**20 de ENERO de 2022**

Forma

Descripción generada automáticamente con confianza bajaIcono

Descripción generada automáticamenteLogotipo, Icono

Descripción generada automáticamenteLogotipo

Descripción generada automáticamenteIcono

Descripción generada automáticamenteIcono

Descripción generada automáticamente

**PROYECTO IoT**

**CURSO:**

4º Grado en Ingeniería Electrónica, Robótica y Mecatrónica

**ASIGNATURA:**

Informática Industrial

**AUTORES:**

Luis Calderón Robustillo – 07274306G

David Rodríguez Onieva – 51211340T

Rubén González Navarro - 77493894V

Pedro Antonio Peña Puente - 77228494V

ÍNDICE

[1.- INTRODUCCIÓN 1](#_Toc94279518)

[1.1.- Descripción del Proyecto 1](#_Toc94279519)

[1.2.- Elementos Utilizados 1](#_Toc94279520)

[1.3.- Esquema General 1](#_Toc94279521)

[2.- TRELLO 2](#_Toc94279522)

[2.1.- Seguimiento del Proyecto 2](#_Toc94279523)

[3.- ELEMETOS Y HERRAMIENTAS IMPLEMENTADOS 2](#_Toc94279524)

[3.1.- Telegram 2](#_Toc94279525)

[3.2.- FOTA 2](#_Toc94279526)

[3.3.- Sensores 2](#_Toc94279527)

[3.3.1.- Interruptor Táctil TTP223B 2](#_Toc94279528)

[3.4.- ALEXA 2](#_Toc94279529)

[ANEXO I 3](#_Toc94279530)

# 1.- INTRODUCCIÓN

Se va a desarrollar un proyecto IoT que a rasgos generales será un proyecto de asistencia al usuario en iluminación, temperatura y audio en el hogar.

## 1.1.- Descripción del Proyecto

La ESP8266 es un módulo para Arduino de bajo costo que permite la conexión Wi-Fi de la placa en cuestión con la red. Esto permitió que todos los proyectos que hasta ahora requerían conexión física pudiesen trabajar a distancia, controlar desde una puerta de garaje, hasta una alarma casera o construir un controlador LED. En nuestro caso, nos adaptaremos a la comodidad del usuario, específicamente en el hogar. El proyecto ofrecerá un servicio de variación de luminosidad de la estancia, otro de estabilización de la temperatura en una referencia agradable y un servicio de sugerencias de música. El sistema incluirá el uso de la aplicación de *Telegram* como interfaz de interacción con el usuario. En ella el usuario podrá elegir entre utilizar las canciones, la temperatura y la luminosidad sugerida o utilizar una personalizada. Las canciones se reproducirán mediante el asistente virtual *Alexa*, para la temperatura se usará un aire acondicionado y para la luminosidad una cinta de LEDs con sensor infrarrojo.

El algoritmo que determinará las sugerencias ofrecidas al usuario se basará en gran medida en la meteorología del exterior. En especial con la medida de la luminosidad exterior que llega a la habitación y la temperatura que hace en la propia habitación. Adicionalmente, el proyecto contará con un sistema de activación mediante sensor táctil, y un sistema de renovación de la sesión mediante un sensor de movimiento. Es decir, el sistema permanecerá apagado hasta que el usuario interaccione con el sensor táctil. El sistema, una vez pasado un tiempo, se apagará suponiendo que nadie lo está usando (sensor de presencia no detecta nada). No obstante, una vez activado no será estrictamente necesario renovar la sesión mediante el sensor táctil, sino que el sistema contará con un sensor de movimiento que detectará el movimiento del usuario por la habitación.

## 1.2.- Elementos Utilizados

Como elementos utilizados para el desarrollo del proyecto, se han usado cuatro módulos ESP-8266, sensores de temperatura y humedad DH11, sensores de movimiento HC-SR501, sensores de luminosidad GL55, un interruptor táctil TTP223B, un aire acondicionado de la compañía Mitsubishi, varios emisores de infrarrojos TSUS4300 y receptores KY-022, un asistente virtual ‘Alexa Echo Dot de 3ª Generación’ y una cinta de iluminación LED.

En cuanto al software utilizado para el interconexionado de todos los elementos, se ha hecho uso de Node-Red, Arduino IDE, Telegram, MQTT Broker Mosquitto, ESP-NOW, MongoDB y Amazon Alexa.

Se hará un pequeño estudio de todos estos elementos más adelante.

## 1.3.- Esquema General

Hay dos *topics* como se puede ver en la Figura X que se usan para la meteorología si el sistema está activo. Como se ha comentado en el apartado anterior, la activación del sistema se realizará mediante un sensor de presencia y/o un sensor táctil. Una vez que tengamos estos *topics*, estos irán a *Telegram*. Mediante funciones convertiremos estos topics en tiempo nublado, lluvioso, soleado, … En función de la meteorología detectada, se recomendará por *Telegram* una *playlist* y una iluminación. El usuario será el encargado de responder si desea esta iluminación y música o ser él mismo el que elija otras.

Una vez que el usuario ha respondido y tenemos esa información y los demás datos, a través de *Node-Red* se enviarán ciertos comandos a *Alexa* y ciertas órdenes a las luces. Además de esta información obtenida por el usuario y la meteorología, se podrá decidir por *Telegram* si se desea encender un aire acondicionado y a qué temperatura. Siendo una placa de tipo esclavo quien se encargue de realizar esta última función.

A fin de poder realizar una mejor recomendación de iluminación y música se ha optado por hacer uso de una base de datos para ir almacenando la lectura de los sensores y cuando se vaya a hacer una recomendación al usuario el resultado proporcionado sea más acertado.

# 2.- TRELLO

## 2.1.- Seguimiento del Proyecto

# 3.- ELEMETOS Y HERRAMIENTAS IMPLEMENTADOS

## 3.1.- Telegram

Haciendo uso de los proyectos realizados previamente en la asignatura, hemos implementado un código similar en nuestro proyecto en Node-Red, con la diferencia de que ahora los comandos de Telegram se usan principalmente para controlar ciertas rutinas creadas en Alexa. Como hemos explicado previamente, tendremos un comando para seguir la recomendación acorde con el tiempo y otros comandos para elegir tu propio estilo de música e iluminación como \alegre, \triste, \bailar o \chill. Lo más trabajoso de esta parte fue ser capaces de sincronizar Alexa con Nodered pero con la ayuda del profesor lo conseguimos.

## 3.2.- FOTA

Al igual que con Telegram, reutilizamos códigos hechos previamente en la asignatura con los cuales pudimos implementar FOTA para actualizar el código directamente desde la nube.

## 3.3.- Sensores

En este proyecto hemos usado múltiples sensores:

### 3.3.1.- Interruptor Táctil TTP223B

Un sensor táctil capacitivo es un dispositivo que presenta un comportamiento similar a un pulsador, pero puede ser activado con poca o ninguna presión.

En este proyecto, el uso de este sensor se ha destinado a la activación del sistema. Cuando el usuario active el sensor, se enviará una señal que habilite todo el sistema y a su vez se comenzará una cuenta atrás para la posterior desactivación del sistema.

### 3.3.2.- Sensor de Movimiento HC-SR501

Los sensores de movimiento son sensores infrarrojos pasivos (PIR). Debido al bajo costo, consumo y fácil uso se ha optado para usarlo en nuestra aplicación domótica.

En este caso, se ha usado el sensor para reestablecer la cuenta atrás de la desactivación del sistema. Esto se hace por un lado para hacer un uso eficiente de la energía y para la comodidad del usuario para que no tenga que acercarse a activar el sensor táctil cada cierto tiempo para no interrumpir sus actividades.

## 3.3.3.- Sensor de temperatura y humedad DHT11

Hemos reutilizado este sensor con el cual hemos estado trabajando durante todo el curso. Este sensor es una parte fundamental de nuestro proyecto ya que junto al sensor de luminosidad nos permite ofrecer una recomendación acorde con el tiempo de un estado de ánimo para posteriormente poner la música e iluminación adecuadas.

## 3.3.4.- Sensor de luminosidad

Hemos usado este sensor para detectar cambios de luz del exterior (soleado o nublado). No ha sido complicado implementarlo ya que tiene un funcionamiento muy simple.

## 3.3.5.- Sensores emisor y receptor de infrarrojos

Estos sensores han sido necesarios para el correcto desarrollo de la práctica, ya que hemos tenido que descifrar las señales infrarrojas con las que funcionaban los leds y el aire acondicionado, y después comprobar si se enviaban correctamente

## 3.4.- ALEXA

# ANEXO I

*Códigos y Flujos.*