



Tecnológico de Monterrey

Materia:

Automatización de sistemas de manufactura

Docente:

Juan Gabino Díaz Martínez

Alumnos:

Héctor Antonio de la Torre Robles (A01412443)

Scarlet Margarita Gutierrez Castro (A01411920)

Joel Andre Contreras Torres (A01412738)

Guillermo Asael Arteaga Torres (A01411592)

Jose Alberto Elizondo Willis (A01412738)

Proyecto:

Control wireless UR3

Fecha de entrega:

14 de Junio del 2024

ÍNDICE

Introducción.....	3
PRIMER EQUIPO.....	5
Matlab.....	5
Add Ons.....	5
Funciones.....	5
Código.....	5
ESP32.....	11
Declaraciones.....	11
Código.....	12
Funciones.....	13
Monitor Serial.....	15
Configuración.....	16
AdaFruit.....	17
Llave de seguridad.....	17
Feeds.....	18
Dashboards.....	19
AWS.....	23
Servicio IoT.....	23
Node-Red.....	30
Descarga Node Red.....	30
Servidor Node-Red.....	31
SEGUNDO EQUIPO.....	41
AWS.....	41
Recuperación de datos.....	41
Node-Red.....	42
Conexión.....	42
Entrada a PLC S7.....	43
Salida MQTT a PLC.....	47
PLC / Siemens S7-1500.....	50
Configuración PLC.....	50
UR3.....	54
Conclusiones.....	55
Futuras Mejoras.....	56
Bibliografías.....	57

Introducción

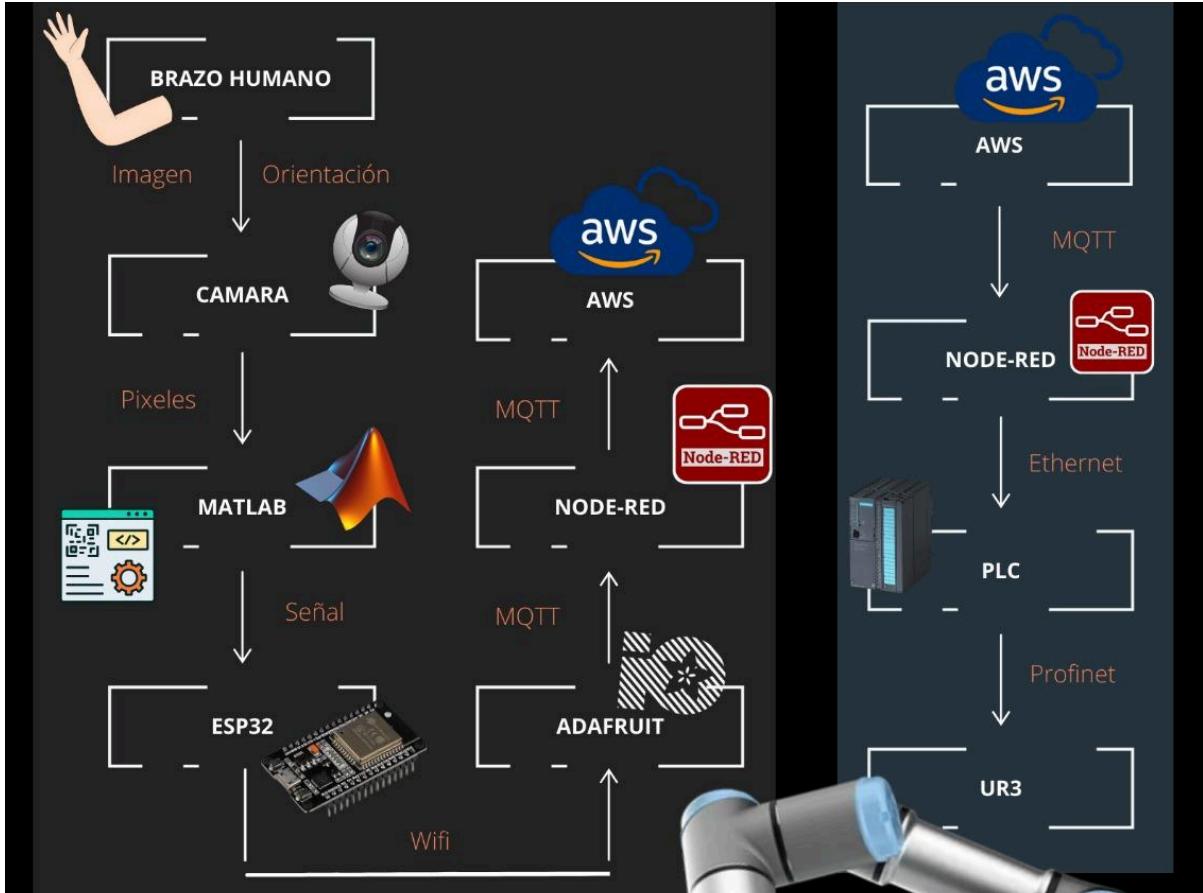
En el contexto de la automatización avanzada y la integración de tecnologías, este proyecto explora la conexión entre MATLAB y un robot UR3, utilizando reconocimiento de imágenes y comunicación en red para ejecutar tareas específicas. El objetivo principal del proyecto es desarrollar un sistema que detecte la orientación de un objeto, en este caso, una manga, y que, al identificar que está en posición vertical, envíe una señal de activación a través de una serie de plataformas interconectadas.

El proceso comienza con el uso de MATLAB para el reconocimiento de imágenes. A través de algoritmos de procesamiento de imágenes, MATLAB detecta la orientación de la manga. Cuando se determina que la manga está en posición vertical, MATLAB envía un dato booleano mediante comunicación serial a un microcontrolador ESP32. Este dato se transmite vía Wi-Fi a un broker de datos llamado Adafruit, utilizando el protocolo MQTT.

Una vez en Adafruit, el dato se reenvía a Node-Red, una plataforma de desarrollo para aplicaciones basadas en eventos. Node-Red envía el dato nuevamente utilizando MQTT a una base de datos en la nube, como Amazon Web Services (AWS). Esto permite el almacenamiento y la recuperación de datos desde cualquier parte del mundo.

En el siguiente paso, otra computadora recupera el dato almacenado en AWS y lo transmite de vuelta a Node-Red. Desde Node-Red, el dato se redirecciona a TIA Portal de Siemens utilizando Profinet, un protocolo de comunicación industrial. TIA Portal gestiona la comunicación con un PLC, asignando el dato recibido a los registros de entradas y salidas del robot UR3. Cuando el bit de activación es recibido, se inicia una secuencia predefinida en el UR3, permitiendo la automatización de tareas específicas basadas en la detección de imágenes.

Este proyecto demuestra la integración efectiva de múltiples tecnologías y plataformas, proporcionando una solución robusta y flexible para la automatización y el control remoto de sistemas robóticos.



El proceso se inicia con el uso de MATLAB para el reconocimiento de imágenes. A través de algoritmos de procesamiento de imágenes, MATLAB detecta la orientación de la manga. Cuando se determina que la manga está en posición vertical, MATLAB envía un dato booleano mediante comunicación serial a un microcontrolador ESP32. Este dato se transmite vía Wi-Fi a un broker de datos llamado Adafruit, utilizando el protocolo MQTT.

Una vez en Adafruit, el dato se reenvía a Node-Red, una plataforma de desarrollo para aplicaciones basadas en eventos. Node-Red envía el dato nuevamente utilizando MQTT a una base de datos en la nube, como Amazon Web Services (AWS). Esto permite el almacenamiento y la recuperación de datos desde cualquier parte del mundo.

En el siguiente paso, otra computadora recupera el dato almacenado en AWS y lo transmite de vuelta a Node-Red. Desde Node-Red, el dato se redirecciona a TIA Portal de Siemens utilizando Profinet, un protocolo de comunicación industrial. TIA Portal gestiona la comunicación con un PLC, asignando el dato recibido a los registros de entradas y salidas del robot UR3. Cuando el bit de activación es recibido, se inicia una secuencia predefinida en el UR3, permitiendo la automatización de tareas específicas basadas en la detección de imágenes.

PRIMER EQUIPO

Matlab

Add Ons



Computer Vision Toolbox version 23.2

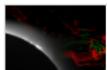


Image Acquisition Toolbox version 23.2



Image Acquisition Toolbox Support Package for OS Generic Video Interface version 23.2.0

Estos son los 3 Add Ons necesarios para el procesamiento de imágenes dentro del programa a realizar para el reconocimiento de formas, colores y orientación.

Funciones

BrazoNaranja.m X | generadorMascaraNaranja.m X | ColorConverter.m X |

Las funciones cargadas se encuentran en la carpeta de OneDrive, con la cual en caso de ser necesarias, se pueden ver, cada una de ellas vienen documentadas para su uso.

Código

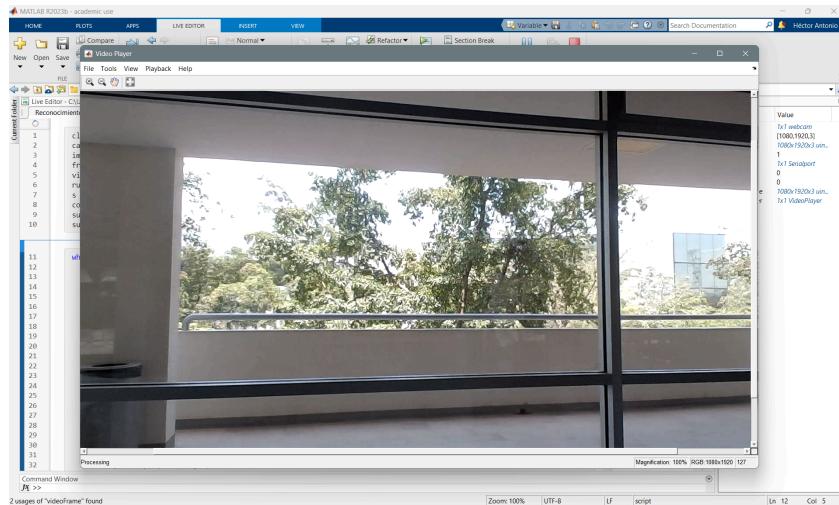
```
clear; clc; close all;
cam = webcam();
im = snapshot(cam);
frameSize = size(im);
videoPlayer = vision.VideoPlayer('Position',[100 100 [frameSize(2), frameSize(1)+30]]);
runLoop = true;
s = serialport('COM3', 115200);
configureTerminator(s, "LF");
suma1 = 0;
suma0 = 0;
```

Esta parte del código se encarga de limpiar todo cómo declarar los objetos necesarios para lo que se ocupara para el reconocimiento de las imágenes y código mas adelante, se realiza la configuración inicial de la cámara, como la grabación de imágenes, el ajuste de las mismas, se declara el serial,

aquí debemos de estar al tanto en que puerto conectaremos nuestro ESP32 y recordar que no podremos abrir el monitor serial en arduino mientras estemos conectados a matlab

```
while runLoop
    videoFrame = snapshot(cam);
    %A= videoFrame;
    [bit,finalFrame,lines] = BrazoNaranja(videoFrame);
    %imshow(finalFrame)
    bit;
    if bit == 1
        suma1 = suma1 + 1;
        if suma1 == 5
            write(s, 1, 'int8')
            suma1=0;
        end
    elseif bit == 0
        suma0 = suma0 + 1;
        if suma0 == 5
            write(s, 0, 'int8')
            suma0 =0;
        end
    end
    step(videoPlayer,finalFrame);
    runLoop = isOpen(videoPlayer);
end
```

Este es el código que correremos, se realiza un bucle en el que se están tomando y reconociendo las imágenes y mandando a llamar a ciertas funciones para el reconocimiento de ángulos y grados para, estos datos son mandados a un contador para poder mandar menos datos en lugar de estar mandando muchos datos, ya que esto puede llegar a bloquear nuestra cuenta de AdaFruit, entonces, cada que el programa registre 5 datos, este mandará 1 dato a AdaFruit para continuar con la comunicación.



Aquí podemos observar ya en funcionamiento lo que es la cámara de nuestro código con la que se buscará el reconocimiento de las imágenes y video.

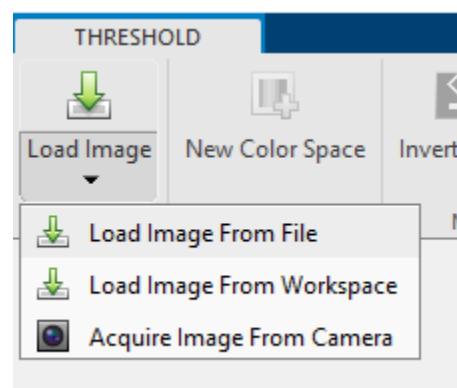


Vemos en ejemplo, cómo es que la cámara es capaz de reconocer la orientación de nuestro objeto y trazar la línea buscando el ángulo para poder mandar datos a través de la red. Las funciones consisten en la detección de color, basada en la app Color Thresholder de Matlab, que nos facilita la segmentación por color a una imagen binarizada.

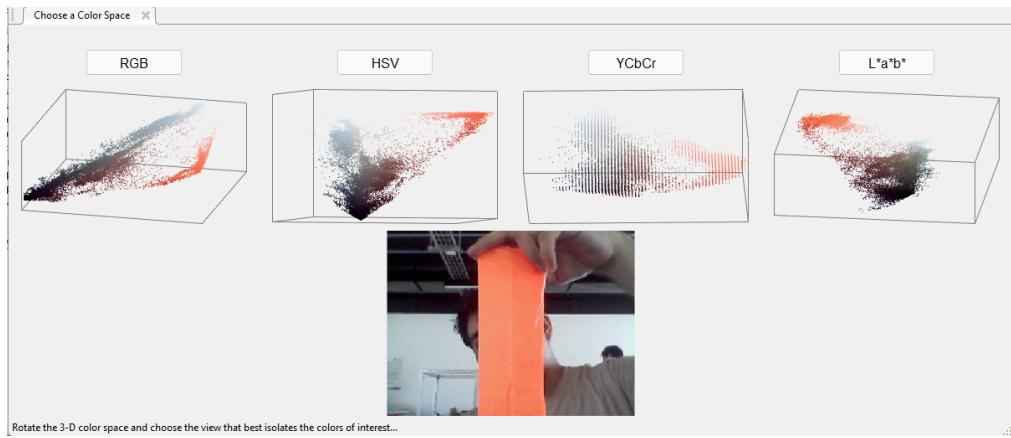
Para ello es necesario tomar una foto inicial de la manga de color en este caso, y guardarla en formato “jpg”, posteriormente la podemos llevar a la app en Matlab.



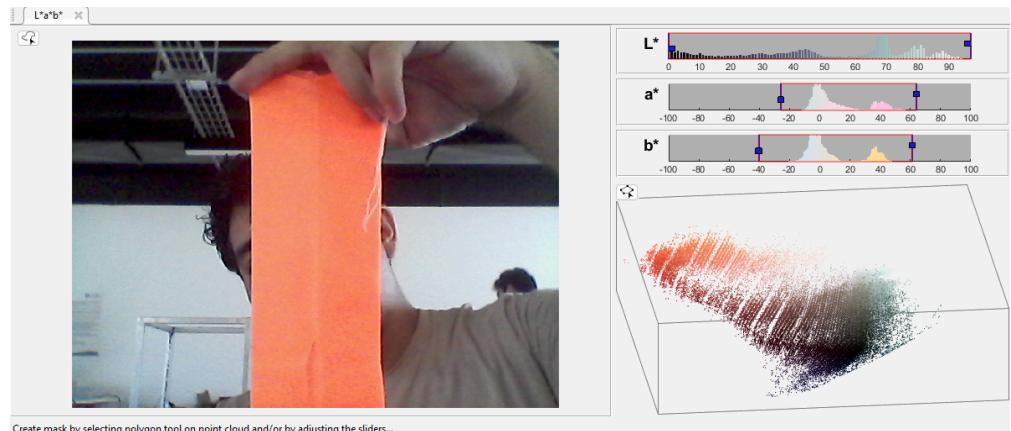
Cargamos la imagen desde archivo, para colocar la foto guardada:



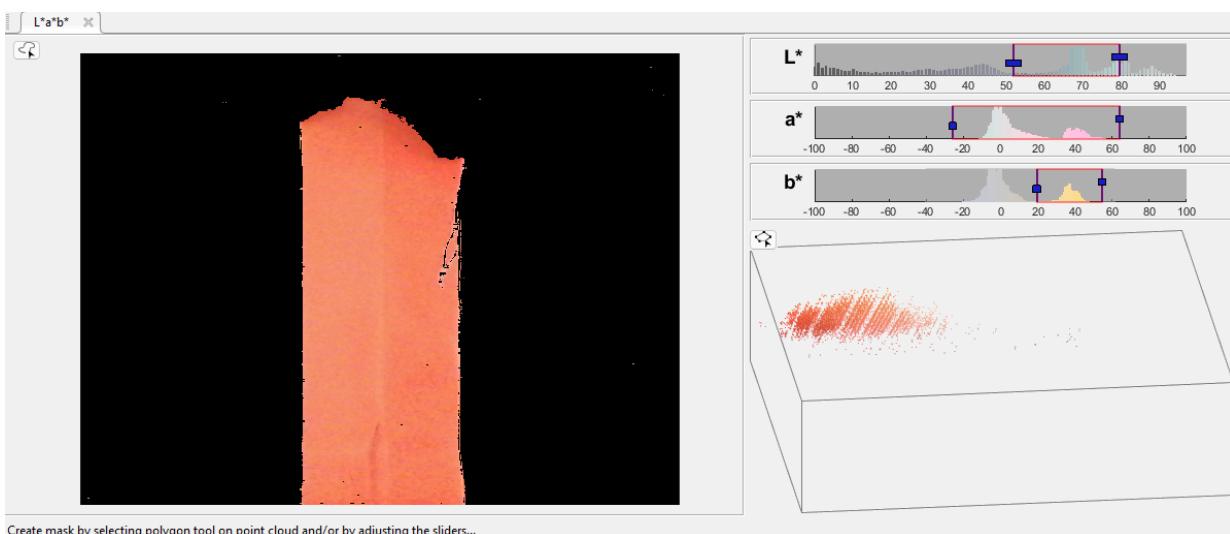
Al hacer esto logramos segmentar la imagen por color de nuestra preferencia, como sugerencia se optó por usar el formato Lab para facilitar el rango de colores a segmentar.



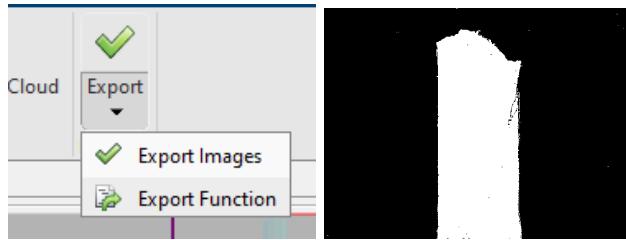
Ajustando los parámetros podemos discernir el color de nuestra manga a la imagen ideal para binarizar.



Con esto logramos separar el color correctamente de nuestro entorno:



Con esto listo solo es necesario exportar el resultado en forma de función para poder replicarlo en una toma de video en tiempo real:



Esta es la función para segmentar por color, solo es necesario dar como variable de entrada cualquier foto y el resultado será la imagen binarizada, así como la imagen original.

```
function [BW,maskedRGBImage] = generadorMascaraNaranja(RGB)
I = rgb2lab(RGB);
% Define thresholds for channel 1 based on histogram settings
channel1Min = 8.717;
channel1Max = 97.059;
% Define thresholds for channel 2 based on histogram settings
channel2Min = 31.460;
channel2Max = 60.611;
% Define thresholds for channel 3 based on histogram settings
channel3Min = 21.221;
channel3Max = 53.948;
% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min) & (I(:,:,3) <= channel3Max);
BW = sliderBW;

% Initialize output masked image based on input image.
maskedRGBImage = RGB;
% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;
end
```

Ahora ya teniendo la función para binarización, dentro de nuestro ciclo while la usamos para dentro de esta función para generar la imagen final que reconoce la orientación del brazo.

```
[bit,finalFrame,lines] = BrazoNaranja(videoFrame);
```

Esta función toma como parámetro de entrada la imagen tomada, usa la función de máscara naranja para binarizar, y hace una limpieza de la imagen para evitar errores.

```
function [bit,finalFrame,lines] = BrazoNaranja(videoFrame)
A= videoFrame;
[BW,~] = generadorMascaraNaranja(A);
im6 = imfill(BW,"holes");
j1 = bwareaopen(im6,300);
BW3 = bwmorph(j1,"skel",25);
[H,theta,rho] = hough(BW3);
peaks = houghpeaks(H,1);
lines = houghlines(BW3,theta,rho,peaks);
numLines = length(lines);
bit = 0;
```

Usamos la función bwmorph para poder esqueletizar la imagen, de manera que sea más sencillo reconocer la orientación en una delgada línea, a su vez usamos la función hough para hacer su respectiva transformación. La transformación de Hough en MATLAB permite detectar líneas rectas en una imagen. Esto se logra transformando los puntos de la imagen a un espacio de parámetros donde las líneas se representan como picos. Estos picos se analizan para determinar la presencia de líneas en la imagen original.

Para detectar solo la orientación del brazo solo colocamos como parámetro que encuentre una sola línea recta, de manera que no se confunda con ruido. Detectamos el número de líneas halladas y si este es mayor que 1 se aplica la graficación de una línea recta sobre los puntos detectados en las coordenadas de la foto. Si no detecta línea la foto resultado será la imagen normal.

```
if numLines==1
    xy = [lines(1).point1; lines(1).point2];
    finalFrame= insertShape(A, 'Line', [xy(1,:); xy(2,:)], ...
                           'LineWidth', 2, 'Color', 'red');
    X1 = lines.point1(1);
    Y1= lines.point1(2);
    X2 = lines.point2(1);
    Y2 = lines.point2(2);
    dy = Y2 - Y1;
    dx = X2 - X1;
    angulo_radianes = atan2(dy, dx);
    angulo_grados = rad2deg(angulo_radianes);
    %r =abs(angulo_radianes);
    g =abs(angulo_grados);
    if (g > -30) && (g < 30)
        bit = 0;
        %disp('Posición A');
    elseif (g > 70) && (g < 120)
        bit = 1;
        %disp('Posición B');
    end
else
    finalFrame = videoFrame;
end
end
```

En el caso de si hay una línea detectada, usamos los valores de los dos puntos en X-Y de la línea recta. Con una simple operación logramos calcular el ángulo de inclinación de nuestro brazo para posteriormente transformarlo a grados, esto nos permite saber mediante la aplicación de una zona muerta (rango de valores) si la posición del brazo está en horizontal o vertical. Si se detecta la posición vertical enviamos un valor bit de 1 como dato de salida de la función que será usado para escribirlo en el ESP32 al detectar varios seguidos. Con esto tenemos los resultados de las funciones que se ejecutan dentro de nuestro programa y nos permiten ver en la reproducción de video.

```
clear cam;
release(videoPlayer);
clear s;
```

Esta parte del código se encarga de cerrar la cámara como la comunicación sería, permitiéndonos seguir usando la comunicación como sea que lo queramos.

ESP32

Declaraciones

En la siguiente imagen se muestra el código de arduino que debe de ser cargado al ESP32, se explica a continuación ciertos parámetros a cambiar:

```
////////////////////////////////////////////////////////////////DEFINICIONES////////////////////////////////////////////////////////////////
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_Sensor.h>
// Credenciales Red WiFi
#define WIFI_SSID "_____"
#define WIFI_PASSWORD "_____"
// Credenciales Adafruit
#define ADAFRUIT_USER "Atorony23"
#define ADAFRUIT_KEY "_____"
// Servidor
#define ADAFRUIT_SERVER "io.adafruit.com"
#define ADAFRUIT_PORT 1883
char ADAFRUIT_ID[30];
// Publicar
#define ADAFRUIT_DATOS ADAFRUIT_USER "/feeds/datos"
// Definimos las variables en las que guardaremos|
int a;
// Define el led del ESP
#define pin_led 22
```

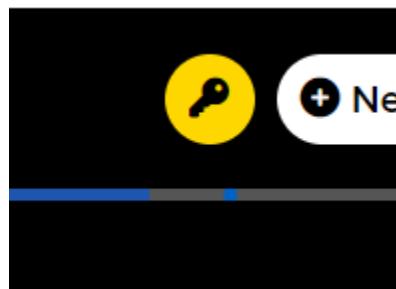
WIFI_SSID: Se modifica en base a la red Wifi a la que estemos conectados, se coloca el nombre de la misma

WIFI_PASSWORD: Se modifica en base a la contraseña de wifi a la que estamos conectando el dispositivo

ADAFRUIT_USER: Aquí escribimos el usuario de AdaFruit de nuestra cuenta

ADAFRUIT_KEY: Esta llave la conseguimos abriendo la cuenta de Adafruit (Véase más adelante en la sección de AdaFruit)

ADAFRUIT_DATOS ADAFRUIT_USER: Aquí ponemos el feed (Véase más adelante en la sección de AdaFruit) este debemos de tener bien en cuenta que se coloca como /feeds/nombredelfeedcreado



Esta es la llave donde conseguiremos las Keys necesarias para el ESP como para Node-Red

programs and scripts will need to be manually changed to the new key.

Username	Atorony23
Active Key	aio_ <input type="text"/>
REGENERATE KEY	

[Hide Code Samples](#)

Tener a consideración que esta llave es privada y única de cada usuario, esta es la que podremos utilizar en distintos lugares.

```
////////////////////////////////////////////////////////////////////////
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
////////////////////////////////////////////////////////////////////////
/////////////////////////////// DECLARAMOS FUNCIONES ///////////////////
void setup_wifi();
void reconnect();
void mqtt_publish(String feed, int val);
void get_MQTT_ID();
/////////////////////////////// DECLARAMOS FUNCIONES ///////////////////
////////////////////////////////////////////////////////////////////////
```

Estos fueron los últimos parámetros a establecer para poder tener la conexión con AdaFruit

Código

```
////////////////////////////////////////////////////////////////////////
void setup()
{
    Serial.begin(115200);
    delay(10);

    pinMode(pin_led, OUTPUT);
    digitalWrite(pin_led, LOW);

    get_MQTT_ID();
    setup_wifi();
    client.setServer(ADAFRUIT_SERVER, ADAFRUIT_PORT);

    digitalWrite(pin_led, HIGH); delay(200);digitalWrite(pin_led, LOW); delay(200);
    digitalWrite(pin_led, HIGH); delay(200);digitalWrite(pin_led, LOW); delay(200);
    digitalWrite(pin_led, HIGH); delay(200);digitalWrite(pin_led, LOW); delay(200);
    digitalWrite(pin_led, HIGH); delay(200);digitalWrite(pin_led, LOW); delay(200);
}

}
```

Se establecen los parámetros que y configuraciones iniciales del setup

```

void loop()
{
    if (!client.connected())
    {
        reconnect();
    }

    client.loop();

    if (Serial.available() > 0) { // Si hay datos disponibles en el puerto serial
        int a = Serial.read(); // Lee el carácter recibido
        if (a > 0){
            Serial.print("Dato: ");Serial.print(a);
            //Publicar
            mqtt_publish(ADAFRUIT_DATOS, a);
            delay(5000);
            a = 0;
            mqtt_publish(ADAFRUIT_DATOS, a);
        }
    }
}

//*****

```

Lo que se realiza es mandar a llamar las funciones que mantendrán la conexiones constantes con AdaFruit

Funciones

```

///////////Función para Publicar por MQTT/////////
void mqtt_publish(String feed, int val){
    String value = String(val);
    if(client.connected()){
        client.publish(feed.c_str(), value.c_str());
        Serial.println("Publicando al tópico: " + String(feed) + " | mensaje: " + value);
    }
}
///////////Función para Publicar por MQTT/////////

```

Esta función es dedicada para publicar datos en AdaFruit

```

//*****
/////////////////Función para configurar Wifi///
void setup_wifi()
{
    delay(10);

    // Nos conectamos a nuestra red Wifi
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(String(WIFI_SSID));

    //Serial.println(ssid);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("Conectado a red WiFi!");
    Serial.println("Dirección IP: ");
    Serial.println(WiFi.localIP());
}

//*****
/////////////////Función para configurar Wifi///
//*****

```

Esta función establece la conexión a wifi

```

//*****
/////////////////Capturar el ChipID para Id de MQTT///
void get_MQTT_ID(){
    uint64_t chipid = ESP.getEfuseMac();
    sprintf(ADAFRUIT_ID, sizeof(ADAFRUIT_ID), "%llu", chipid );
}
//*****
/////////////////Capturar el ChipID para Id de MQTT///
//*****

```

Esta función captura el ID para poder comunicar por medio de MQTT a AdaFruit

```

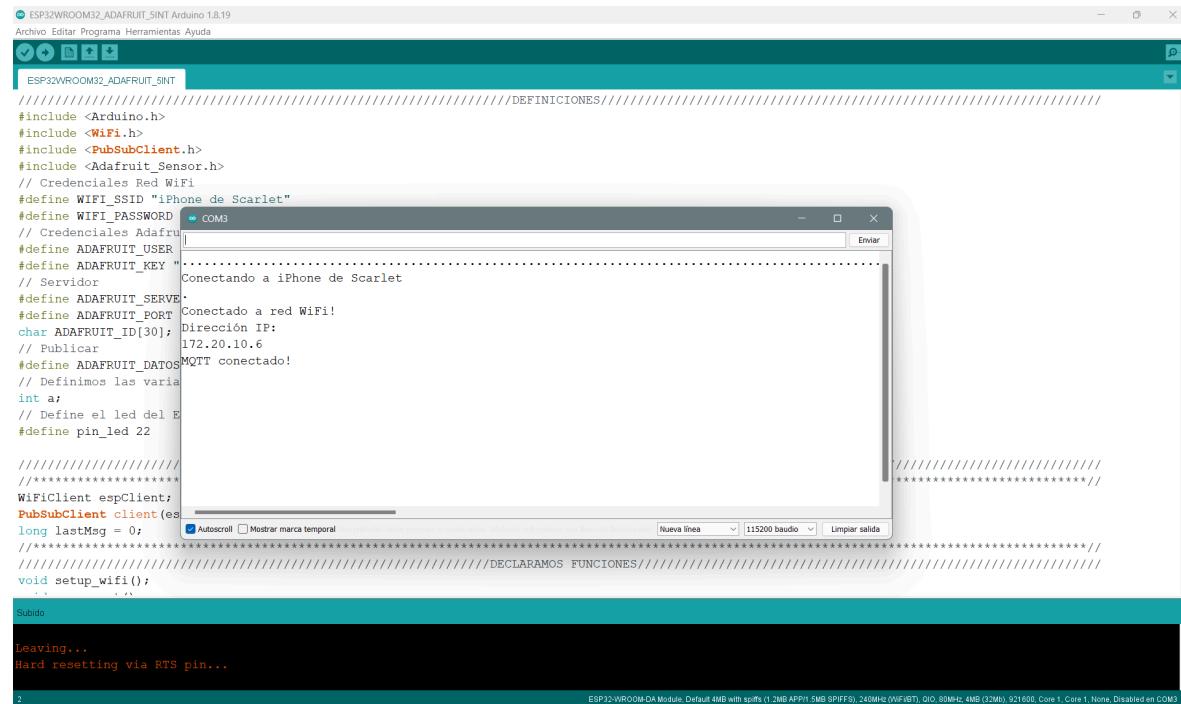
//*****DEFINICIONES*****
/////////FUNCION DE RECONECTADO////
void reconnect()
{
    while (!client.connected())
    {
        if(client.connect(ADAFRUIT_ID, ADAFRUIT_USER, ADAFRUIT_KEY) )
        {
            Serial.println("MQTT conectado!");

        } else {
            Serial.print("falló :( con error -> ");
            Serial.print(client.state());
            Serial.println(" Intentamos de nuevo en 5 segundos");
            delay(5000);
        }
    }
}
/////////FUNCION DE RECONECTADO////

```

Esta función es la encargada de conectar el servidor en caso de que este no se conecte en un cierto periodo de tiempo.

Monitor Serial



Hay que tener en cuenta que el monitor serial, será la única vez en la que podremos abrirlo durante el proyecto, ya que al momento de conectar Matlab con el serial, este no puede estar ocupado por alguien más, en este caso solo lo abrimos para verificar que se realizó correctamente la conexión por medio de MQTT al servidor.

Configuración

Herramientas	Ayuda
Auto Formato	Ctrl+T
Archivo de programa.	
Reparar codificación & Recargar.	
Administrar Bibliotecas...	Ctrl+Mayús+I
Monitor Serie	Ctrl+Mayús+M
Serial Plotter	Ctrl+Mayús+L
ESP32 Sketch Data Upload	
WiFi101 / WiFiINA Firmware Updater	
Placa: "ESP32-WROOM-DA Module"	>
Upload Speed: "921600"	>
CPU Frequency: "240MHz (WiFi/BT)"	>
Flash Frequency: "80MHz"	>
Flash Mode: "QIO"	>
Flash Size: "4MB (32Mb)"	>
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"	>
Core Debug Level: "Ninguno"	>
Arduino Runs On: "Core 1"	>
Events Run On: "Core 1"	>
Erase All Flash Before Sketch Upload: "Disabled"	>
Puerto: "COM3"	>
Obtén información de la placa	
Programador	>
Quemar Bootloader	

Esta es la configuración que debe de tener al momento de cargar el archivo al ESP32

AdaFruit

Llave de seguridad

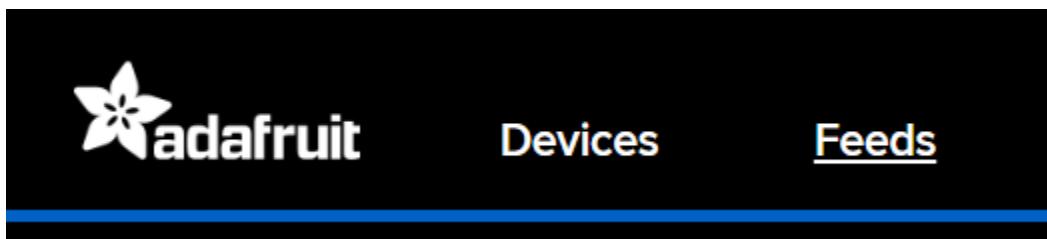
The screenshot shows the Adafruit IO Overview page. At the top, there's a navigation bar with links for Devices, Feeds, Dashboards, Actions, and Power-Ups, along with a 'New Device' button. Below the navigation is a breadcrumb trail: Atorony23 / Overview. To the right of the breadcrumb is a 'Help' link. The main content area has tabs for Overview, Privacy & Sharing, My Plan, My Data, and Activity. A prominent green banner at the top says: 'You are currently using a Adafruit IO Basic plan. For just \$10/month, upgrade to AIO+ to unlock unlimited devices, groups, feeds, dashboards, and more! Learn about the other features and benefits of upgrading your account here.' Below the banner, there are sections for Account Status (Devices: 0 of 2, Groups: 1 of 5, Feeds: 3 of 10, Dashboards: 1 of 5, Data Rate: 0 of 30), My Dashboards (Dashboard Name: MioLol), My Feeds (Feed Name: activacion, Last Value: false), and Live Errors (No errors since page load). There's also a section for Connections (No connections).

Una vez creada la cuenta en io.adafruit.com tendremos el siguiente tablero.

The screenshot shows the 'YOUR ADAFRUIT IO KEY' page. It contains a warning: 'Your Adafruit IO Key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your Adafruit IO Key can view all of your data, create new feeds for your account, and manipulate your active feeds.' To the right of the text is a QR code. Below the warning, there's a note: 'If you need to regenerate a new Adafruit IO Key, all of your existing programs and scripts will need to be manually changed to the new key.' At the bottom, there are fields for 'Username' (Atorony23) and 'Active Key' (aio_ followed by a redacted string). A 'REGENERATE KEY' button is located to the right of the key field.

En el simbolo de la llave con fondo amarillo, ahí encontraremos nuestras credenciales necesarias para poder establecer la conexión con AdaFruit con otros dispositivo.

Feeds



Procedemos irnos a la sección de Feeds, ya que necesitamos primero establecer una conexión que será la que estará mandando los datos (Tener en cuenta que si queremos mandar mas datos podremos mandarlos por el mismo feed o tendríamos que hacer otros específicos)

Atorony23 / Feeds

[+ New Feed](#) [+ New Group](#)

Default

Procedemos a dar click en New Feed y le damos un nombre y una descripción al feed que utilizaremos.

Atorony23 / Feeds

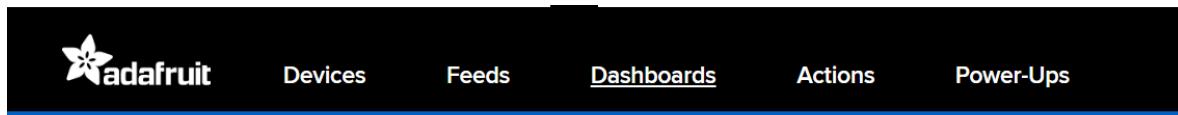
[+ New Feed](#) [+ New Group](#)

Default

Feed Name	Key
<input type="checkbox"/> activacion	activacion
<input type="checkbox"/> datos	datos

Aquí nos aparecerán los feeds ya creados (En nuestro caso, utilizaremos el feed de “datos”)

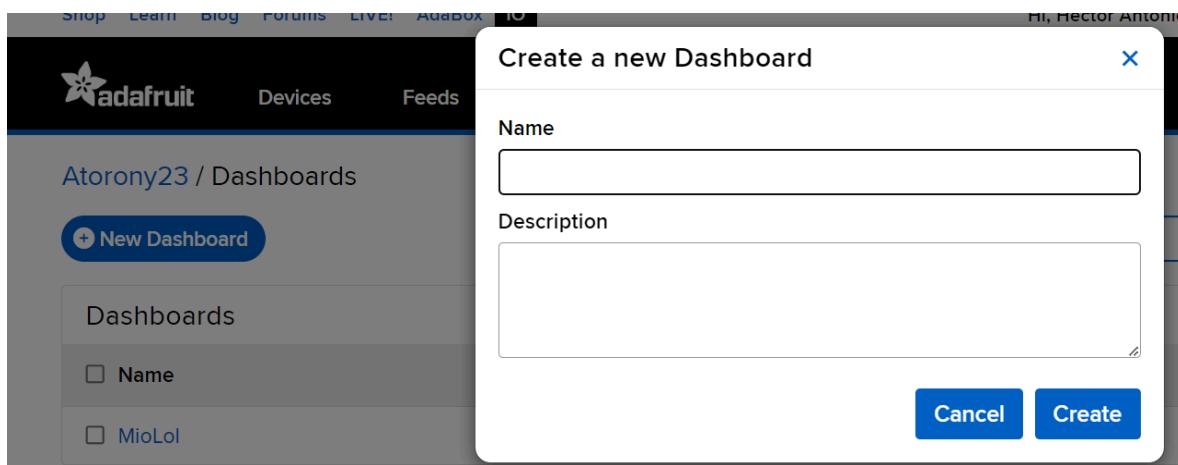
Dashboards



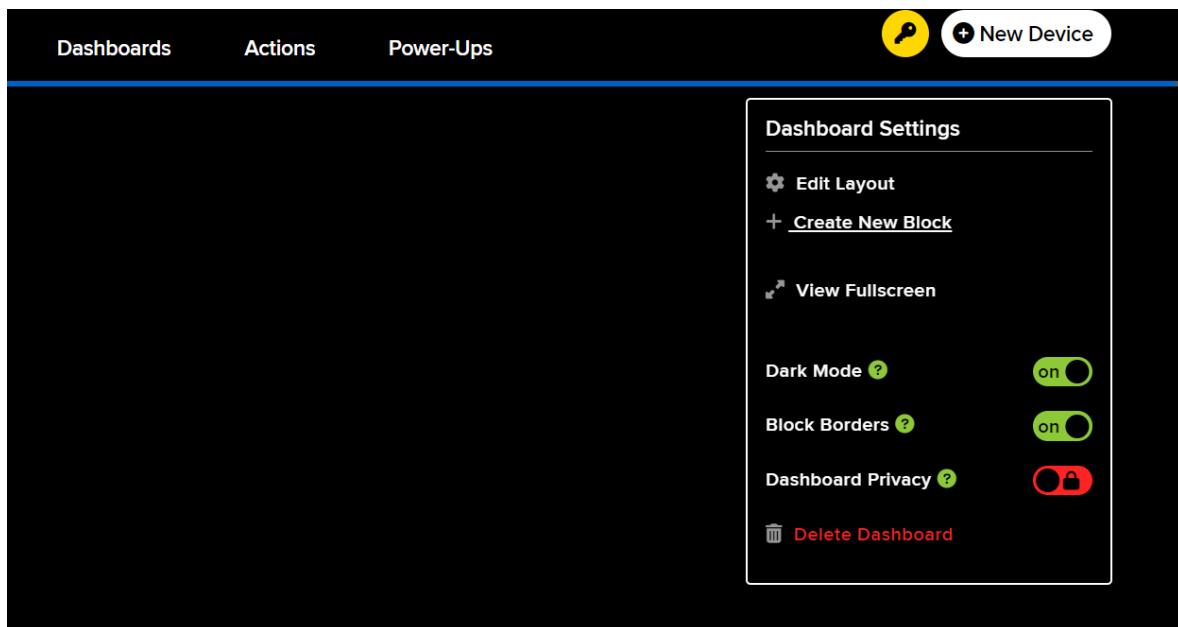
Atorony23 / Overview

[Overview](#) [Privacy & Sharing](#) [My Plan](#) [My Data](#) [Activity](#)

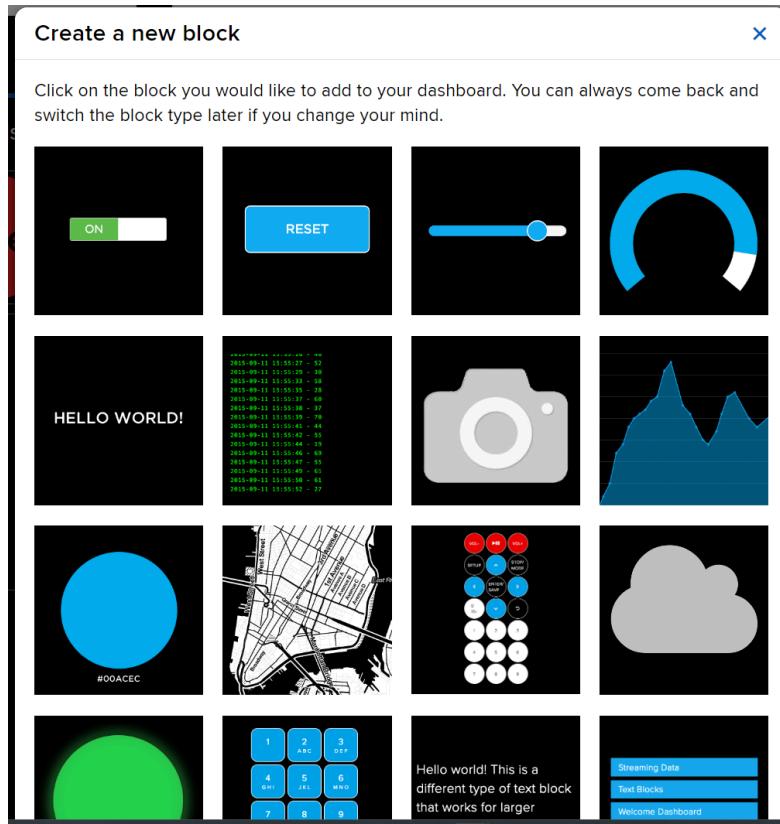
Después procederemos a irnos a la sección de Dashboards



Procedemos a crear uno nuevo y lo damos en crear.



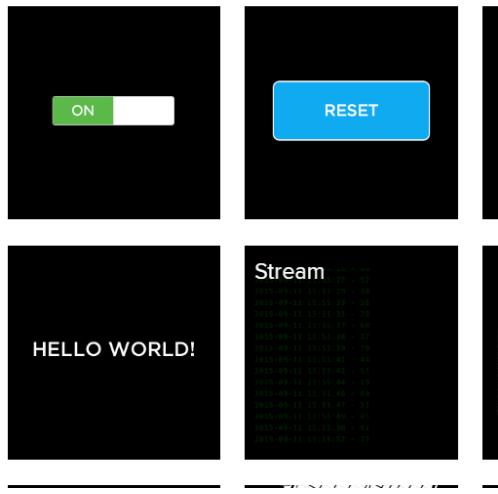
Nos abrirá una ventana negra como la que vemos, aquí es donde podremos colocar todo lo necesario para las conexiones que tendremos que establecer



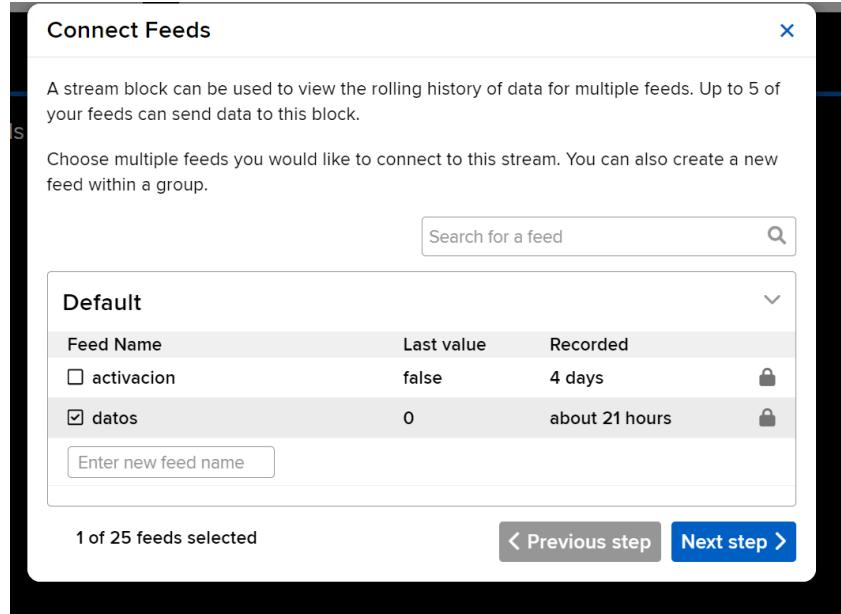
Estas son las opciones con las que contamos actualmente, cada una de estas las podremos utilizar para mandar como para recibir datos

Create a new block

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



En nuestro caso, solo para este proyecto, utilizaremos el bloque de “Stream”, el cual, recibirá los datos y los refleja para poder ver que se está recibiendo la información.



Una vez que lo seleccionamos, nos aparecerá que tipo de datos (el tipo de feed previamente creado) queremos cargar en él, en nuestro caso, seleccionaremos la opción de “datos”

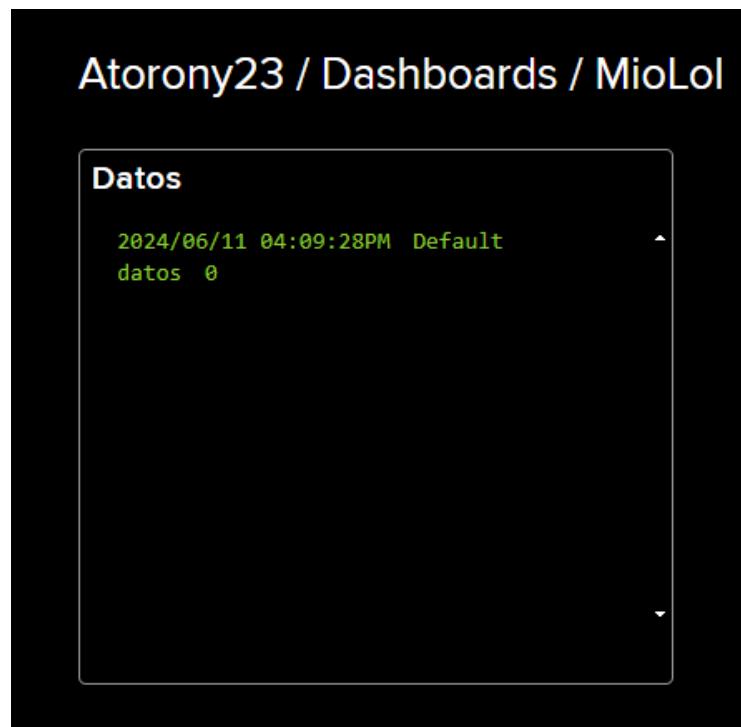
Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional)	Block Preview
Datos	Datos
Hours of History (0 for realtime)	2018/02/01 08:23AM Home Temperature 78.34 2018/02/01 08:25AM Office Temperature 78.34 2018/02/01 08:25AM Home Lamp Color #FF0028 2018/02/01 08:25AM username/errors Validation failed: Name may contain only letters, digits, underscores, spaces, or dashes 2018/02/01 08:27AM Coffee Shop Temperature 78.34 2018/02/01 08:28AM username/throttle username data rate limit reached, 58 seconds until throttle reset
Font Size	Stream A stream block can be used to view the rolling history of data for multiple feeds. Up to 5 of your feeds can send data to this block.
Small	
Colorscheme	
Green	
Show Group Name?	
Yes	
Show Feed Name?	
Yes	
Show Timestamp?	
Yes	
Show Location?	
No	
Show Errors?	
Yes	

< Previous step Create block

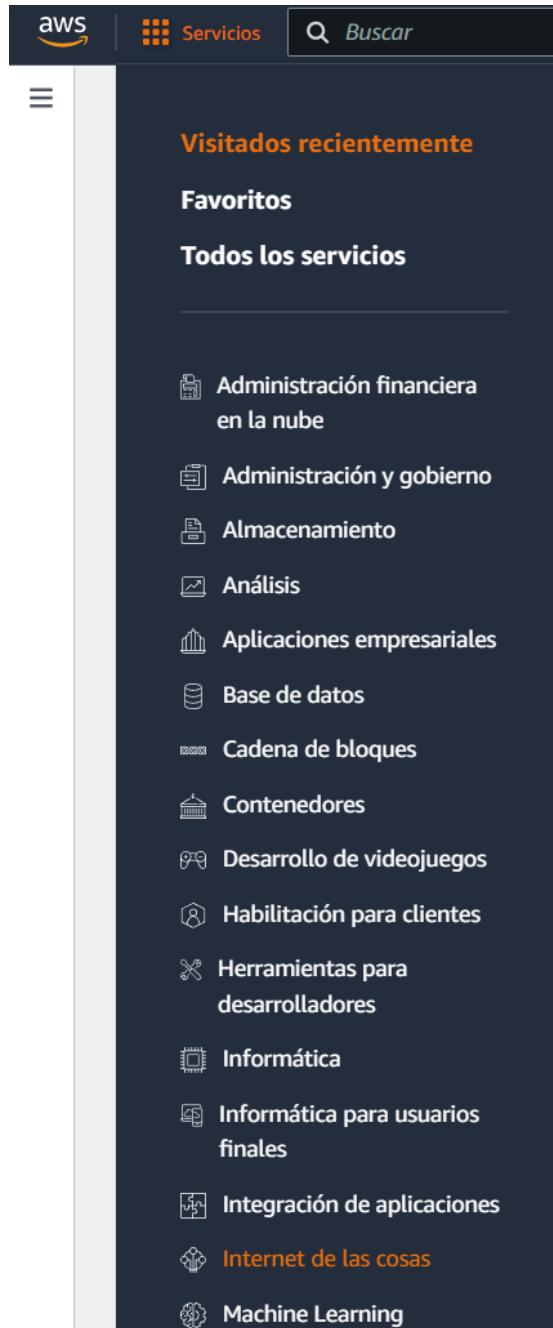
Después, nos aparecerá la opción de continuar con la configuración del bloque, en nuestro caso, lo dejaremos tal como está y damos en crear bloque.



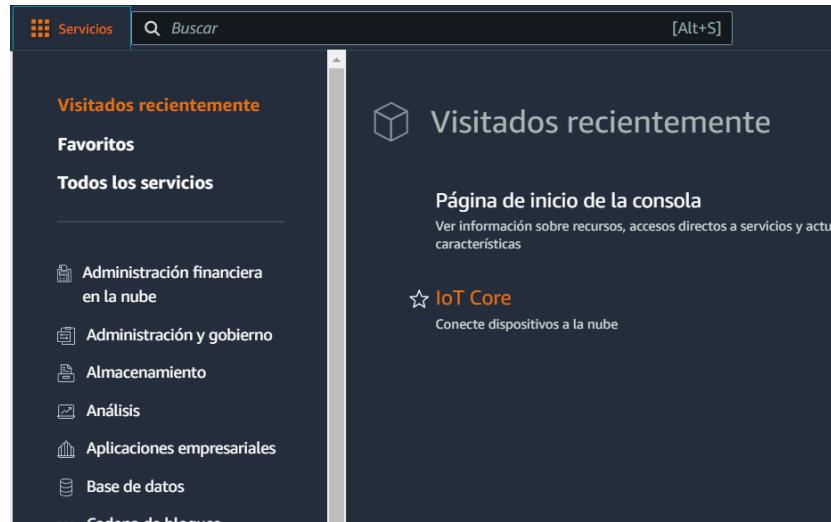
Este es el formato y tipo de bloque que creamos para poder ver y leer los datos que estamos mandando.

AWS

Servicio IoT



Nos dirigimos a la pagina de AWS, creamos una cuenta, una vez con la sesion iniciada, procedemos a irnos a [Servicios > Internet de las cosas]



Después nos dirigimos al link de IoT Core e ingresamos.

Probar

Cliente de prueba de MQTT

- Administrador
- ▼ Todos los dispositivos
- Objetos**
- Grupos de objetos
- Tipos de objetos

Tendremos que dirigirnos a la sección de objetos, ya que necesitamos crear un nuevo objeto que será con el que trabajaremos.

A screenshot of the AWS IoT Objects management page. The top navigation shows 'AWS IoT > Administrar > Objetos'. Below is a header with 'Objetos (2) Información' and buttons for 'Crear objetos' (highlighted in orange), 'Búsqueda avanzada', 'Ejecutar agregaciones', 'Editar', and 'Eliminar'. A note states: 'Un objeto de IoT es una representación y un registro del dispositivo físico en la nube. El dispositivo físico necesita un registro de objeto para poder trabajar con AWS IoT.' A search bar at the bottom says 'Filtre los objetos por: nombre, tipo, grupo, facturación o atributo que se puede buscar.' Navigation icons for pages 1 and 2 are also visible.

Crear objetos Información

Un recurso de objeto es una representación digital de un dispositivo físico o una entidad lógica en AWS IoT. Su dispositivo o entidad necesita un recurso de objeto en el registro para utilizar características de AWS IoT, como las sombras de dispositivos, los eventos, los trabajos y las características de administración de dispositivos.

Número de objetos que crear

Crear un único objeto

Cree un recurso de objeto para registrar un dispositivo. Aprovisione el certificado y la política necesarios para permitir que el dispositivo se conecte a AWS IoT.

Crear varios objetos

Cree una tarea que cree varios recursos de objetos para registrar dispositivos y aprovisionar los recursos que necesitan dichos dispositivos para conectarse a AWS IoT.

[Cancelar](#)

[Siguiente](#)

Creamos un único objeto.

Especificar propiedades de objeto Información

Un recurso de objeto es una representación digital de un dispositivo físico o una entidad lógica en AWS IoT. Su dispositivo o entidad necesita un recurso de objeto en el registro para utilizar características de AWS IoT, como las sombras de dispositivos, los eventos, los trabajos y las características de administración de dispositivos.

Propiedades del objeto

Nombre del objeto

Comunicacion

Ingrese un nombre único que contenga únicamente letras, números, guiones, dos puntos o guiones bajos. Un nombre de objeto no puede contener espacios.

Configuraciones adicionales

Puede utilizar estas configuraciones para agregar detalles que le ayuden a organizar, administrar y buscar sus objetos.

- ▶ **Tipo de objeto:** *opcional*
- ▶ **Atributos de objeto que permiten búsquedas :** *opcional*
- ▶ **Grupos de objetos:** *opcional*
- ▶ **Grupo de facturación :** *opcional*
- ▶ **Packages and versions - optional**

Sombra de dispositivo

Las sombras de dispositivos permiten que los dispositivos conectados sincronicen sus estados con AWS. También es posible obtener, actualizar o eliminar la información de estado de la sombra de este objeto mediante temas MQTT o HTTP.

Sin sombra

Sombra con nombre

Cree varias sombras con nombres diferentes para administrar el acceso a las propiedades y agrupar lógicamente las propiedades del dispositivo.

Sombra sin nombre (clásica)

Un objeto solo puede tener una sombra sin nombre.

[Cancelar](#)

[Siguiente](#)

Lo abrimos como sin sombra y le damos un nombre al objeto.

Configurar el certificado del dispositivo - *opcional* Información

Un dispositivo requiere un certificado para establecer conexión con AWS IoT. Puede elegir cómo registrar un certificado para el dispositivo ahora, o puede crear y registrar un certificado para el dispositivo posteriormente. El dispositivo no se podrá conectar a AWS IoT hasta que tenga un certificado activo con una política adecuada.

Certificado de dispositivo

Generar automáticamente un certificado nuevo (recomendado)
Genera un certificado, una clave pública y una clave privada con la entidad de certificación de AWS IoT.

Usar mi certificado
Utilice un certificado firmado por su propia entidad de certificación.

Cargar CSR
Registre la entidad de certificación y use sus propios certificados en uno o varios dispositivos.

Saltar la creación de un certificado en este momento
Puede crear un certificado para este objeto y asociar una política al certificado más adelante.

[Cancelar](#) [Anterior](#) [Siguiente](#)

Le damos que nos genere automáticamente el certificado

Asociar políticas al certificado - *opcional* Información

Las políticas de AWS IoT conceden o deniegan el acceso a los recursos de AWS IoT. Al asociar políticas al certificado de dispositivo, se aplica este acceso al dispositivo.

Políticas (1) [C](#) [Crear política](#)

Seleccione hasta 10 políticas para asociarlas a este certificado.

[Filtrar políticas](#) [<](#) [1](#) [>](#) [@](#)

<input type="checkbox"/>	Nombre
<input type="checkbox"/>	políticaNodeReddit

[Cancelar](#) [Anterior](#) [Crear un objeto](#)

Después debemos de crear una nueva política para poder asociar los objetos creados. Le damos click al botón de crear política.

Crear política Información

Las políticas de AWS IoT Core permiten administrar el acceso a las operaciones del plano de datos de AWS IoT Core.

Propiedades de la política

AWS IoT Core admite políticas con nombre para que muchas identidades puedan hacer referencia al mismo documento de políticas.

Nombre de la política

politicaNodeReddd

Un nombre de política es una cadena alfanumérica que también puede contener caracteres de punto (.), coma (,), guion (-), guion bajo (_), signo más (+), signo igual (=) y arroba (@), pero sin espacios.

▶ Etiquetas: *opcional***Instrucciones de política**

Ejemplos de políticas

Documento de política Información

Una política de AWS IoT contiene una o varias instrucciones de política. Cada instrucción de política contiene acciones, recursos y un efecto que concede o deniega las acciones por parte de los recursos.

Creador**JSON****Efecto de la política**

Permitir

Acción de la política

*

Recurso de la política

|

Eliminar

Agregar nueva instrucción

Cancelar

Crear

Una vez aquí, le damos el nombre que queramos a la política y en la sección de abajo, dejamos la primera política en permitir y las otras dos con un * [asterisco]

Asociar políticas al certificado - *opcional* Información

Las políticas de AWS IoT conceden o deniegan el acceso a los recursos de AWS IoT. Al asociar políticas al certificado de dispositivo, se aplica este acceso al dispositivo.

Políticas (1/2)

G

Crear política

Seleccione hasta 10 políticas para asociarlas a este certificado.

Filtrar políticas

< 1 >

@

Nombre

 politicaNodeReddd politicaNodeRedd

Cancelar

Anterior

Crear un objeto

Una vez creada la política, regresamos a asociar las políticas con los objetos y creamos el objeto con la política creada.

Descargar certificados y claves

Descargue archivos de claves y certificados para instalarlos en el dispositivo y que este pueda conectarse a AWS.

Certificado de dispositivo

Puede activar el certificado ahora o posteriormente. El certificado debe estar activo para que un dispositivo se conecte a AWS IoT.

Certificado de dispositivo
ce15a41d511...te.pem.crt

[Desactivar certificado](#) [Descargar](#)

Archivos de claves

Los archivos de claves son exclusivos de este certificado y no se pueden descargar después de salir de esta página. Descárguelos ahora y guárdelos en un lugar seguro.

⚠ Esta es la única vez que puede descargar los archivos de claves correspondientes a este certificado.

Archivo de clave pública
ce15a41d5115e053740f984...e51d51b-public.pem.key

[Descargar](#)

Archivo de clave privada
ce15a41d5115e053740f984...51d51b-private.pem.key

[Descargar](#)

Certificados de entidad de certificación raíz

Descargue el archivo de certificado de entidad de certificación raíz correspondiente al tipo de punto de enlace de datos y conjunto de cifrado que está utilizando. También puede descargar los certificados de entidad de certificación raíz más adelante.

Punto de enlace de servicios de confianza de Amazon
Clave RSA de 2048 bits: Amazon Root CA 1

[Descargar](#)

Punto de enlace de servicios de confianza de Amazon
Clave ECC de 256 bits: Amazon Root CA 3

[Descargar](#)

Si no ve el certificado de entidad de certificación raíz que necesita aquí, AWS IoT admite certificados de entidad de certificación raíz adicionales. Estos certificados de entidad de certificación raíz están disponibles en nuestras guías para desarrolladores. [Más información](#)

[Listo](#)

MUY IMPORTANTE!! Ocuparemos los siguientes archivos:

- 1.- Archivo de Clave Pública
- 2.- Archivo de Clave Privada
- 3.- Punto de enlace de servicios de confianza de Amazon

Debemos de descargar estos archivos ya que si no se descargan, no podremos volver a generarlos, una vez descargados, debemos de guardarlos bien para que estos no se nos pierdan.

AWS IoT

X

Monitorear

Conectarse

Conectar un dispositivo

► Conectar muchos dispositivos

Probar

Cliente de prueba de MQTT

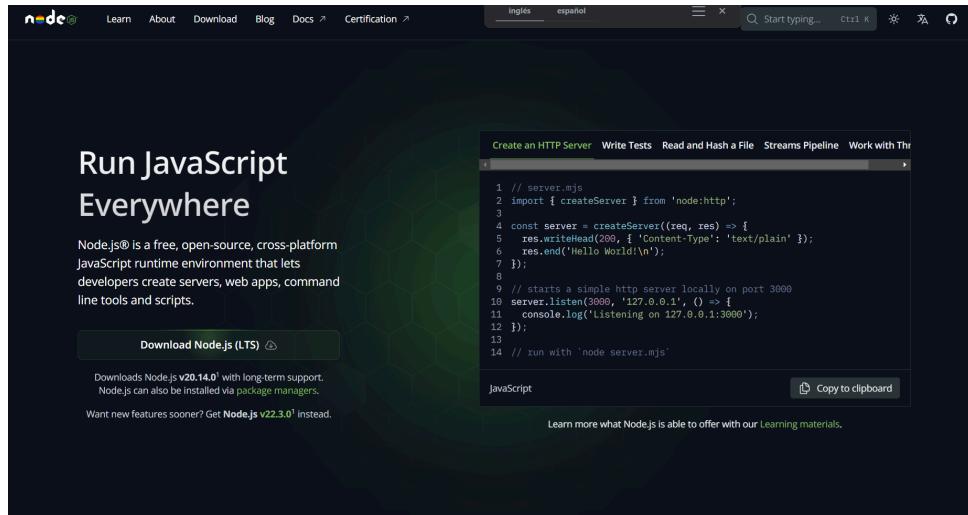
Una vez creado el objeto y con los certificados listos, podemos irnos a cliente de prueba de MQTT

The screenshot shows the AWS IoT Test Client interface. At the top, there are two tabs: "Suscribirse a un tema" (Subscribe to a topic) and "Publicar en un tema" (Publish to a topic). Below these tabs, there is a "Filtro de temas" (Topic filter) input field containing the text "Atorony23/datos". Underneath the input field, there is a "Configuración adicional" (Additional configuration) section with a "Suscribir" (Subscribe) button. In the main content area, there is a table with two columns: "Suscripciones" (Subscriptions) and "Atorony23/datos". The "Atorony23/datos" row has several buttons: "Poner en pausa" (Pause), "Borrar" (Delete), "Exportar" (Export), and "Editar" (Edit). Below the table, there is a "Carga del mensaje" (Message payload) input field containing the JSON message: { "message": "Hello from AWS IoT console" }. There is also a "Configuración adicional" (Additional configuration) section with a "Publicar" (Publish) button.

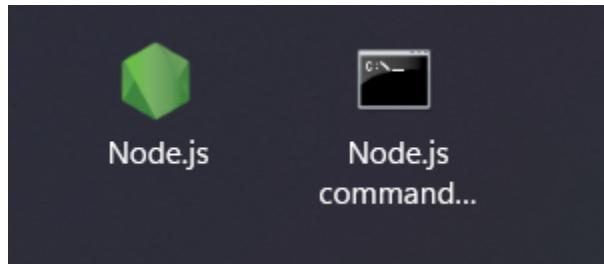
Una vez aquí, podremos suscribirnos a la red, colocando [Nombre/ “el tema”]

Node-Red

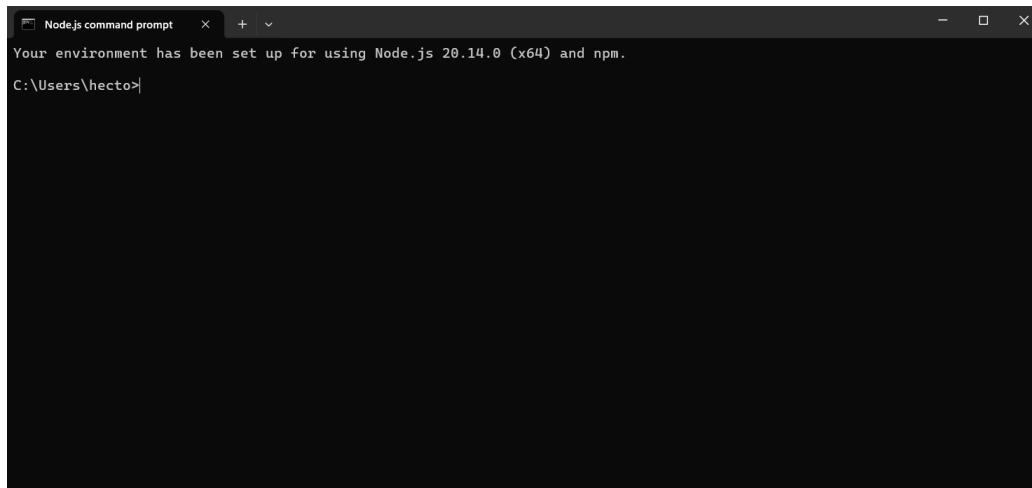
Descarga Node Red



Nos vamos a la página de Node.js y descargamos el software



Nos instalara estas dos aplicaciones, la que mas utilizaremos será “Node.js command prompt”



Procedemos a abrirlo y escribir el siguiente comando:

Unset

```
npm install -g --unsafe-perm node-red
```

Escribimos esta linea de código en el command prompt [Esto solo se hará una vez]

```
Your environment has been set up for using Node.js 20.14.0 (x64) and npm.

C:\Users\hecto>node-red
12 Jun 13:04:37 - [info]

Welcome to Node-RED
=====

12 Jun 13:04:37 - [info] Node-RED version: v3.1.9
12 Jun 13:04:37 - [info] Node.js version: v20.14.0
12 Jun 13:04:37 - [info] Windows_NT 10.0.22631 x64 LE
12 Jun 13:04:37 - [info] Loading palette nodes
12 Jun 13:04:38 - [info] Dashboard version 1.0.2 started at /ui
12 Jun 13:04:38 - [warn] -----
12 Jun 13:04:38 - [warn] [tuya-led-node/tuya-led] ReferenceError: TuyaLedNode is not defined (line:139)
12 Jun 13:04:38 - [warn] -----
12 Jun 13:04:38 - [info] Settings file : C:\Users\hecto\.node-red\settings.json
12 Jun 13:04:38 - [info] Context store : 'default' [module=memory]
12 Jun 13:04:38 - [info] User directory : \Users\hecto\.node-red
12 Jun 13:04:38 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Jun 13:04:38 - [info] Flows file : \Users\hecto\.node-red\flows.json
12 Jun 13:04:38 - [info] Server now running at http://127.0.0.1:1880/
12 Jun 13:04:38 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.

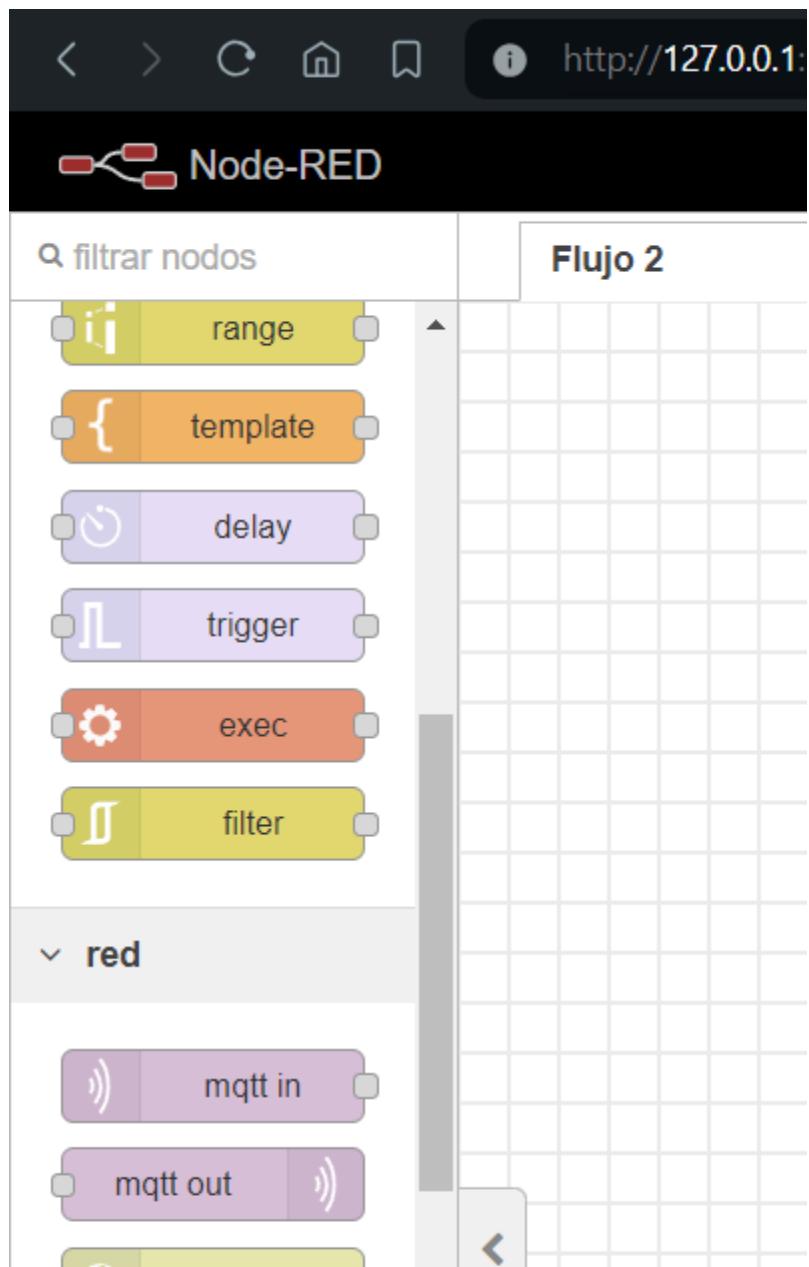
-----
12 Jun 13:04:38 - [info] Starting flows
12 Jun 13:04:39 - [info] Started flows
```

Una vez instalado el node-red, procedemos a abrirlo escribiendo en el mismo command prompt, escribiendo “Node-red” y dando enter, nos abrirá todo las líneas de código que se ven arriba, una vez corriendo y que aparezca “Started flows” podemos ingresar a la red.

Servidor Node-Red



Nos vamos a un navegador y escribimos la dirección IP: “127.0.0.1: 1880” e ingresamos.



Una vez dentro, podremos ver los distintos bloques que nos aparecen, en este caso usaremos el mqtt in y el mqtt out

Editar nodo mqtt in

Eliminar Cancelar Hecho

Propiedades

Servidor: io.adafruit.com:1883

Acción: Suscríbete a un solo tema

Tema: Atorony23/f/datos

CdS: 2

Salida: auto-detectar (objeto JSON, texto o buffer)

Nombre: SalidaAdaFruita_

Para establecer la conexión de entrada, abrimos el bloque de mqtt in, en este, colocamos los siguientes datos:

ATENCIÓN!! En la sección de Tema, va lo que es nuestro tema a recopilar datos que son los de AdaFruit, importante colocar la /f/ en medio, ya que esta hace referencia a los Feeds, entonces nos queda la línea: [NombreUsuario/f/tema]

Editar nodo mqtt in > **Editar nodo mqtt-broker**

Eliminar Cancelar **Actualizar**

Propiedades

Nombre

Conexión Seguridad Mensajes

Servidor Puerto
 Conectar automáticamente
 Utilizar TLS

Protocolo ▼

ID Cliente

Mantener activo

Sesión Usar sesión limpia

Después le damos click en el lápiz del servidor, en este debemos de establecer la siguiente configuración:

Editar nodo mqtt in > **Editar nodo mqtt-broker**

Eliminar Cancelar Actualizar

Propiedades

Nombre:

Conexión **Seguridad** Mensajes

Usuario: Atorony23

Contraseña:

Y por último para el primer bloque, establecemos la seguridad, que es únicamente tu nombre de usuario de Adafruit y tu contraseña.

Editar nodo mqtt out

Eliminar Cancelar Hecho

Propiedades

Servidor: awsCifrado

Tema: Atorony23/datos

CdS: 0 Retener

Nombre: AdaFruitAAWS

Consejo: Deja el tema, CdS o manténgalo en blanco si quieres configurarlos a través de las propiedades del mensaje.

Dentro del segundo bloque es la entrada a AWS, colocamos los siguientes datos.

ATENCIÓN!! En el tema, será el tema al que nos suscribimos anteriormente en AWS y es en donde queremos que se escriban los datos.

Editar nodo mqtt out > **Editar nodo mqtt-broker**

Eliminar Cancelar **Actualizar**

Propiedades

Nombre: awsCifrado

Conexión Seguridad Mensajes

Servidor: a5vlqeu49x343-ats.iot.us-east-2.amazonaws.com Puerto: 8883

Conectar automáticamente

Utilizar TLS Configuración TLS

Protocolo: MQTT V3.1.1

ID Cliente: Dejar en blanco para auto generado

Mantener activo: 60

Sesión: Usar sesión limpia

Ed

Ingresando al lapizito de servidor, colocamos los siguientes datos:

ATENCIÓN!! El servidor podemos ver en la siguiente imagen de donde se saca.

The screenshot shows the AWS IoT Configuration page. On the left, there's a sidebar with navigation links like 'Conectar un dispositivo', 'Probar' (Client de prueba de MQTT), 'Administrar' (Todos los dispositivos, Dispositivos de Greengrass, Paquetes de software, Acciones remotas, Enrutamiento de mensajes, Mensajes retenidos, Seguridad, Fleet Hub), 'Software de dispositivos', 'Grupos de facturación', 'Configuración' (Destacar foco, Documentación), and a link to 'Coméntenos su opinión'. The main content area is titled 'Configuración' and contains sections for 'Punto de enlace de datos de dispositivo' (with a note about using the device data endpoint API REST), 'Punto de enlace' (with a copyable endpoint URL 'a5vlqeu49x343-ats.iot.us-east-2.amazonaws.com'), 'Seleccionar política de seguridad' (with a selected policy 'IoTSecurityPolicy_TLS13_1_2_2022_10' and a 'Comparación de políticas de seguridad' link), and 'Configuraciones de dominio' (with a note about creating domain configurations for tasks like certificate management). A table for domain configurations is shown with one row: 'No 1'.

En la página de AWS, nos vamos a la sección de configuración y la página que colocaremos será la de Punto de enlace.

Editar nodo mqtt out > **Editar nodo mqtt-broker**

Propiedades

Nombre	awsCifrado	
Conexión	Seguridad	Mensajes
Usuario	Atorony23	
Contraseña	

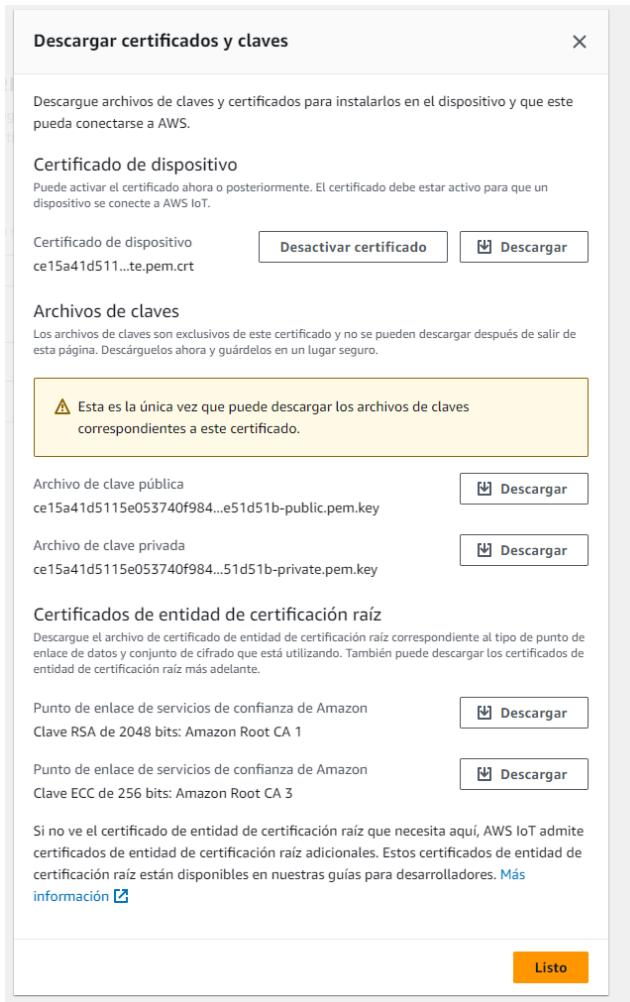
Después nuevamente en el bloque de salida a AWS, en seguridad, colocamos la cuenta y contraseña de nuestra cuenta de AWS.

Editar nodo mqtt out > Editar nodo mqtt-broker > **Editar nodo tls-config**

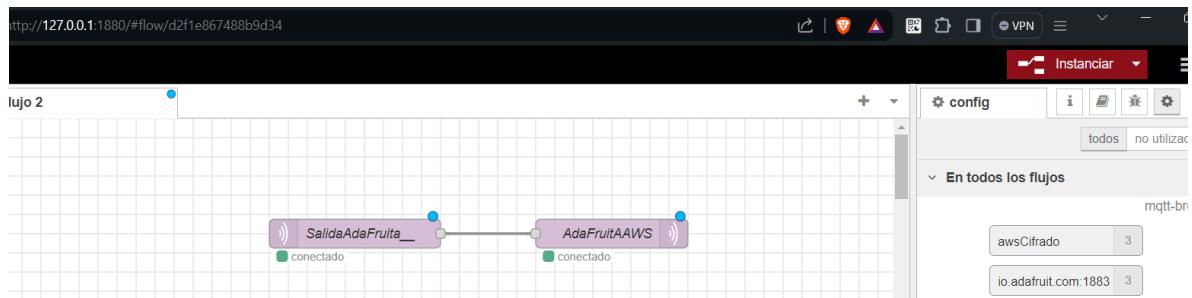
Propiedades

<input type="checkbox"/> Utilizar claves y certificados de archivos locales
<input type="file"/> Certificado <input type="button" value="Cargar"/> cbfc525a2206be9c01136fb6fc4a714893aa4efc272923ac595de15e6c986358-certificate.pem <input type="button" value="x"/>
<input type="file"/> Clave Privada <input type="button" value="Cargar"/> cbfc525a2206be9c01136fb6fc4a714893aa4efc272923ac595de15e6c986358-private.pem <input type="button" value="x"/>
Frase de contraseña <input type="text" value="frase de contraseña de clave privada (opcional)"/>
<input type="file"/> Certificado CA <input type="button" value="Cargar"/> AmazonRootCA1.pem <input type="button" value="x"/>
<input checked="" type="checkbox"/> Verificar el certificado del servidor
<input type="text"/> Nombre del servidor <input type="text" value="para uso con SNI"/>
<input type="text"/> Protocolo ALPN <input type="text" value="para uso con ALPN"/>
Nombre <input type="text" value="Nombre"/>

Regresando a conexión, en el lapizito ingresamos al siguiente enlace, cargamos los certificados de AWS que previamente descargamos desde AWS, ingresamos en certificado el Certificate, en clave privada el private, y por último, el certificado CA, sería el Amazon RootCA1



Estos son los certificados que previamente descargamos del servidor de AWS



Y por último, así nos queda en Node-red las conexiones de la misma.



Ya solo le damos click en “Instanciar” y listo, la forma de comprobar que si conectaron es ver que aparece conectado en ambos bloques.

```

node-red
+ ~
12 Jun 13:04:37 - [info]
Welcome to Node-RED
=====
12 Jun 13:04:37 - [info] Node-RED version: v3.1.9
12 Jun 13:04:37 - [info] Node.js version: v20.14.0
12 Jun 13:04:37 - [info] Windows_NT 10.0.22631 x64 LE
12 Jun 13:04:37 - [info] Loading palette nodes
12 Jun 13:04:38 - [info] Dashboard version 1.0.2 started at /ui
12 Jun 13:04:38 - [warn]
12 Jun 13:04:38 - [warn] [tuya-led-node/tuya-led] ReferenceError: TuyaLedNode is not defined (line:139)
12 Jun 13:04:38 - [warn]
12 Jun 13:04:38 - [info] Settings file : C:\Users\hecto\.node-red\settings.js
12 Jun 13:04:38 - [info] Context store : 'default' [module=memory]
12 Jun 13:04:38 - [info] User directory : \Users\hecto\.node-red
12 Jun 13:04:38 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Jun 13:04:38 - [info] Flows file : \Users\hecto\.node-red\flows.json
12 Jun 13:04:38 - [info] Server now running at http://127.0.0.1:1880/
12 Jun 13:04:38 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

12 Jun 13:04:38 - [info] Starting flows
12 Jun 13:04:39 - [info] Started flows
12 Jun 13:04:39 - [info] [mqtt-broker:f491a795268f515b] Connected to broker: mqtt://io.adafruit.com:1883
12 Jun 13:04:39 - [info] [mqtt-broker:awsCifrado] Connected to broker: mqts://a5vlqeu49x343-ats.iot.us-east-2.amazonaws.com:8883

```

Y por último solo revisamos en el command prompt que aparezca que conectó a ambos brokers conectados.

SEGUNDO EQUIPO

AWS

Recuperación de datos

The screenshot shows the AWS IoT Client de prueba de MQTT interface. The left sidebar has a navigation menu with options like 'Monitorear', 'Conectarse' (selected), 'Probar' (selected), and 'Administrar'. Under 'Probar', 'Cliente de prueba de MQTT' is selected. The main content area is titled 'Cliente de prueba de MQTT' and contains a section 'Detalles de la conexión' with a status indicator 'Conectado'. Below this are tabs for 'Suscribirse a un tema' (selected) and 'Publicar en un tema'. A 'Filtro de temas' input field contains the value 'Atorony/datos'. At the bottom of the page, there's a footer with links to 'https://us-east-2.console.aws.amazon.com/console/home?region=us-east-2', '© 2024, Amazon Web Services, Inc. o sus filiales.', 'Privacidad', 'Términos', and 'Preferencias de cookies'.

En el nuevo equipo, únicamente necesitamos abrir la cuenta de AWS, en el dispositivo en el que recuperaremos los datos enviados. Nos vamos a Cliente de prueba y suscribimos el tema que queremos ver.

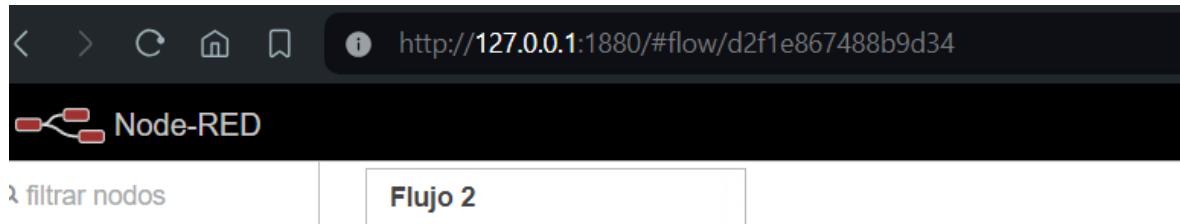
Node-Red

Conexión

```
node-red
our environment has been set up for using Node.js 19.4.0 (x64) and npm.
:C:\Users\Juan Gabino\node-red
1 Jun 13:32:08 - [info] 
-----[info] Welcome to Node-RED-----[info]
1 Jun 13:32:08 - [info] Node-RED version: v3.0.2
1 Jun 13:32:08 - [info] Node.js version: v19.4.0
1 Jun 13:32:08 - [info] Windows_NT 10.0.19045 x64 LE
1 Jun 13:32:13 - [info] Loading palette nodes
1 Jun 13:32:17 - [info] Dashboard version 3.3.1 started at /ui
1 Jun 13:32:17 - [info] Settings file : C:\Users\Juan Gabino\.node-red\settings.json
1 Jun 13:32:17 - [info] Context store : 'default' [module=memory]
1 Jun 13:32:17 - [info] User directory : \Users\Juan Gabino\.node-red
1 Jun 13:32:17 - [warn] Projects disabled : editorTheme.projects.enabled=false
1 Jun 13:32:17 - [info] Flows file : \Users\Juan Gabino\.node-red\flows.json
1 Jun 13:32:17 - [info] Server now running at http://127.0.0.1:1880/
1 Jun 13:32:17 - [warn] 

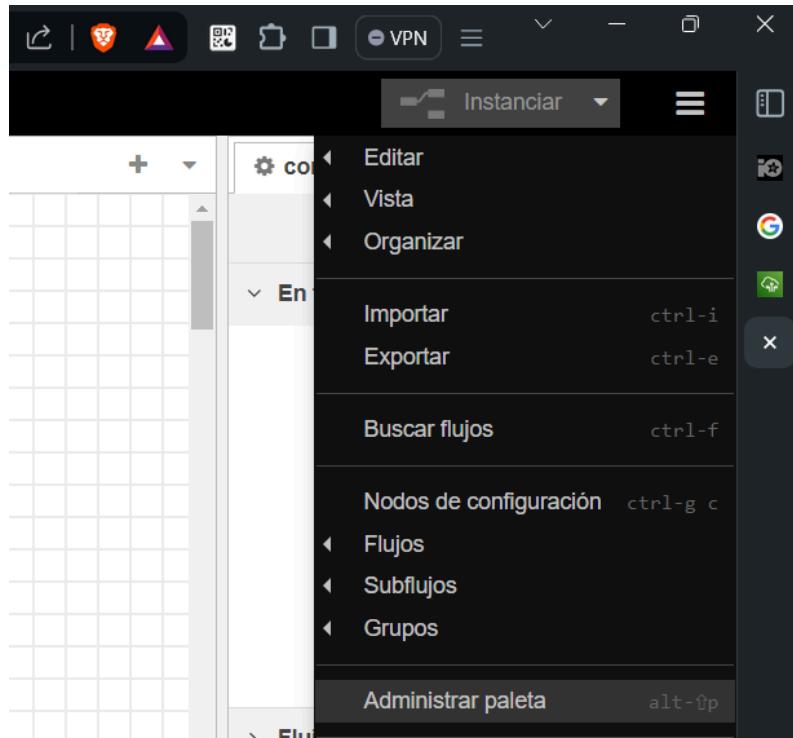
-----[info] Your flow credentials file is encrypted using a system-generated key.
[info] If the system-generated key is lost for any reason, your credentials
[info] file will not be recoverable, you will have to delete it and re-enter
[info] our credentials.
-----[info] You should set your own key using the 'credentialSecret' option in
[info] your settings file. Node-RED will then re-encrypt your credentials
[info] file using your chosen key the next time you deploy a change.
-----[info] Starting flows
1 Jun 13:32:17 - [info] [mindconnect:Agente_Empaqueadora] settings: validates timeseries validates events keep-alive rotation: every hour control topic: disabled verb
use info: enabled data lake only: false parallel requests: 1 async requests wait: 10s retries: 3
1 Jun 13:32:17 - [info] Started flows
1 Jun 13:32:18 - [info] [mqtt-broker:AWSRed] Connected to broker: mqtt://a5vlqe49x343-ats.iot.us-east-2.amazonaws.com:8883
```

Debemos de realizar la instalación de node.js en el otro dispositivo, como previamente se vio en los pasos anteriores



Ingresamos en el nuevo dispositivo a la misma red.

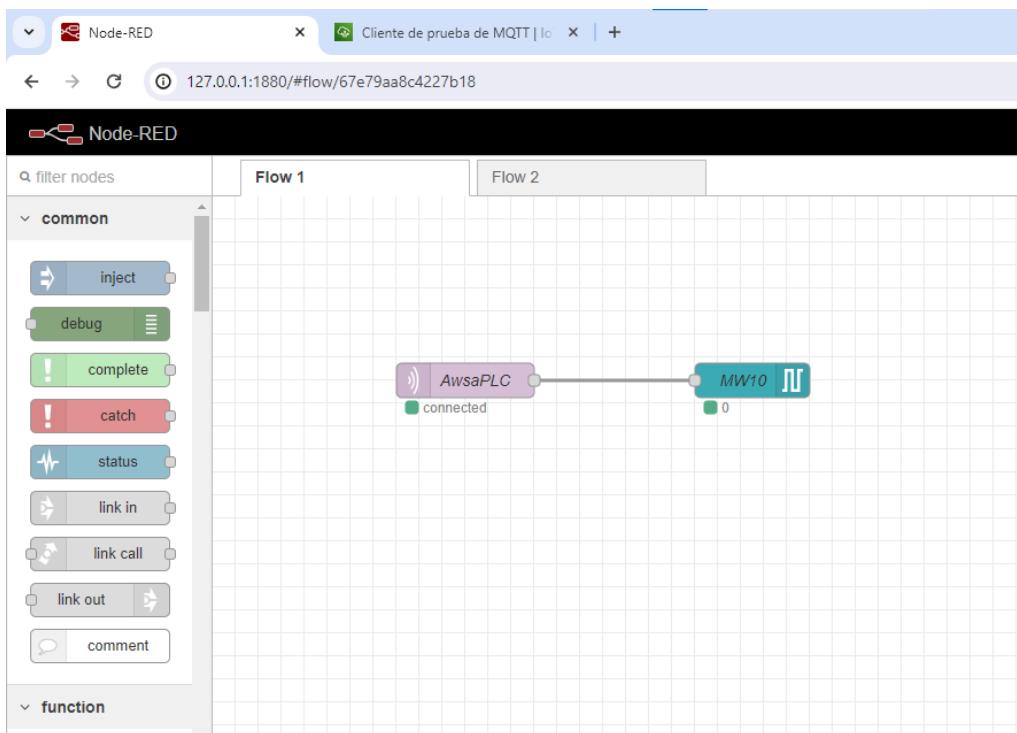
Entrada a PLC S7



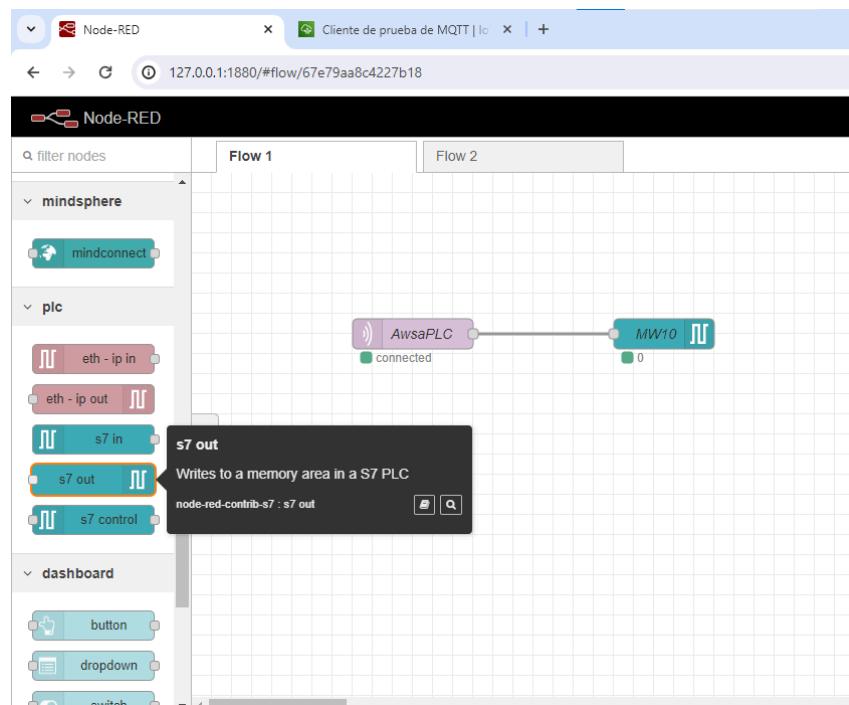
Primero para poder sacar los datos de la red de amazon ocupamos instalar a donde llegaran, que sera un dispositivo de siemens S7, entonces nos dirigimos a Paleta

A screenshot of the Node-RED 'Ajustes de usuario' (User Settings) window. The 'Paleta' tab is active. In the search bar at the top, the text 's7' is typed. Below the search bar, there are three listed packages: 1. 'ncd-red-ads7828' by 'Node-RED Community catalogue', version 1.1.5, released 5 years and 9 months ago. It has an 'instalar' (Install) button. 2. '@rakwireless/ads7830' by 'Node-RED Community catalogue', version 1.0.0, released 2 years and 1 month ago. It has an 'instalar' (Install) button. 3. 'node-red-contrib-s7' by 'Node-RED Community catalogue', version 3.1.0, released 2 years and 7 months ago. It has an 'instalado' (Installed) button. The left sidebar shows tabs for 'Vista', 'Nodos', and 'Instalar', with 'Instalar' currently selected. There are also tabs for 'Teclado' and 'Entorno'.

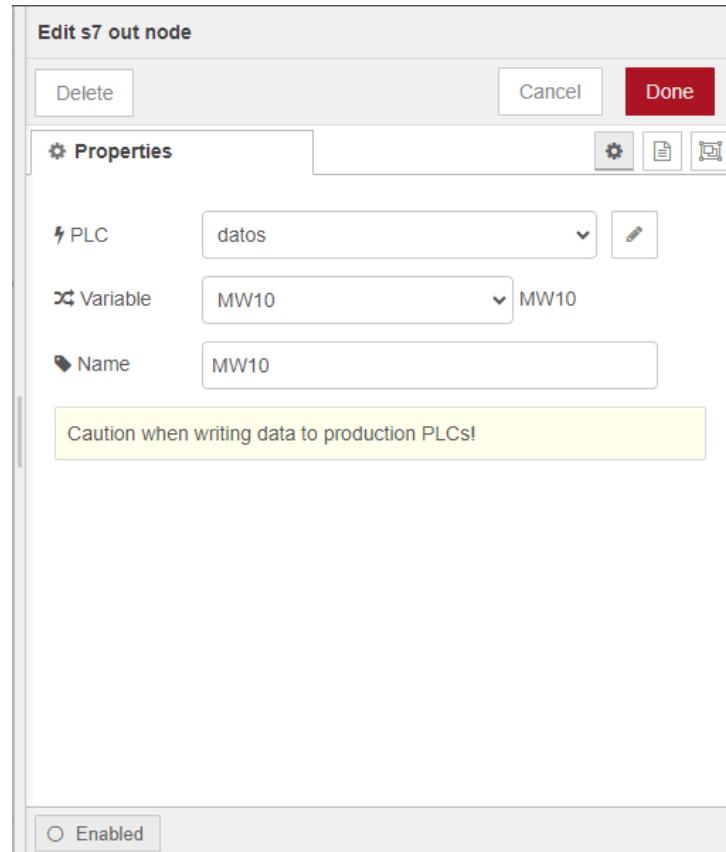
Ahí buscamos S7 e instalamos el “node-red-contrib-s7”



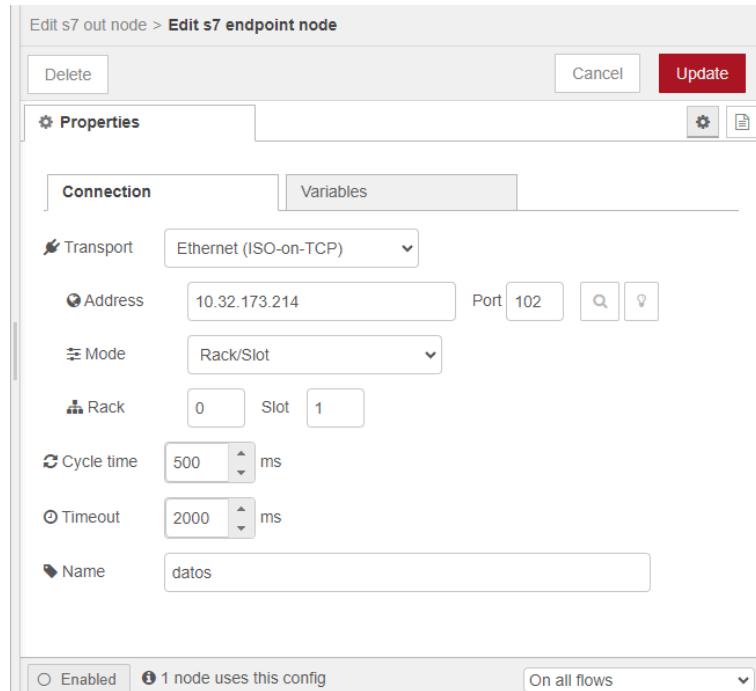
Una vez instalado, procedemos a sacar los bloques y los conectamos de la siguiente manera (En ese momento no aparecerán los bloques verdes de conectado y el dato)



Cómo escribiremos en el PLC debemos de colocar un bloque de “s7 out”

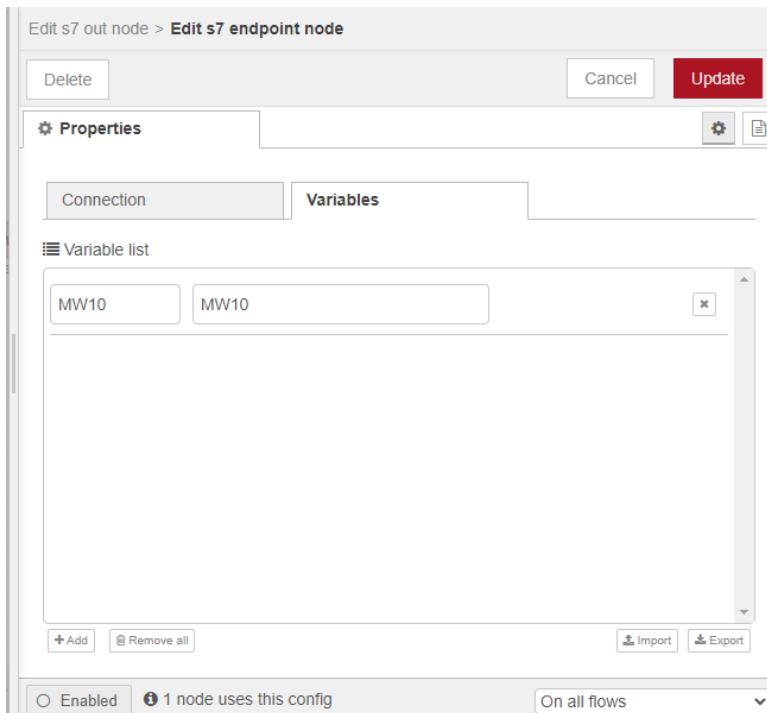


Abrimos el bloque del PLC e ingresamos, como se llama la fuente de donde viene el dato, en nuestro caso “datos”



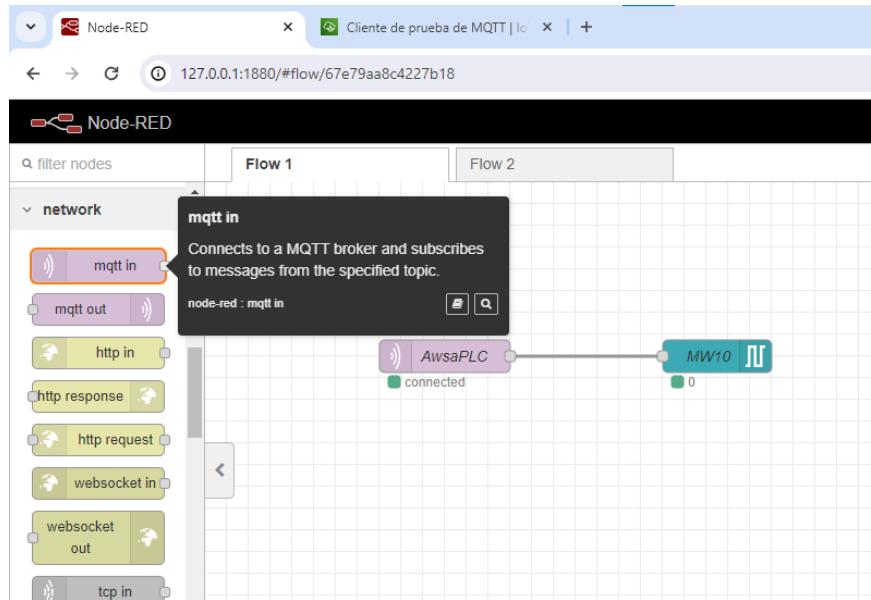
Damos click en el lapizito e ingresamos a la siguiente ventana la cual, se debe de colocar de la siguiente manera.

ATENCIÓN!! La dirección es la dirección que tiene actualmente el PLC y debemos de fijarnos bien en qué Slot aparece el procesador de nuestro PLC

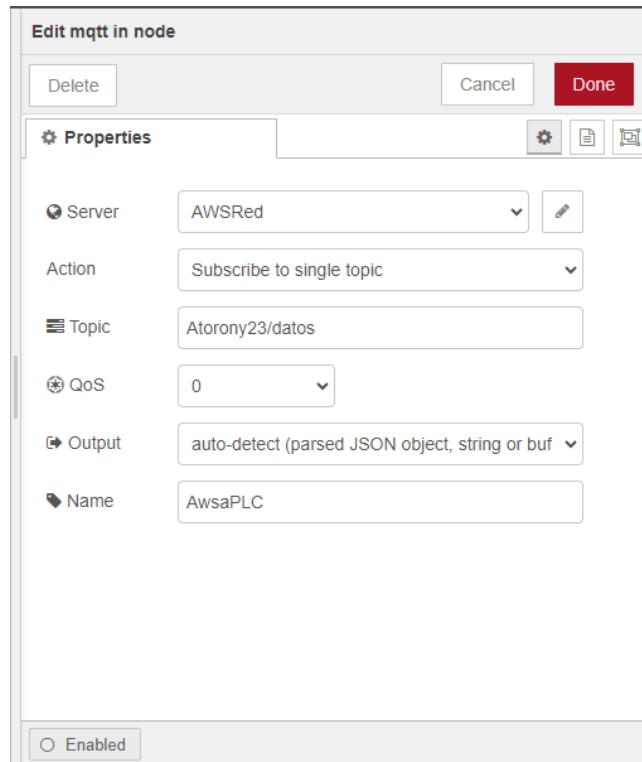


En variables escribimos el nombre de la variable en el PLC a la que se le estará enviando el dato.

Salida MQTT a PLC



En el bloque de MQTT In es el que nos permitirá recuperar los datos de AWS para poder mandarlos al PLC



En el bloque metemos la siguiente información (El tópico es el mismo que pusimos en la otra PC y en AWS al que estamos suscritos)

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel **Update**

Properties

Name AWSRed

Connection Security Messages

Server a5vlqeu49x343-ats.iot.us-east-2.amazonaws.com Port 8883

Connect automatically

Use TLS **TLS configuration**

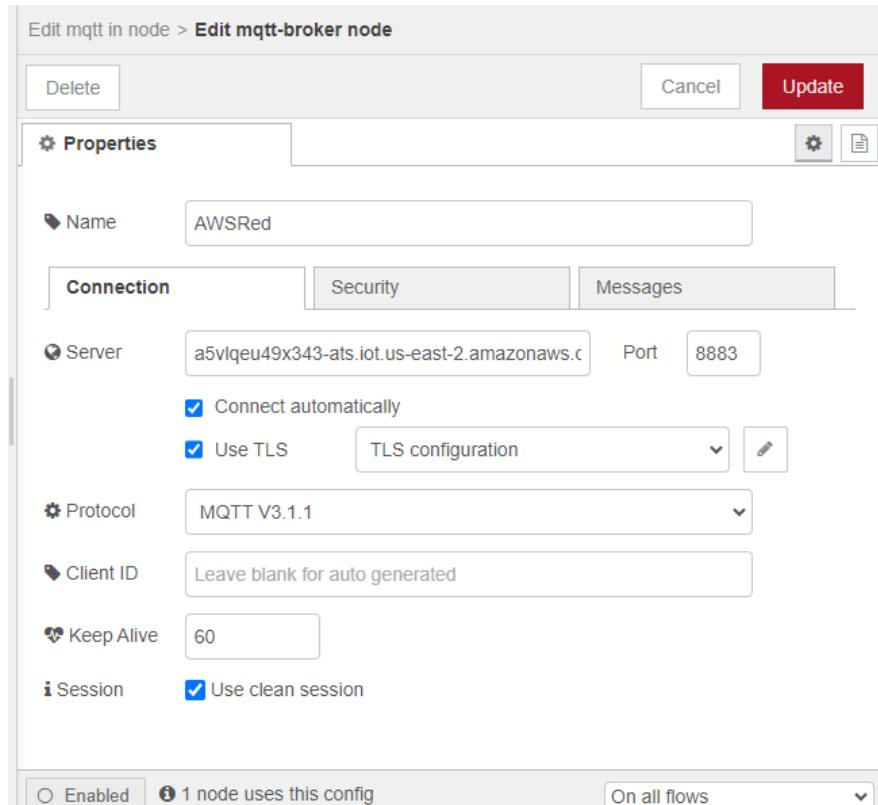
Protocol MQTT V3.1.1

Client ID Leave blank for auto generated

Keep Alive 60

Session Use clean session

Enabled **1 node uses this config** On all flows



Realizamos la misma configuración que cuando enviamos los datos a la red AWS

Edit mqtt in node > **Edit mqtt-broker node**

Delete Cancel **Update**

Properties

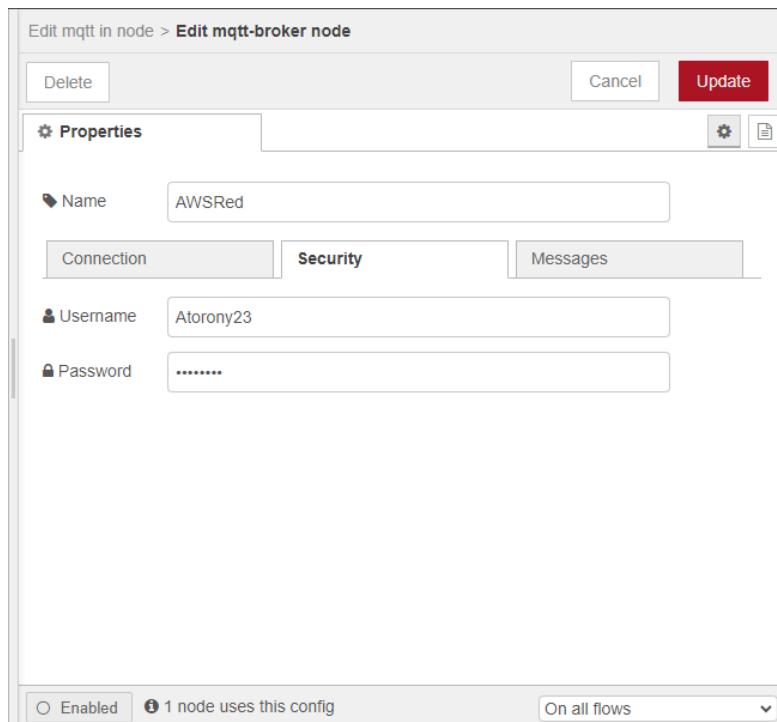
Name AWSRed

Connection **Security** Messages

Username Atorony23

Password **.....**

Enabled **1 node uses this config** On all flows



Cuenta y clave de AWS

Edit mqtt in node > Edit mqtt-broker node > **Edit tls-config node**

Properties

Use key and certificates from local files

Certificate cbfc525a2206be9c01136fb6fc4a714893aa4efc272923ac595de15e6c986358-certificate...

Private Key cbfc525a2206be9c01136fb6fc4a714893aa4efc272923ac595de15e6c986358-private.p...

Passphrase

CA Certificate AmazonRootCA1.pem

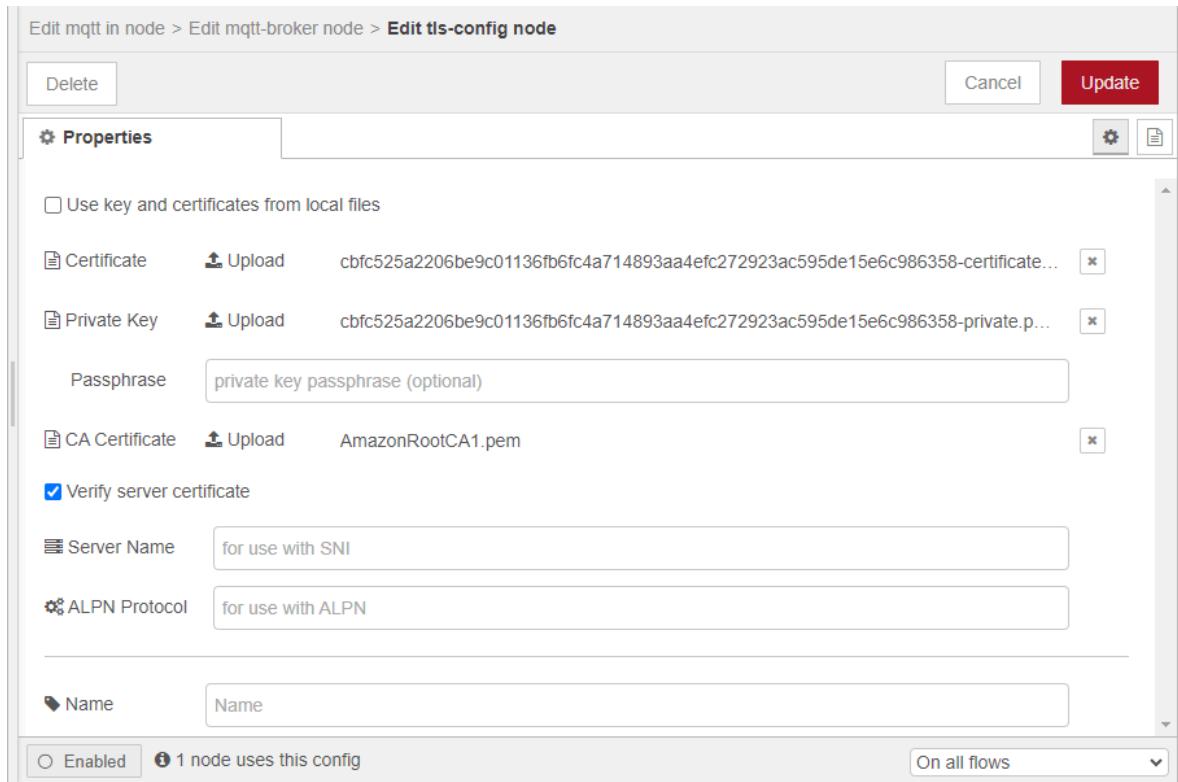
Verify server certificate

Server Name for use with SNI

ALPN Protocol for use with ALPN

Name

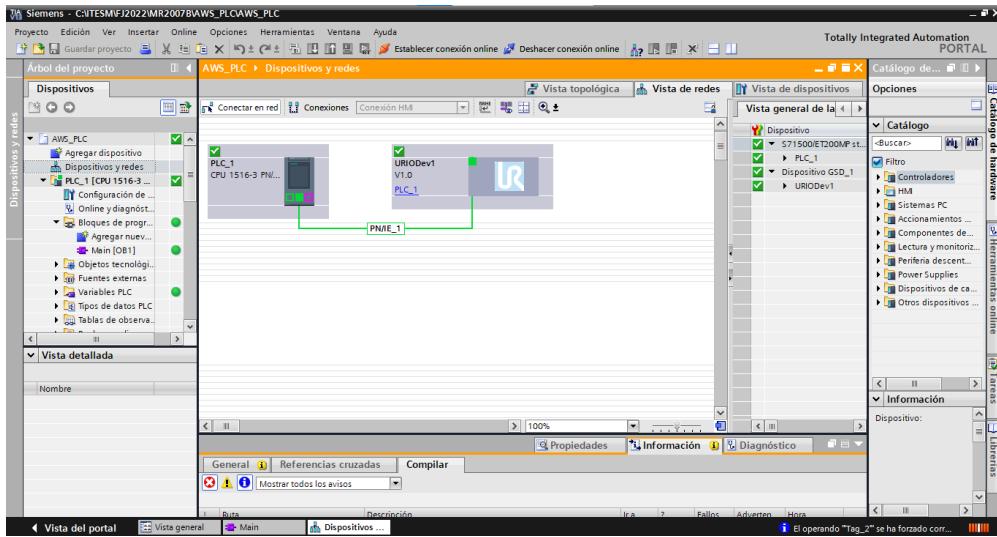
Enabled 1 node uses this config



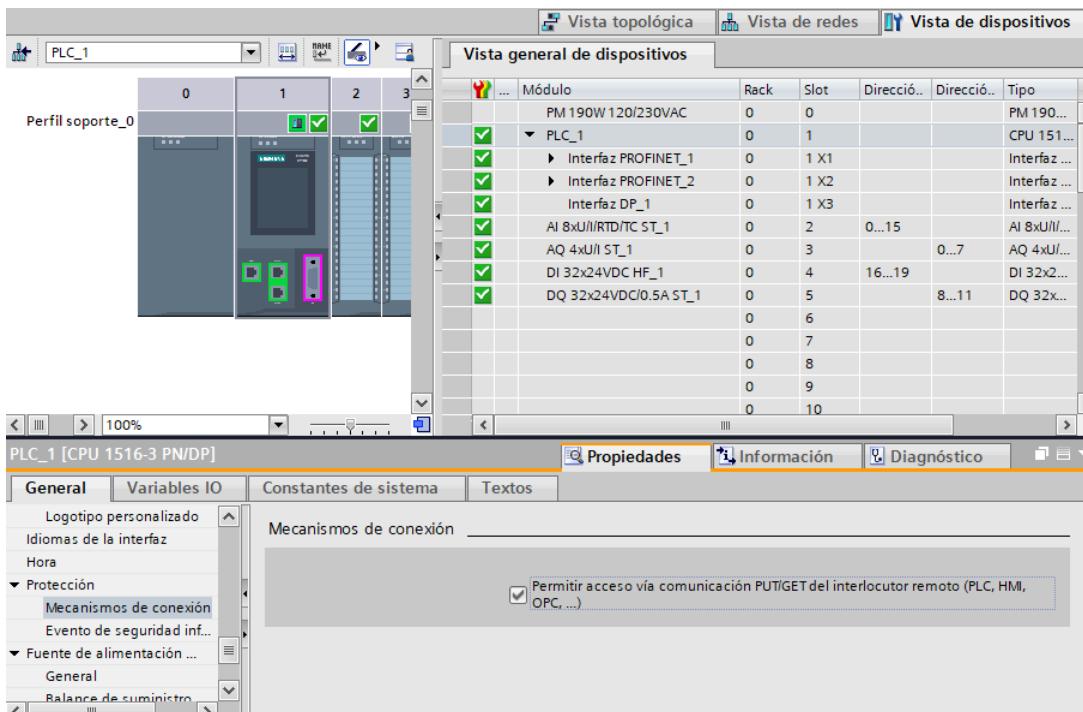
Certificados que teníamos en AWS

PLC / Siemens S7-1500

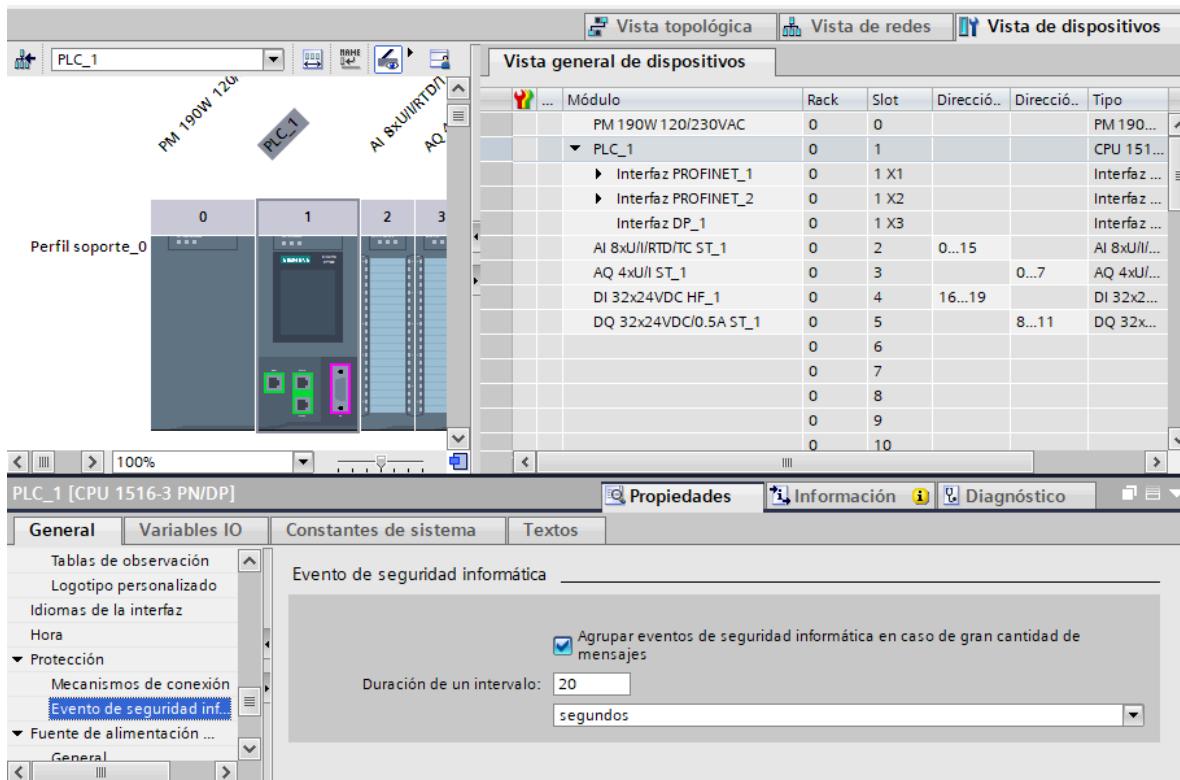
Configuración PLC



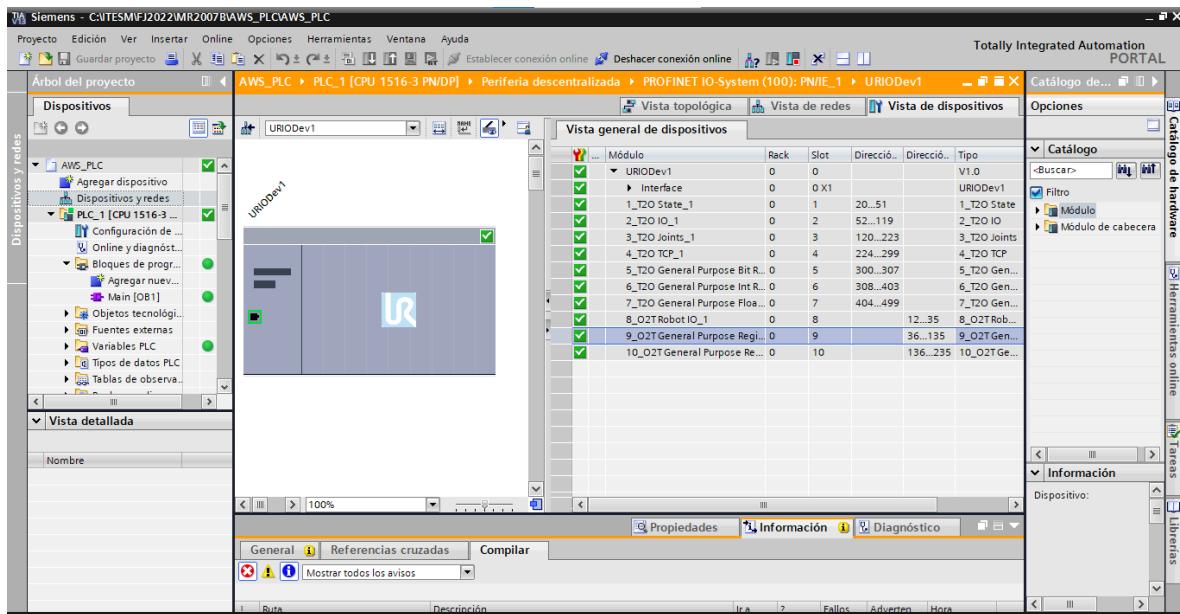
Después de hacer la configuración de nuestro PLC con el UR, como ya hemos visto en cursos anteriores



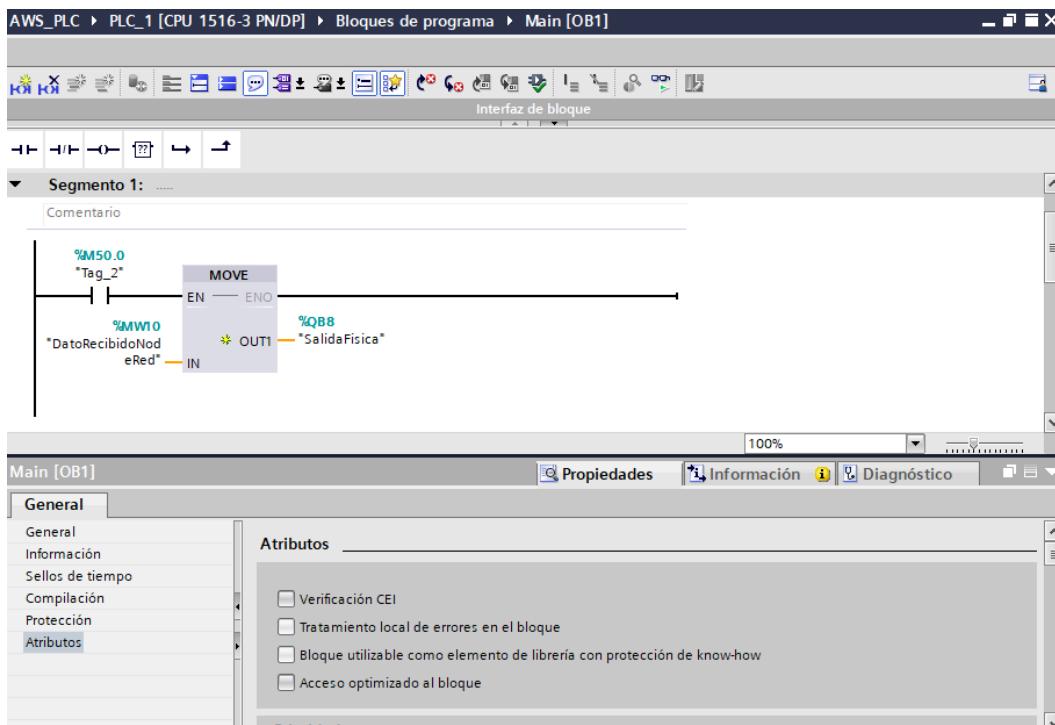
Hay que tener en cuenta que debemos de activar en nuestro PLC en [Protección>Mecanismos de conexión> Permitir acceso via...]



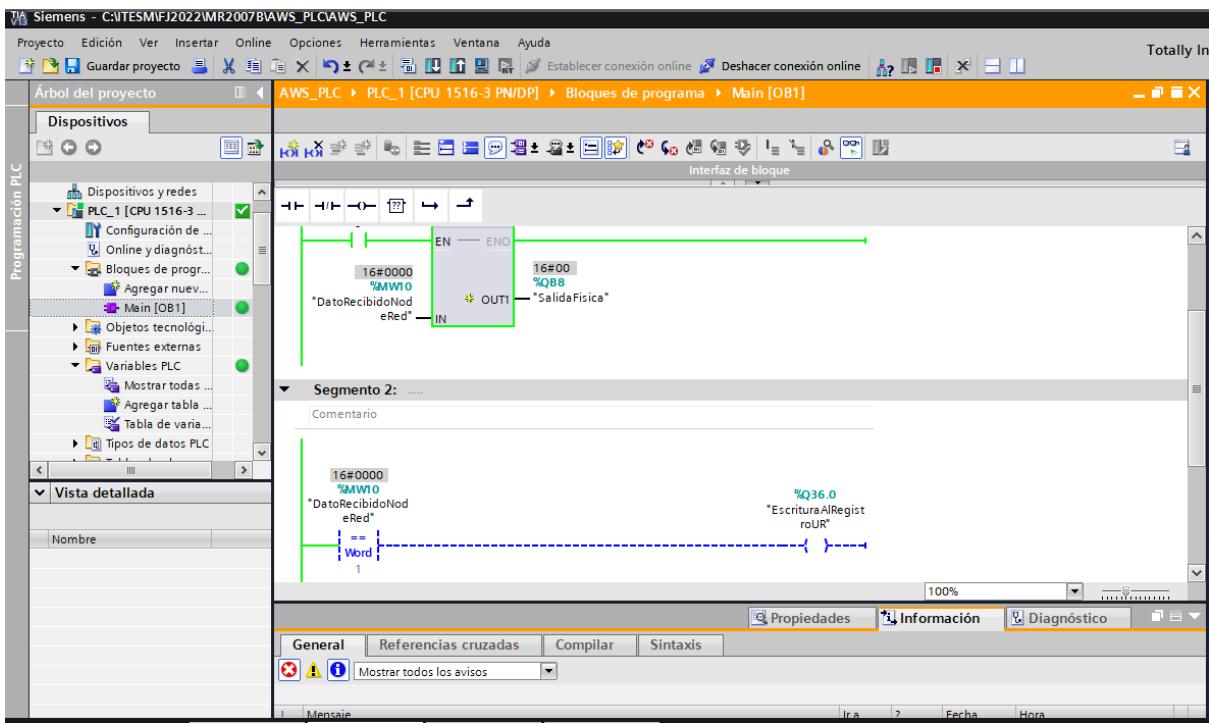
También activar en Eventos de seguridad el agrupar eventos de seguridad



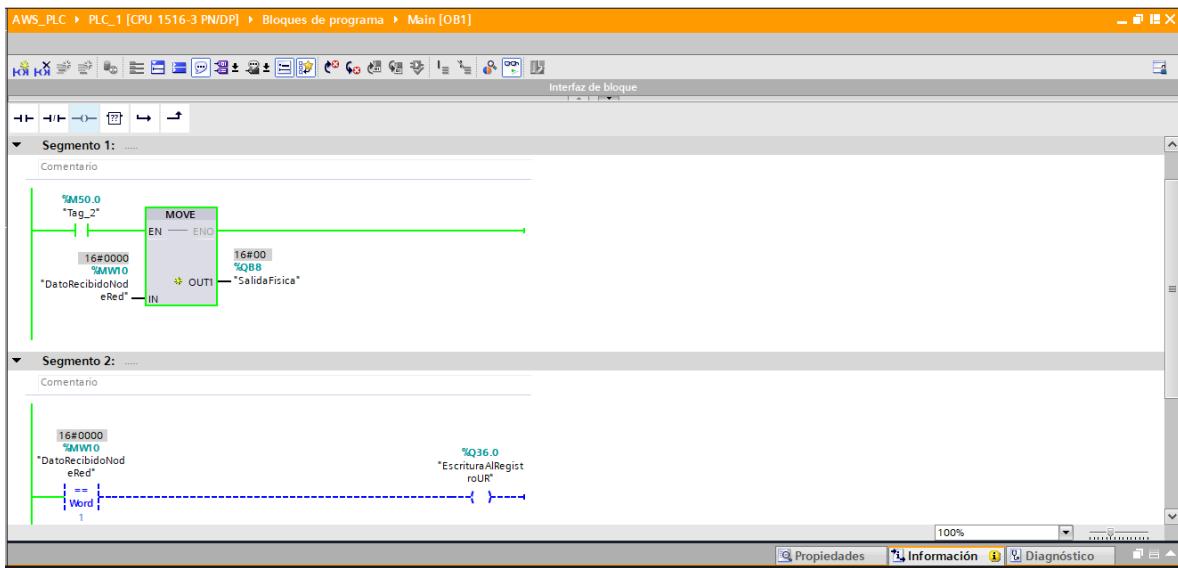
Realizamos la configuración del UR en nuestro PLC y debemos de recordar que para mandar bits de activacion y desactivacion debemos de trabajar con la salida Q36 en nuestro caso, ya que fue la que asignó el PLC



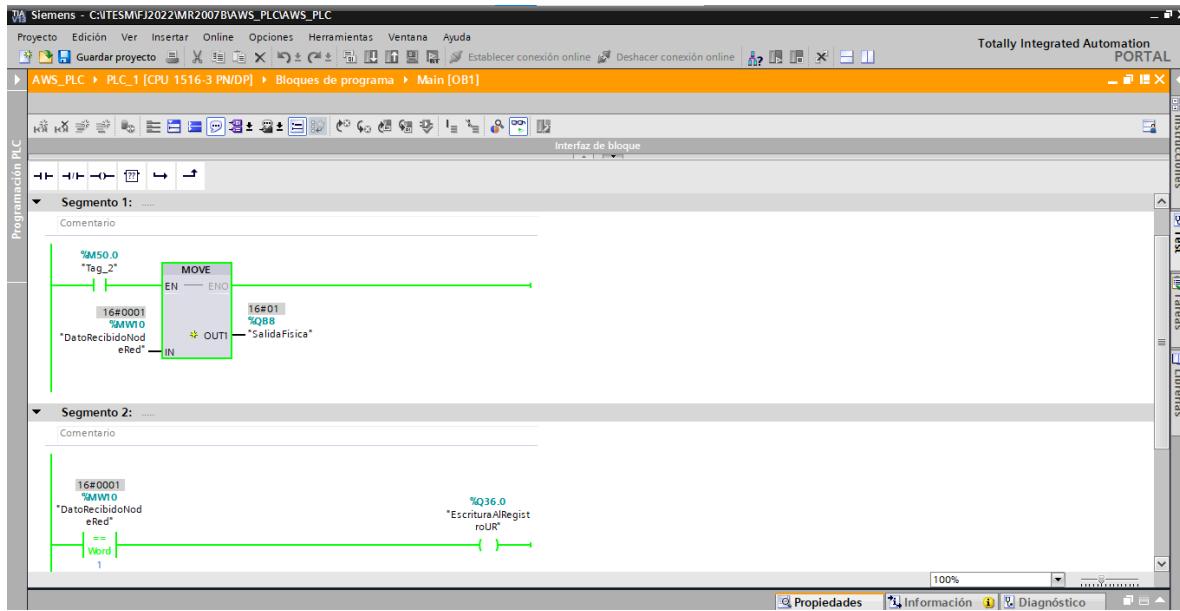
Una vez en el OB1 creamos un bloque en el que moveremos nuestra memoria que creamos en la red de Node-red y la colocamos en una entrada para verificar que nos este llegando y la mandamos a una salida física de nuestro PLC / ATENCIÓN!! Debemos de tener en cuenta que debemos de irnos a atributos y desactivar todo lo que tengamos activado, esto para poder recibir los datos en el PLC desde Node-Red



Y a su vez colocamos un comparador, para que cuando el valor recibido sea igual a 1, este active nuestro registro que tendremos en el Cobot para poder controlarlo.

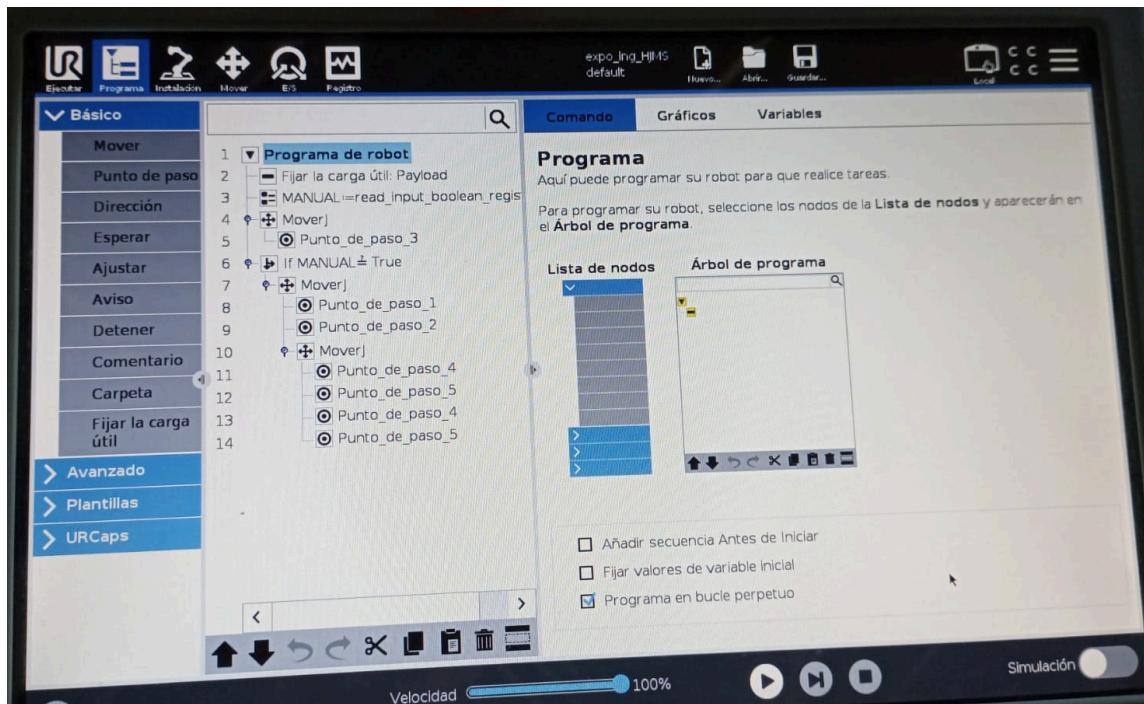


Así es como nos queda nuestro programa.



Y así cada que recibimos un 1 desde Node-Red

UR3



La configuración del UR3 y código debe de ser el siguiente, debemos de tener en cuenta que lo estamos guardando en el registro 0 del cobot, por ende será el “read_input_boolean_register(0)” y cuando esta condición sea igual a 1, activará la secuencia dada en la programación del mismo cobot.

Conclusiones

El proyecto demostró la capacidad de integrar eficientemente múltiples tecnologías y plataformas para lograr un sistema de automatización robusto y flexible. A través de MATLAB, se logró implementar un sistema de reconocimiento de imágenes que detecta la forma y orientación de una manga. Esta detección desencadena un flujo de comunicación que pasa por un ESP32, Adafruit, Node-Red, AWS, y finalmente se conecta a TIA Portal de Siemens para controlar un robot UR3.

La comunicación serial entre MATLAB y el ESP32 funcionó de manera óptima, permitiendo la transferencia precisa del dato booleano necesario para la activación del proceso. La transmisión de datos a través de Wi-Fi al broker Adafruit y la utilización del protocolo MQTT mostraron ser métodos efectivos para el manejo de la información. Node-Red desempeñó un papel crucial en la orquestación de los datos, facilitando la comunicación entre las diversas plataformas involucradas.

La utilización de AWS como base de datos en la nube demostró ser una solución eficaz para el almacenamiento y recuperación de datos, garantizando que la información esté disponible desde cualquier parte del mundo. La integración de Profinet con TIA Portal permitió una comunicación fluida con el PLC del UR3, asegurando que los datos recibidos se asignen correctamente a los registros de entradas y salidas del robot.

El proyecto cumplió con su objetivo principal, logrando automatizar el proceso de detección y activación mediante la integración de tecnologías avanzadas. La secuencia predefinida en el UR3 se ejecutó con precisión, validando la efectividad del sistema desarrollado.

Futuras Mejoras

A pesar del éxito alcanzado, siempre existen oportunidades para optimizar y mejorar el sistema. Una de las mejoras más significativas que se propone es la implementación del cálculo de la cinemática inversa directamente desde MATLAB. Actualmente, el UR3 realiza estos cálculos de manera interna, pero al externalizar este proceso, podríamos obtener varias ventajas.

Primero, al calcular la cinemática inversa desde MATLAB, podríamos utilizar algoritmos más avanzados y personalizados, optimizando la precisión y eficiencia del robot en sus movimientos. Esto permitiría una mayor flexibilidad en la programación de tareas complejas, adaptando las trayectorias del robot a necesidades específicas y mejorando su rendimiento en aplicaciones industriales.

Además, esta mejora facilita la integración de MATLAB con otras herramientas y plataformas de simulación y control, proporcionando un entorno más cohesivo para el desarrollo y prueba de nuevas aplicaciones robóticas. La capacidad de realizar cálculos externos y enviar comandos precisos al UR3 abriría la puerta a nuevas posibilidades en la automatización, permitiendo el control remoto y la supervisión en tiempo real de las operaciones del robot.

Otra posible mejora sería la implementación de técnicas de aprendizaje automático para mejorar el sistema de reconocimiento de imágenes. Utilizando modelos de aprendizaje profundo, podríamos aumentar la precisión y velocidad del proceso de detección, adaptándolo a una variedad más amplia de objetos y escenarios. Esto haría que el sistema sea más versátil y aplicable a diferentes industrias y casos de uso.

Finalmente, la seguridad y la redundancia de la comunicación pueden mejorarse mediante la implementación de protocolos de cifrado y sistemas de respaldo. Esto garantizaría la integridad y confiabilidad de los datos transmitidos, reduciendo el riesgo de fallos en la comunicación y asegurando un funcionamiento continuo del sistema.

En resumen, el proyecto ha sentado una base sólida para la automatización avanzada utilizando MATLAB y el robot UR3. Las futuras mejoras propuestas no solo optimizarán el rendimiento del sistema, sino que también ampliarán sus capacidades, abriendo nuevas oportunidades para la innovación y el desarrollo en el campo de la robótica y la automatización.

Bibliografías

1. https://www.youtube.com/playlist?list=PLGMZwZq6OIt8PV_e2Dhib03P17znfePjg
2. <https://forum.universal-robots.com/t/ur-vs-siemens-consultar-programa-cargado-en-robot/27637>
3. <https://nodejs.org/>
4. <https://la.mathworks.com/solutions/image-video-processing/video-processing.html>