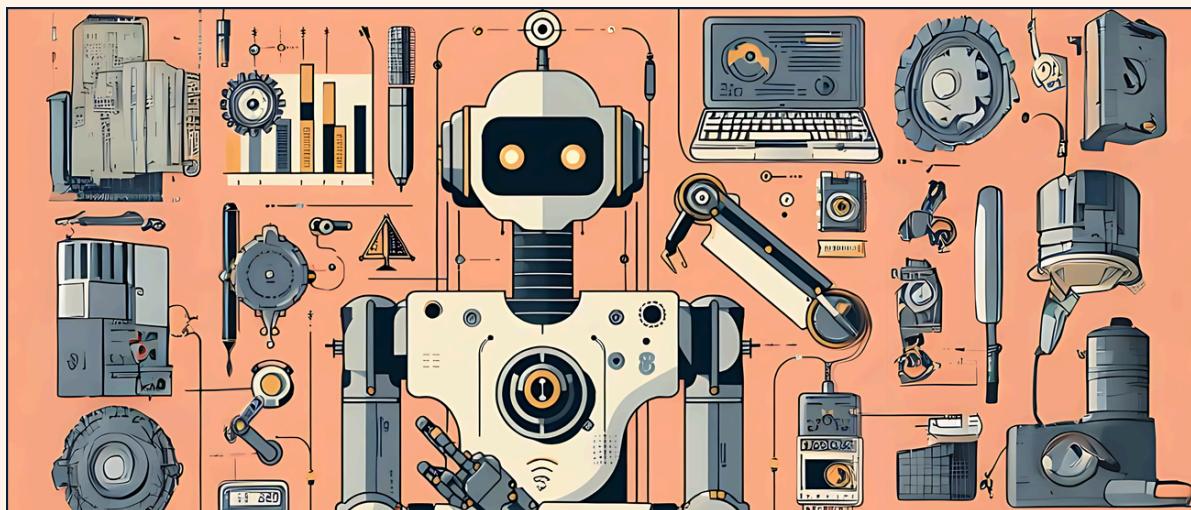
FEBRERO - MAYO 2024



# Logística Inteligente

**ROBOLOGIX**  
Manual de Usuario

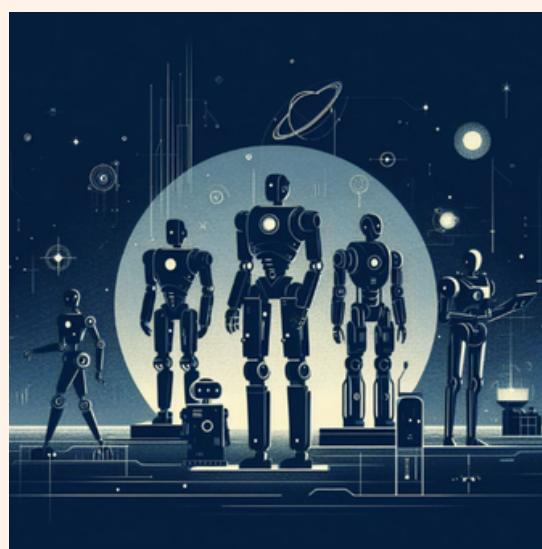
Jose Luis Galán | Alberto Andrés | Gonzalo Albelda | Gonzalo Martín



jolugaav@gmail.com

# ÍNDICE

1. Introducción para el usuario.....	3
a. Componentes principales del sistema.....	3
b. Objetivo del manual.....	4
2. Requisitos previos.....	4
3. Dependencias necesarias.....	5
4. Despliegue de la web.....	6
5. Despliegue de la API REST.....	10
6. Despliegue de la BBDD.....	11
7. Ejecutar el algoritmo.....	13
8. Lanzar los ESP32.....	13
9. Iniciar los archivos RoboDK.....	15
10. Siguientes pasos.....	16



# Introducción

Este manual de usuario está diseñado para guiarte a través de la instalación, configuración y uso de nuestro sistema de gestión de envíos automatizado y modular. Nuestra solución, desarrollada con las últimas tecnologías en comunicación y software, tiene como objetivo optimizar los procesos logísticos, permitiendo un funcionamiento continuo las 24 horas, reduciendo las cargas de trabajo manuales y aumentando la eficiencia operativa.

**Descripción general del sistema:** Nuestro sistema abarca todo el ciclo de vida de un pedido, desde su creación en la página web hasta su preparación final para el envío. Utilizamos un protocolo de comunicación MQTT para que los diferentes componentes del sistema (robots, programas y dispositivos) interactúen de manera eficiente y coordinada. Cada dispositivo ejecuta una serie de órdenes simples, asegurando una operación sincronizada sin la necesidad de conocimiento completo del proceso.

## Componentes principales del sistema

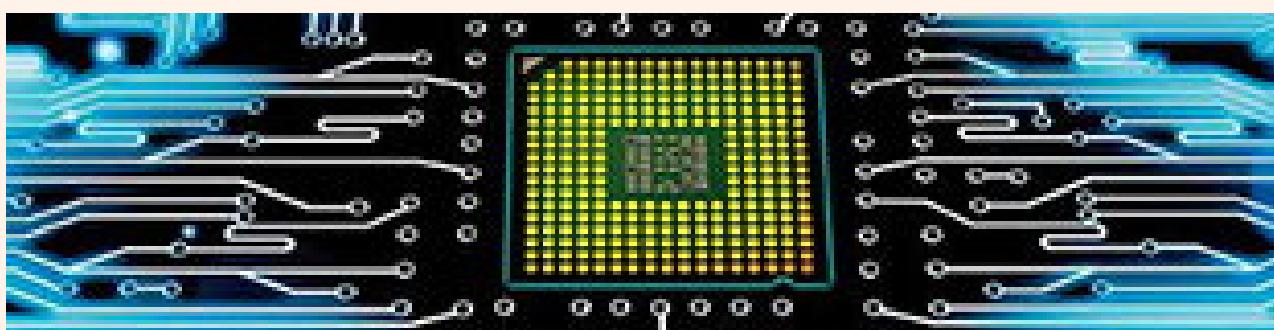
**Plataforma web:** Desarrollada con HTML, CSS y JavaScript, esta interfaz permite a los usuarios realizar pedidos de manera fácil y eficiente.

**Base de datos:** Implementada en PostgreSQL, gestiona y almacena toda la información relevante de los pedidos.

**Algoritmo de optimización:** Un algoritmo en C++ que determina la disposición óptima de los objetos en las cajas para su envío.

**Simulación robótica:** Utilizando RoboDK y scripts de Python, simulamos y controlamos los movimientos de los robots que manipulan los objetos.

**Control y sensorización:** Dos dispositivos ESP32-S3 se encargan del control del proceso y la captación de datos en tiempo real, respectivamente.



# Objetivo de este manual

Este manual tiene como propósito proporcionar a los usuarios toda la información necesaria para:

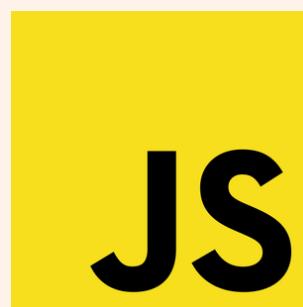
1. Instalación y configuración del sistema.
2. Operación diaria del sistema de gestión de envíos.
3. Mantenimiento y resolución de problemas comunes.

Esperamos que este manual te sea de gran ayuda para aprovechar al máximo las capacidades de nuestro sistema. Si tienes alguna pregunta o necesitas asistencia adicional, no dudes en contactarnos a través de nuestro soporte técnico.

## Requisitos previos

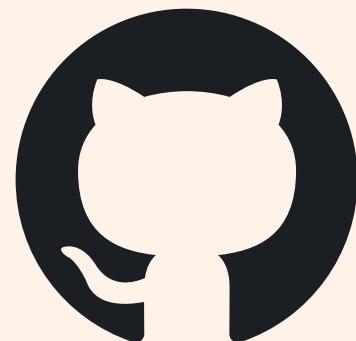
Para instalar este proyecto y utilizarlo de forma local en un ordenador, debe cumplir los siguientes requisitos:

- Navegador Web
- Entorno de Desarrollo Integrado (IDE) que soporte el uso de C++20, Python 3.12.3 y JavaScript
- PostgreSQL
- RoboDK
- Esp32-S3
- Git/GitHub (opcional)



# Descargar las dependencias

Existen varios métodos de instalación, sin embargo, el más simple y recomendado es a través de la terminal de GitHub (GitBash). Esta opción es preferida por su simplicidad, ya que con la ejecución de un solo comando se instalarán todas las dependencias necesarias para ejecutar el proyecto. Para ello, sitúate en el directorio donde deseas descargar el proyecto y ejecuta el siguiente comando en la terminal: `git clone https://github.com/jlgalavi/Automatic_stock`.



Una vez se haya descargado este repositorio en su ordenador personal, debe comprobar que todos los archivos se hayan descargado exitosamente. Es decir, su carpeta debería tener la siguiente estructura:

APP	cambios	last week
COM_RDK	Save code of projects of RoboDK	2 months ago
Comunicaciones_ESP32-S3	Merge pull request #54 from jlgalavi/Albelda	15 hours ago
RoboDK	first functional project with all parts connected	2 weeks ago
Web	Merge branch 'master' into Alberto	last month
bin	fixed carrito pc	2 weeks ago
filesTXT	first functional project with all parts connected	2 weeks ago
libs	last prove for connect algorithm	2 weeks ago
src	last prove for connect algorithm	2 weeks ago
Makefile	update new algorithms	2 weeks ago
README.md	Create README.md	3 months ago
Task.txt	Update and fixed communications mqtt & roboDK	last month
main.cpp	last prove for connect algorithm	2 weeks ago
main2.cpp	update new algorithms	2 weeks ago
notes.txt	update alberto photo	last month

# Despliegue de la web

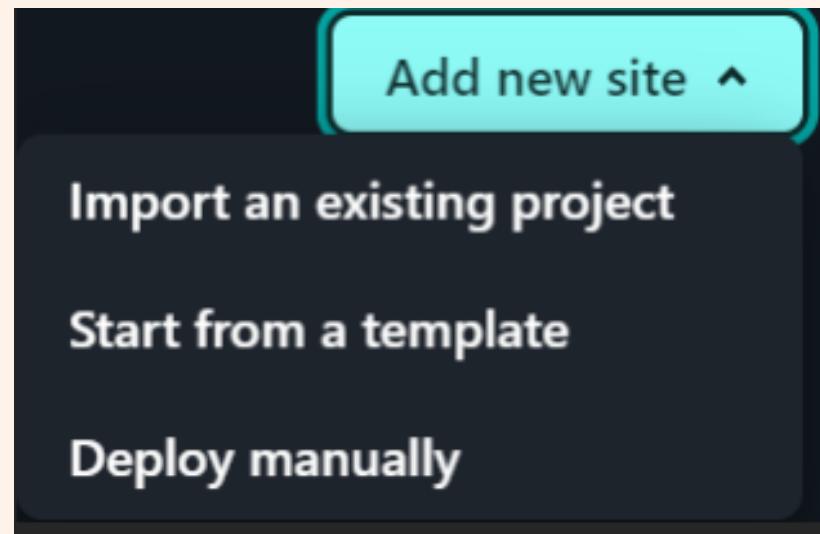
Una vez verificada la correcta descarga de todos los archivos, será necesario desplegar la web para que esté accesible para todos los usuarios desde sus respectivos dispositivos.

Para desplegar la web necesita primero subirla a la nube, esto lo podemos hacer mediante cualquier aplicación de guardado en nube.

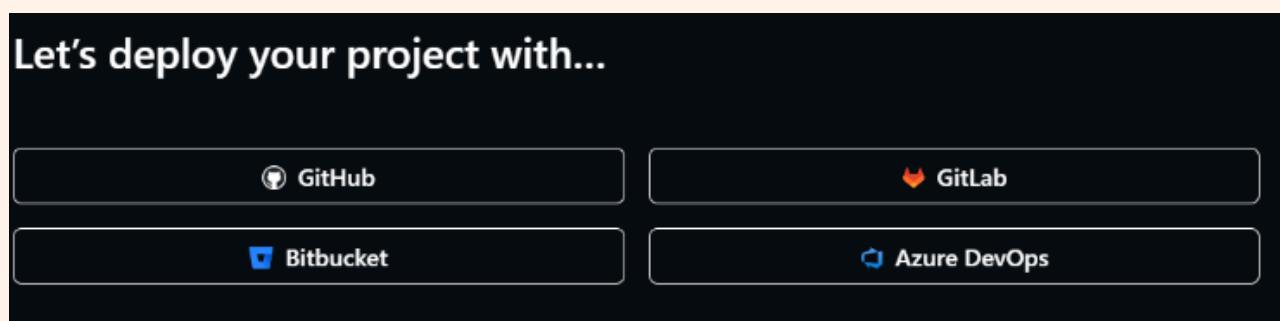
En nuestro caso, recomendamos el uso de "Netlify", que es una plataforma de alojamiento web y servicios de desarrollo centrada en la creación, implementación y administración de sitios web y aplicaciones web estáticas, así como en la automatización de flujos de trabajo de desarrollo.



Primero te tienes que crear una cuenta en Netlify. Una vez creada, vas a subir tu web de la siguiente manera:



Luego, clicando a "Import an existing project" te saldrá la siguiente ventana:



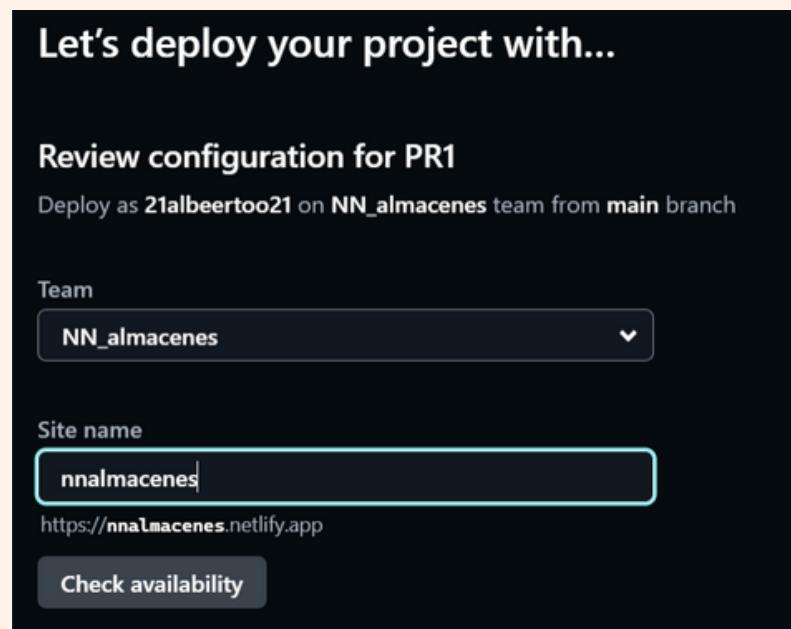
Observamos que hay 4 métodos distintos para desplegar tu proyecto, en nuestro caso, vamos a elegir el de GitHub.

Al clicar, te redirirá a GitHub para que autentiques que eres tú. Una vez comprobado, en Netlify aparecerán todos tus repositorios de Github como vemos en la siguiente imagen:



Anteriormente, deberemos tener subidos todos los archivos de la web en un repositorio en GitHub.

Una vez hayamos elegido el repositorio correspondiente nos aparecerá la siguiente ventana:



En este paso, elegiremos la url que queramos y que esté libre y desplegaremos nuestra web.

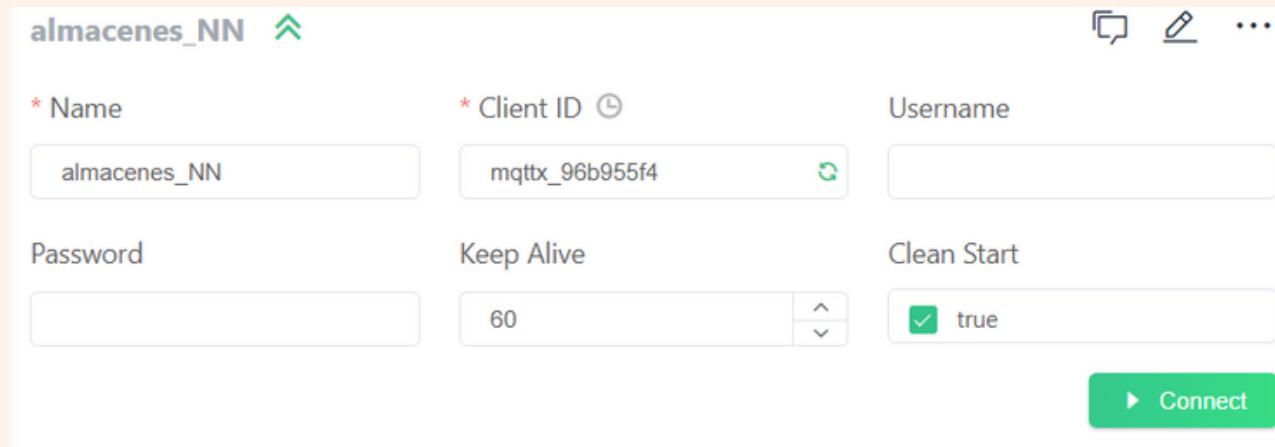
Ahora si vamos al apartado “Sites” comprobaremos que nuestra web se ha desplegado (¡Este proceso puede tardar unos minutos, así que no te alarmes si no aparece tu web!).

El último paso para tener la web operativa es configurar el MQTTX para poder transmitir los datos del pedido.

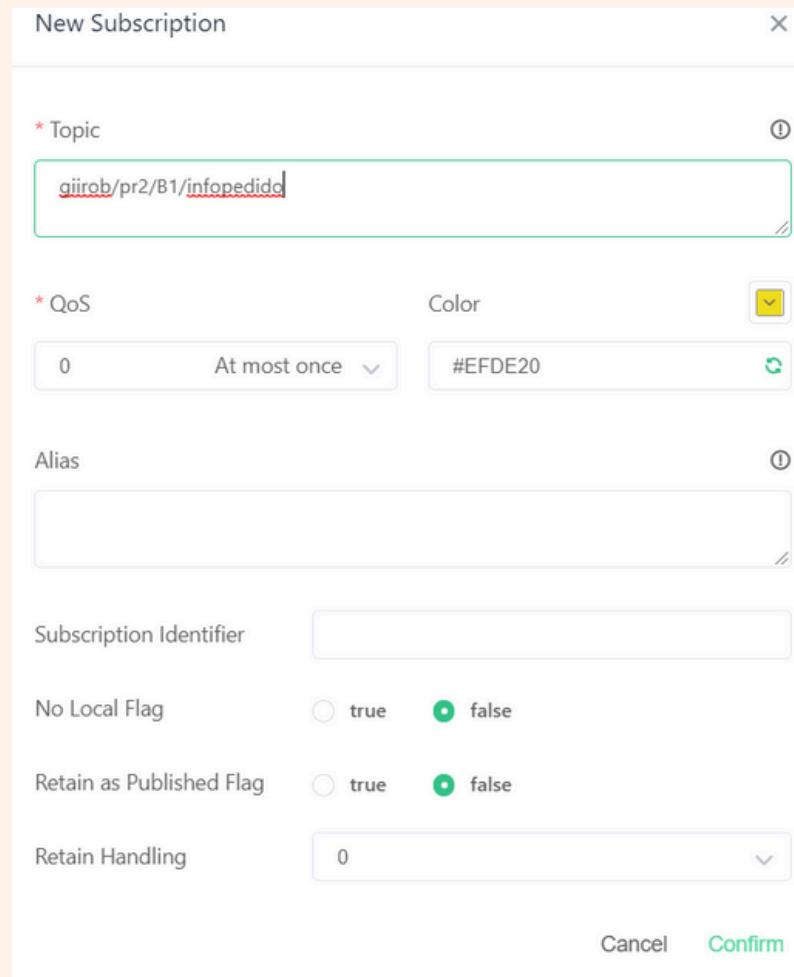
Para ello, primero, entras en la aplicación de MQTTX y te creas un servidor de la siguiente manera:

Field	Value
Name	Nombre_Servidor
Client ID	mqttx_3c5dafeb
Host	mqtt:// broker.emqx.io
Port	1883
Username	
Password	
SSL/TLS	Off

A continuación, nos conectamos al broker de MQTT para poder transmitir los datos del pedido:



y luego suscribirse al tópico “giirob/pr2/B1/infopedido” que va a ser el canal donde se van a escribir y leer los datos:



**¡Felicidades, ya tienes tu web operativa! El siguiente paso es subir la API REST a la nube.**

# Desplegar la API REST

Una vez desplegada la web en la nube, para poder conectarse con el backend necesitamos tener el código de ésta subido en la nube. En el caso de nuestro proyecto, la parte del backend la hemos hecho localmente, es decir, no la llegamos a desplegar en la web, pero a continuación les dejamos los pasos a seguir para conseguirlo:

## 1. Preparación de la API:

Empaquetar la aplicación: Asegúrate de que tu API esté empaquetada correctamente y que todas las dependencias estén especificadas en un archivo de dependencias (como package.json en el caso de Node.js).

Pruebas locales: Realiza pruebas exhaustivas de tu API en un entorno local para garantizar que funcione como se espera.

## 2. Elección de la plataforma en la nube:

Investigación: Investiga y elige la plataforma en la nube que mejor se adapte a tus necesidades. Algunas opciones comunes son AWS, Azure, Google Cloud Platform, Heroku, DigitalOcean, entre otras.

Cuenta y configuración: Crea una cuenta en la plataforma en la nube seleccionada y realiza la configuración inicial necesaria.

## 3. Despliegue de la API:

Subir el código: Sube el código de tu API a la plataforma en la nube. Esto puede hacerse a través de un repositorio de control de versiones como GitHub o directamente desde tu máquina local.

Configuración del entorno: Configura el entorno de ejecución para tu API en la plataforma en la nube. Esto puede incluir la configuración de variables de entorno, ajustes de red, escalabilidad, entre otros.

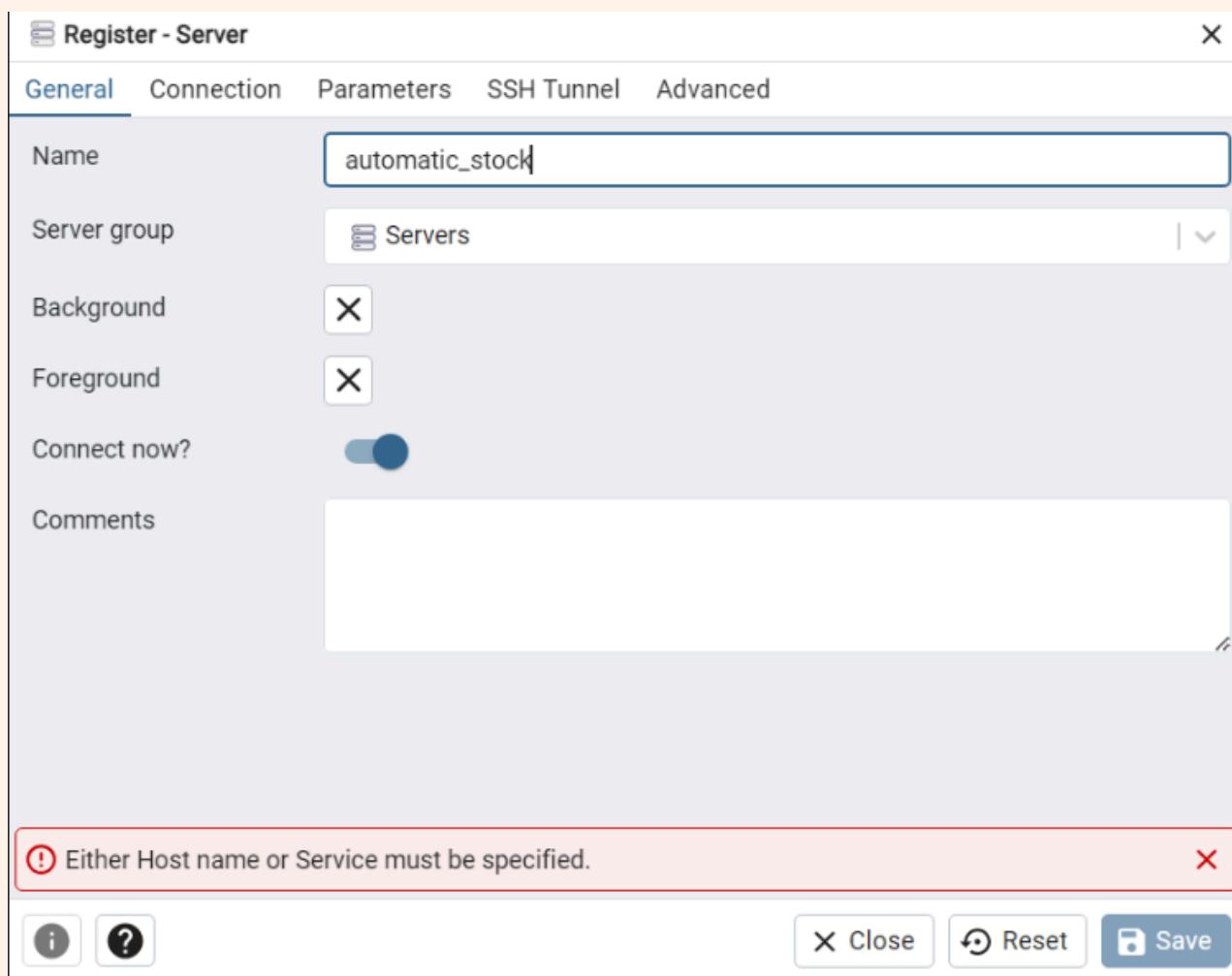
Una vez desplegada la API en la nube, podremos conectarnos con dispositivos en la nube



# Desplegar la base de datos

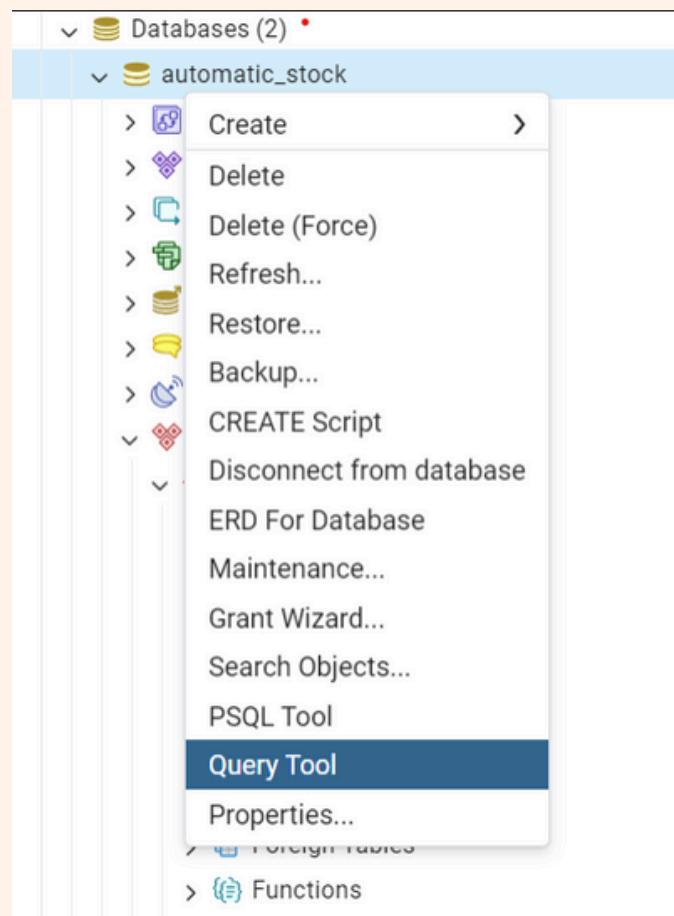
Tras el despliegue de la página web y la API, será necesario implementar la base de datos. Esta base de datos es fundamental para el proyecto, ya que almacenará la información relevante generada por la web, o sea, la necesaria para el correcto funcionamiento y seguimiento del sistema.

Primero, entramos en la aplicación para ordenador de PostgreSQL “PgAdmin 4” y nos creamos un servidor:



Una vez creada, en el mismo menú, hacemos click derecho sobre nuestra base de datos que acabamos de crear:





Seleccionamos “Query Tool” para abrir la terminal de SQL para poder escribir código. Aquí, crearemos en lenguaje SQL la tabla pedido con sus atributos:

```
CREATE TABLE pedido (
    id_pedido SERIAL PRIMARY KEY,
    cajas_s INTEGER,
    cajas_m INTEGER,
    cajas_l INTEGER,
    cajas_xl INTEGER,
    precio_total NUMERIC(10, 2),
    fecha_pedido TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Una vez escrito, escogemos con el ratón todo el código y pulsamos F5, si has seguido los pasos correctamente tu tabla pedido se habrá creado correctamente.

La correcta implementación y despliegue de la base de datos garantizará la integridad y disponibilidad de los datos para su posterior procesamiento y análisis.

# Ejecución del algoritmo

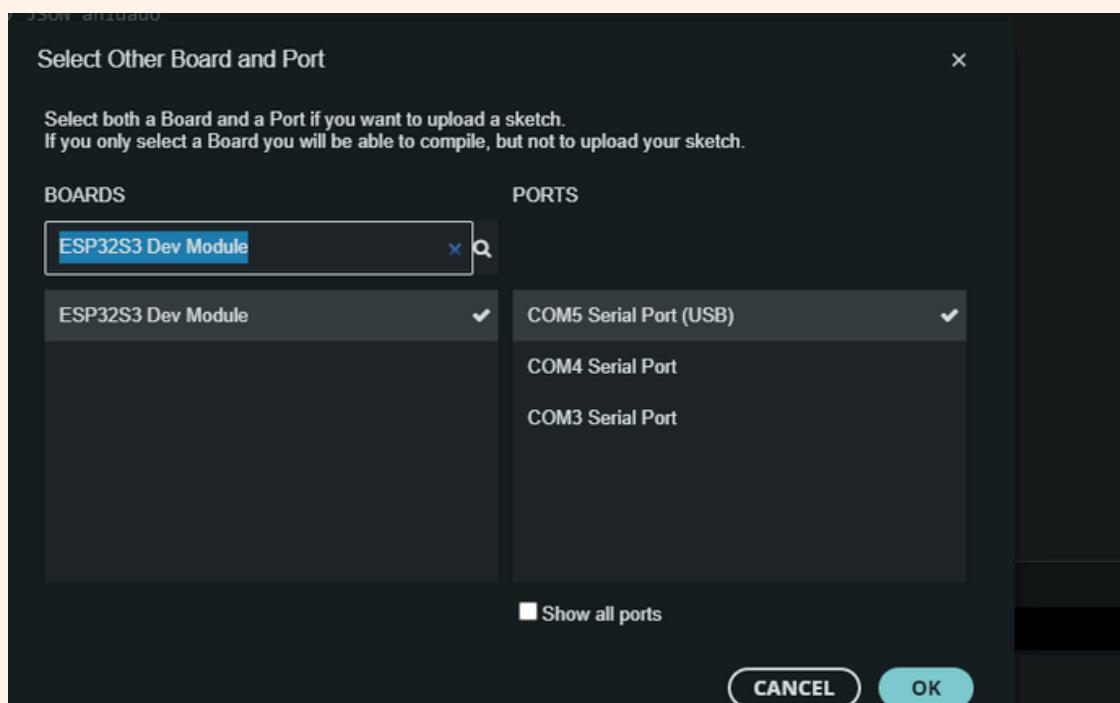
Para iniciar las operaciones relacionadas con el pedido, como el cálculo de las posiciones de las cajas en los contenedores, es necesario que el algoritmo se ejecute continuamente, a la espera de la llegada de un pedido. Esto se realizará a través de la terminal del repositorio del proyecto mediante el siguiente comando:

```
make all  
./bin/main
```

El comando anterior ejecutará el archivo de forma continua, ya que el archivo main contiene un bucle while que espera la llegada de una orden.

## Lanzar los ESP-32

Conecte los dispositivos ESP32-S3 a un portátil con un cable USB, necesario para transferir el código. Utilice un entorno de desarrollo como Arduino IDE para cargar el código en los ESP32-S3, asegurándose de tener todos los archivos y librerías necesarios. Cuando todo esté preparado, seleccione el puerto COM del ESP32-S3 y cargue el código:



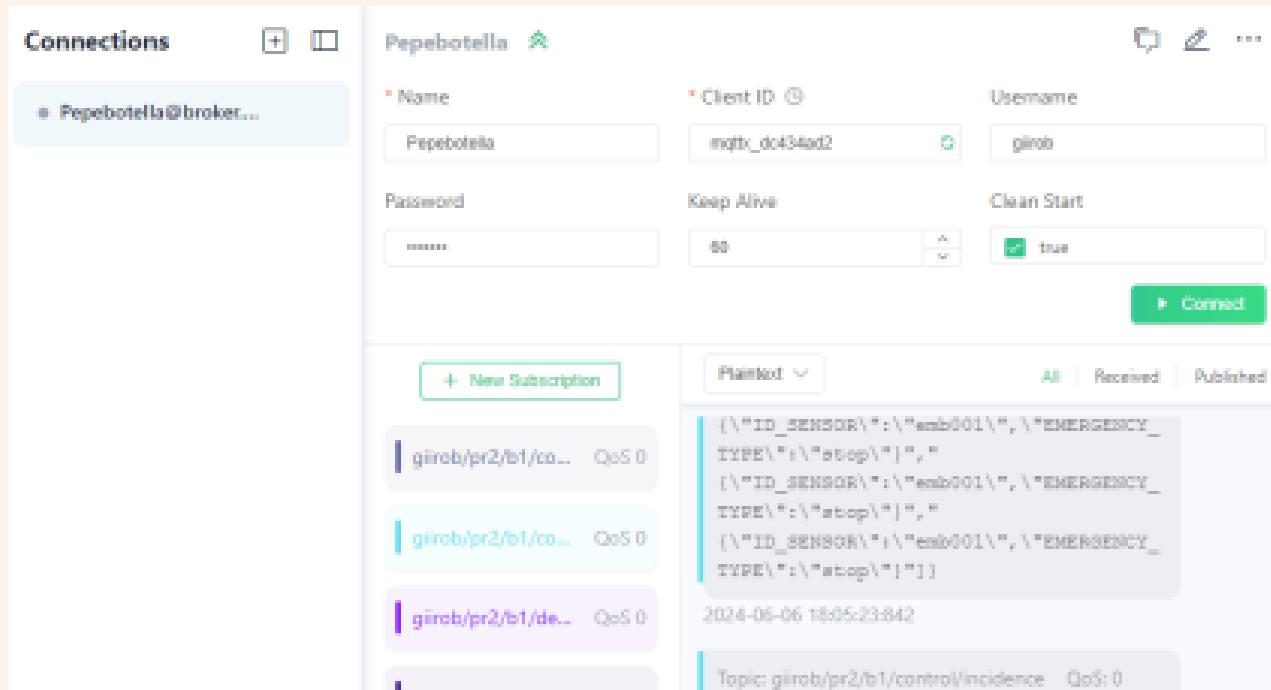
```

ESP32 PROYECTO - f_funciones.ino | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32S3 Dev Module Upload
Upload PROYECTO.ino Código_QR.ino Config.h c_log
56     bool llena(Buffer_Circ* buff)
57     {
58         return buff->contador == buff->capacidad;
59     }
60
61     void listanddelete(Buffer_Circ* buff, String
62     {

```

Después de esta carga, configure los ESP32-S3 para conectarse a la red Wi-Fi, incluyendo las credenciales (SSID y contraseña) en el código, y verifique la conexión mediante mensajes de estado en el monitor serie. Con los dispositivos ahora conectados a la red, comenzarán a ejecutar el código en un bucle continuo, esperando nuevas órdenes y procesando datos sin interrupción.

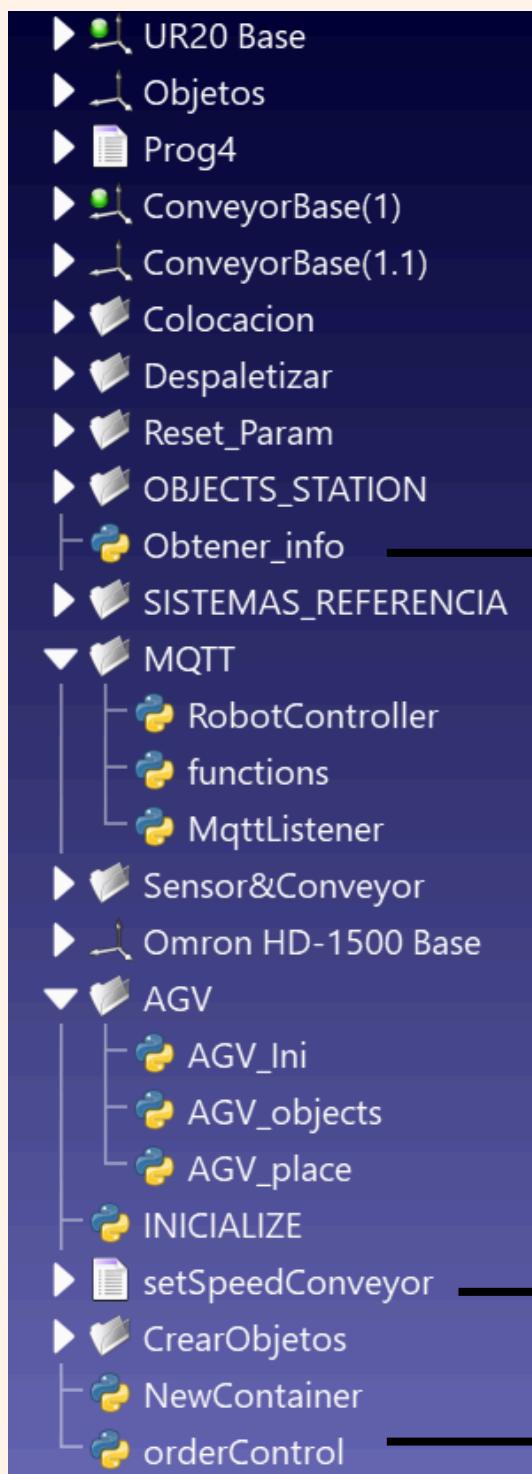
Luego en la app de MQTTX nos conectamos al broker y nos conectamos a los tópicos necesarios



Finalmente, supervise regularmente los ESP32-S3 usando herramientas de monitoreo para revisar registros y validar el rendimiento, y realice actualizaciones de firmware y código según sea necesario para mejorar el rendimiento o solucionar problemas.

# Iniciar archivos de RoboDK

Por último, en la parte del simulador del RoboDK necesitamos iniciar los siguientes archivos:



Para recibir la información de la orden

Llamada a los archivos que inicializan todos los elementos de la simulación

Para inicializar el flujo de órdenes

# Siguientes Pasos

Si experimenta algún problema cuya solución no se encuentra contemplada en el presente documento, no dude en ponerse en contacto con nuestro soporte técnico. Puede enviar un correo electrónico a [contacto@robologix.com](mailto:contacto@robologix.com), donde nuestro equipo especializado estará disponible para asistirle.

Nos comprometemos a proporcionarle una solución efectiva y rápida a cualquier inconveniente que pueda surgir. Nuestro objetivo es asegurar que su experiencia con nuestro sistema sea lo más fluida y satisfactoria posible. Por favor, incluya todos los detalles relevantes sobre su problema en el correo para que podamos brindarle una ayuda más precisa y eficiente.



## TRABAJO HECHO POR



Gonzalo Martín



Gonzalo Albelda



José Luís Galán



Alberto Andrés

## Agradecimientos

Gracias por utilizar nuestro producto.  
Esperamos que este manual le haya  
sido de ayuda.

## Versión de documento

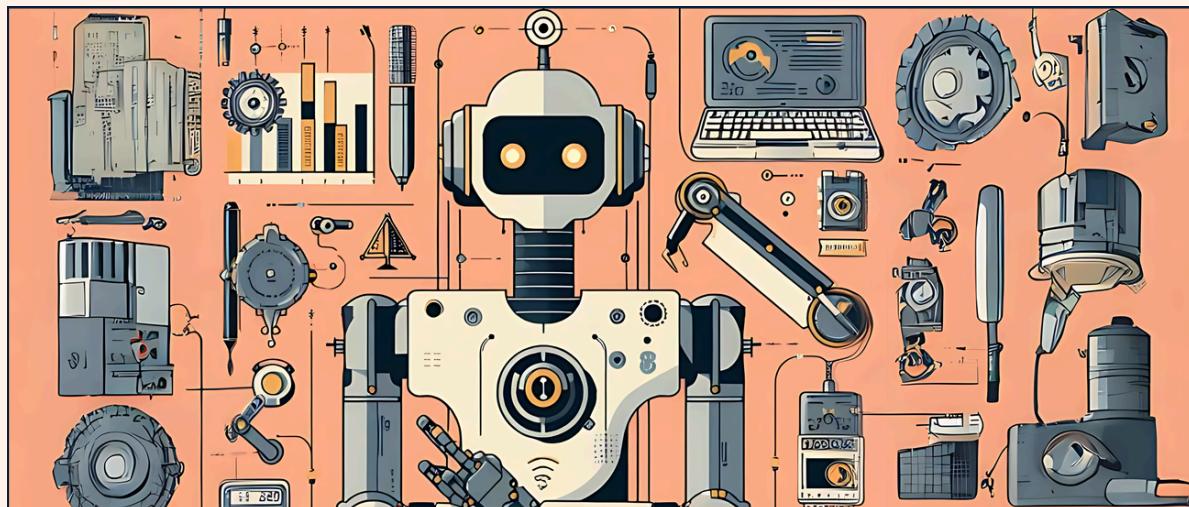
Versión 1.0, Actualizado en junio de  
2024.

## Información de contacto

**Correo electrónico:** almacenes\_nn@gmail.com

**Teléfono:** (34) 722 200 422

**Sitio web:** nn-almacenes-inteligentes.netlify.app



© 2024 Almacenes NN. Todos los derechos reservados. Este documento  
está protegido por derechos de autor y no puede ser reproducido,  
distribuido o utilizado sin permiso explícito de Almacenes NN.