



# COMUNICACIONES AVANZADAS

## PROYECTO FINAL

Integrantes: Vinicio Guzman, Kevin Zuñiga, Carlos Orozco  
Profesor: Santiago Manzano

Fecha de realización: 17 de agosto de 2021  
Fecha de entrega: 17 de agosto de 2021  
Ambato, Ecuador

# Resumen

En el presente proyecto se desarrolló una alarma utilizando herramientas tecnológicas que nos permiten la comunicación inalámbrica además el accionamiento de los diferentes sensores con la utilización de Telegram. También para el proyecto se ha utilizado una sp-32 cam la cual nos permite realizar la toma de fotografías de las personas que accionan nuestra alarma para lo cual se ha desarrollado un archivo php el cual da instrucciones a nuestro servidor para la ejecución de nuestro programa en la ESP8266 el mismo que permite la comunicación inalámbrica y así tener informado al usuario de quién se encuentra accionando la alarma.

# Índice de Contenidos

1. OBJETIVO GENERAL	1
2. OBJETIVOS ESPECIFICOS	1
3. TELEGRAM	1
TELEGRAMA: DETECCIÓN DE MOVIMIENTO ESP8266 NODEMCU CON NOTIFICACIONES (ARDUINO IDE)	1
CREACIÓN DE UN BOT DE TELEGRAM	3
OBTENER ID DE USUARIO DE TELEGRAM	5
4. ARDUINO IDE	6
Biblioteca de bots de Telegram universal.	6
Biblioteca ArduinoJson.	6
5. Configuración de la ESP32-CAM	6
6. Formateo de la tarjeta MicroSD	7
7. ESP32-CAM Publica imágenes en un servidor local o en la nube usando PHP	7
8. Alojamiento de su aplicación PHP	8
9. Servidor en la nube	9
10. ESP32-CAM HTTP Publicar imágenes	9
11. RESULTADOS	9
ESP32-CAM Toma una foto con PIR Sketch	10
12. CONCLUSIONES	11
Referencias	12

# Índice de Figuras

1. Estructura. . . . .	2
2. Instalacion de Telegram . . . . .	3
3. Crear un Telegram Bot . . . . .	3
4. Inicio de Telegram . . . . .	4
5. Configuracion de boot . . . . .	4

6.	Acceso al bot . . . . .	4
7.	IDBot . . . . .	5
8.	Identificación de usuario . . . . .	5
9.	Formateo MicroSD . . . . .	7
10.	Proceso de formateo . . . . .	7
11.	Servidor local Máquina Virtual RED HAT . . . . .	8
12.	Servidor en la nube . . . . .	9
13.	Diagrama de conexion . . . . .	10
14.	Prototipo . . . . .	10

## 1. OBJETIVO GENERAL

Diseñar un sistema de control y monitoreo de una alarma con la utilización de la aplicación Telegram y una ESP32-CAM.

## 2. OBJETIVOS ESPECIFICOS

- Investigar los aspectos mas importantes de telegram y las características del mismo.
- Investigar las bibliotecas de Arduino para la comunicacion con telegram
- Diseñar la metodología de implementación en un sistema de alarma utilizando APIS para modelar el prototipo experimental.

## 3. TELEGRAM

Telegram Messenger es un servicio de mensajería instantánea y voz sobre IP basado en la nube. Puede instalarlo fácilmente en su teléfono inteligente (Android y iPhone) o computadora (PC, Mac y Linux). Es gratis y sin publicidad. Telegram te permite crear bots con los que puedes interactuar. “Los bots son aplicaciones de terceros que se ejecutan dentro de Telegram. Los usuarios pueden interactuar con los bots enviándoles mensajes, comandos y solicitudes en línea. Controlas tus bots usando solicitudes HTTPS a la API de Telegram Bot ”. El ESP8266 interactuará con el bot de Telegram para enviar mensajes a su cuenta de Telegram. Siempre que se detecte movimiento, recibirá una notificación en su teléfono inteligente (siempre que tenga acceso a Internet).

## TELEGRAMA: DETECCIÓN DE MOVIMIENTO ESP8266 NODEMCU CON NOTIFICACIONES (ARDUINO IDE)

En esta etapa del proyecto muestra cómo enviar notificaciones a su cuenta de Telegram cuando el ESP8266 NodeMCU detecta movimiento. Siempre que tenga acceso a Internet en su teléfono inteligente, se le notificará sin importar dónde se encuentre. La placa ESP se programará utilizando Arduino IDE

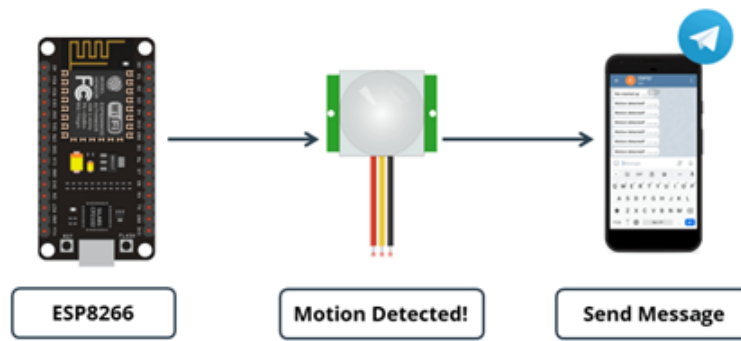


Figura 1: Estructura.

Descripción general:

- Creará un bot de Telegram para su ESP8266.
- El ESP8266 está conectado a un sensor de movimiento PIR.
- Cuando el sensor detecta movimiento, el ESP8266 envía un mensaje de advertencia a su cuenta de telegram.
- Se le notificará en su cuenta de telegram cada vez que se detecte movimiento.

# CREACIÓN DE UN BOT DE TELEGRAM

Vaya a Google Play o App Store, descargue e instale Telegram

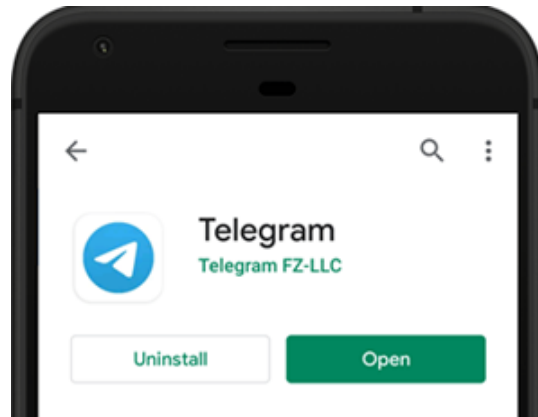


Figura 2: Instalacion de Telegram

Abra Telegram y sigue los siguientes pasos para crear un Telegram Bot. Primero, busque "botfather" haga clic en BotFather como se muestra a continuación. O abra este enlace [t.me/botfather](https://t.me/botfather) en su teléfono inteligente.

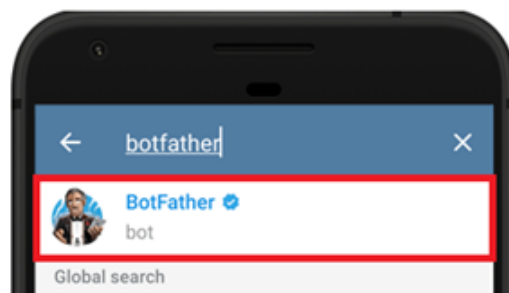


Figura 3: Crear un Telegram Bot

La siguiente ventana debería abrirse y se le pedirá que haga clic en el botón de inicio.

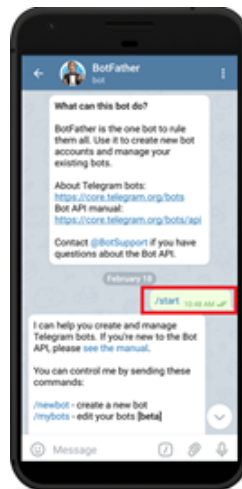


Figura 4: Inicio de Telegram

Escriba / newbot y siga las instrucciones para crear su bot. Dale un nombre y un nombre de usuario



Figura 5: Configuración de bot

Si su bot se crea correctamente, recibirá un mensaje con un enlace para acceder al bot y al token del bot. Guarde el token del bot porque lo necesitará para que el ESP8266 pueda interactuar con el bot.

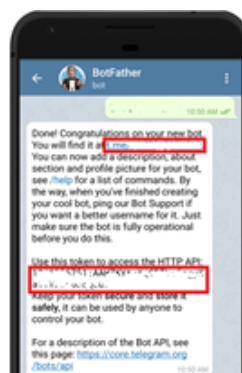


Figura 6: Acceso al bot



## OBTENER ID DE USUARIO DE TELEGRAM

Cualquiera que conozca el nombre de usuario de su bot puede interactuar con él. Para asegurarse de que ignoramos los mensajes que no son de nuestra cuenta de Telegram (o de cualquier usuario autorizado), puede obtener su ID de usuario de Telegram. Luego, cuando su bot de telegramas recibe un mensaje, el ESP puede verificar si la identificación del remitente corresponde a su identificación de usuario y manejar el mensaje o ignorarlo.

En su cuenta de Telegram, busque “IDBot” o abra este enlace [t.me/myidbot](https://t.me/myidbot) en su teléfono inteligente.

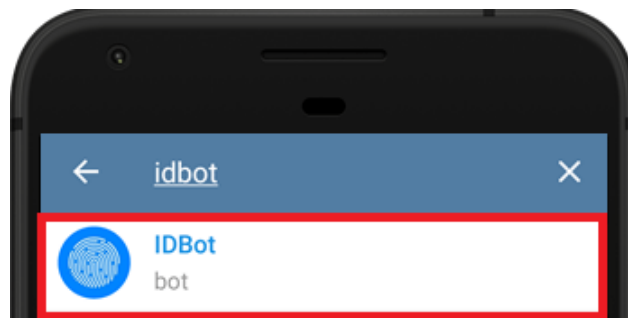


Figura 7: IDBot

Inicie una conversación con ese bot y escriba `/getid`. Recibirá una respuesta con su ID de usuario. Guarde esa identificación de usuario, ya que la necesitará más adelante en este tutorial.

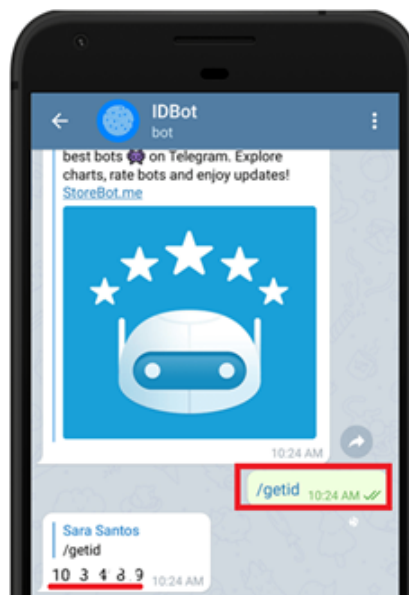


Figura 8: Identificación de usuario

## 4. ARDUINO IDE

Programaremos la placa ESP8266 usando Arduino IDE, así que asegúrese de tenerlos instalados en su Arduino IDE. Instalación de la placa ESP8266 en Arduino IDE (Windows, Mac OS X, Linux).

### Biblioteca de bots de Telegram universal.

Para interactuar con el bot de Telegram, usaremos la biblioteca de bots de Telegram universal creada por Brian Lough que proporciona una interfaz fácil para la API del bot de Telegram. Pasos para instalar la última versión de la biblioteca.


- Haga clic aquí para descargar la biblioteca Universal Arduino Telegram Bot.
- Ir a Boceto > Incluir Biblioteca > Add.ZIP Biblioteca.
- Agregue la biblioteca que acaba de descargar.

**Importante:** no instale la biblioteca a través de Arduino Library Manager porque podría instalar una versión obsoleta. Para obtener todos los detalles sobre la biblioteca, eche un vistazo a la página de GitHub de la biblioteca de bots de Telegram de Universal Arduino

### Biblioteca ArduinoJson.

También debe instalar la biblioteca ArduinoJson . Siga los siguientes pasos para instalar la biblioteca.

- Vaya a Sketch > Incluir biblioteca > Administrar bibliotecas
- Busque "ArduinoJson".
- Instale la biblioteca.

Estamos usando la versión 6.5.12 de la biblioteca ArduinoJson.  scale=0.90 Version de ArduinoJson

## 5. Configuración de la ESP32-CAM

Descripción general rápida de cómo funciona el proyecto.

- El ESP32-CAM está en modo de suspensión profunda con la activación externa habilitada.
- Cuando se detecta movimiento, el sensor de movimiento PIR envía una señal para activar el ESP32.
- La ESP32-CAM toma una foto y la guarda en la tarjeta microSD.
- Vuelve al modo de reposo profundo hasta que se recibe una nueva señal del sensor de movimiento PIR.

## 6. Formateo de la tarjeta MicroSD

Inserte la tarjeta microSD en su computadora. Vaya a Mi PC y haga clic derecho en la tarjeta SD. Seleccione Formato como se muestra en la figura siguiente.

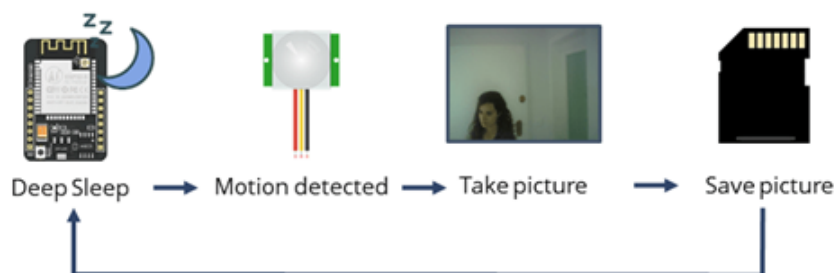


Figura 9: Formateo MicroSD

Aparece una nueva ventana. Seleccione FAT32, presione Iniciar para inicializar el proceso de formateo y siga las instrucciones en pantalla.

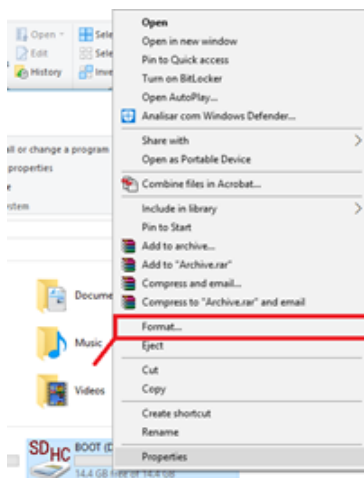


Figura 10: Proceso de formateo

## 7. ESP32-CAM Publica imágenes en un servidor local o en la nube usando PHP

Para construir este proyecto, debe seguir los siguientes pasos. Siga las instrucciones del servidor LAMP o del servidor de alojamiento dependiendo de si desea acceder a las fotos localmente o desde cualquier lugar.

- Alojamiento de su aplicación PHP.
- Servidor Raspberry Pi LAMP (acceso local).

- Servidor de alojamiento (acceso desde cualquier lugar).
- Scripts PHP para guardar y mostrar fotos en el servidor.
- Servidor Raspberry Pi LAMP (acceso local).
- Programe el ESP32-CAM con Arduino IDE.
- Prueba y demostración final

## 8. Alojamiento de su aplicación PHP

El objetivo de este proyecto es tener un servidor local o en la nube para almacenar y acceder a sus fotos ESP32-CAM. Servidor local Máquina Virtual RED HAT Con un servidor LAMP de Raspberry Pi , puede acceder a sus imágenes localmente (como se ilustra a continuación).

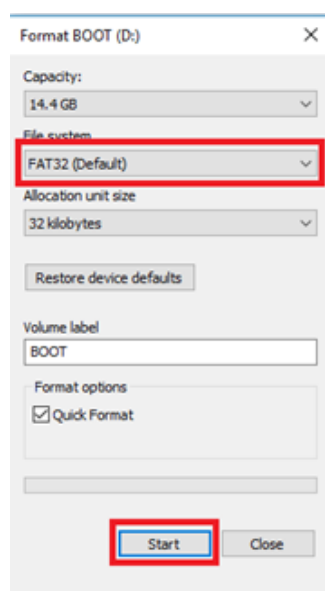


Figura 11: Servidor local Máquina Virtual RED HAT

Puede ejecutar un servidor LAMP (Linux, Apache, MySQL, PHP) en una máquina virtual RED HAT para acceder a los datos en su red local . Servidor LAMP de REDHAT: servidor Linux local que utiliza para acceder a sus imágenes localmente

## 9. Servidor en la nube

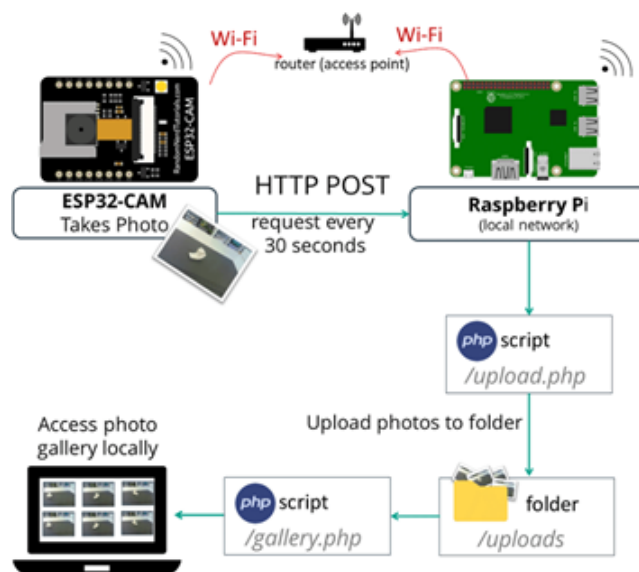


Figura 12: Servidor en la nube

Bluehost (fácil de usar con cPanel) : nombre de dominio gratuito cuando te registras en el plan de 3 años. Recomiendo elegir la opción de sitios web ilimitados; Tenga en cuenta que cualquier servicio de alojamiento que ofrezca PHP funcionará con este tutorial. Si no tiene una cuenta de alojamiento, le recomiendo que se registre en Bluehost. Método de solicitud HTTP POST

- El Protocolo de transferencia de hipertexto (HTTP) funciona como un protocolo de solicitud-respuesta entre un cliente y un servidor.
- El ESP32 (cliente) envía una solicitud HTTP a un servidor (por ejemplo: servidor RPi Lamp local o example.com);
- El servidor devuelve una respuesta al ESP32 (cliente);

## 10. ESP32-CAM HTTP Publicar imágenes

Ahora que ya tienes tu servidor listo (servidor REDHAT LAMP o servidor en la nube), es hora de preparar la ESP32-CAM con el código para publicar una nueva imagen en tu servidor cada 30 segundos. Antes de continuar con este tutorial, asegúrese de completar los siguientes requisitos previos.

## 11. RESULTADOS

## ESP32-CAM Toma una foto con PIR Sketch

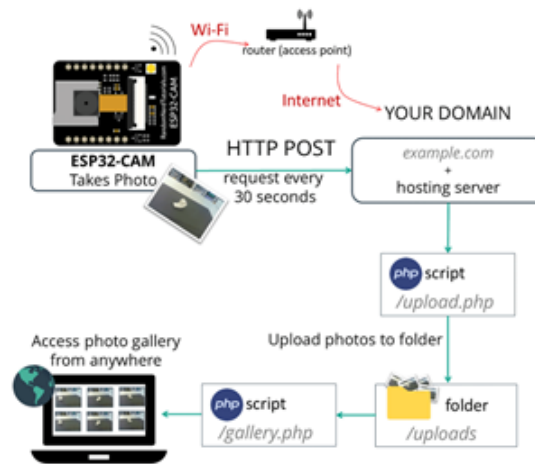


Figura 13: Diagrama de conexión

Código fuente para la programación

[https://drive.google.com/file/d/15umW9y-a6tKW3cI28T8F0LaNOyZZCcm\\_/view?usp=sharing](https://drive.google.com/file/d/15umW9y-a6tKW3cI28T8F0LaNOyZZCcm_/view?usp=sharing)

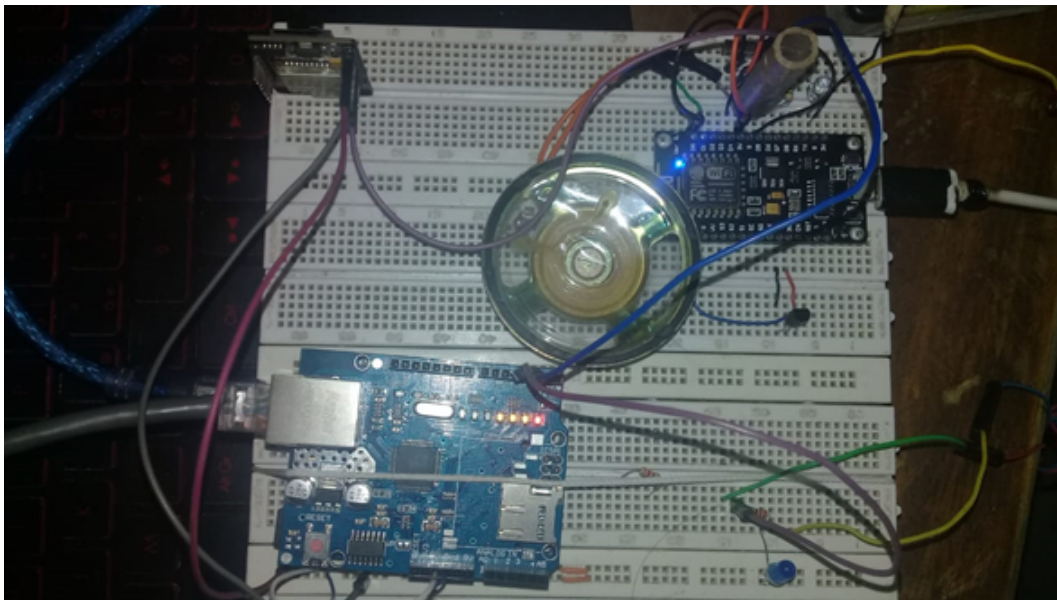


Figura 14: Prototipo

## 12. CONCLUSIONES

- Telegram api bot es una pequeña utilidad para programar tu propia bot api lo que permite a los programadores crear su propio entorno con diferentes dispositivos electronicos.
- Las funcionalidades de asterisk permiten obtener un prototipo experimental mas complejo con mayores prestaciones.
- El estudio de las diferentes formas de realizar la comunicación entre nuestro servidor y la aplicación Telegram nos permitió conocer que la comunicación se la puede realizar de manera inalámbrica para lo cual se utiliza una esp8266 además nuestro caso se ha utilizado un servidor en la nube para el almacenamiento de las imágenes obtenidas a través de la esp 32 CAM.
- La implementación del prototipo nos permite verificar el correcto funcionamiento de nuestra alarma la cual funciona con nuestro servidor el cual es el encargado de comunicar la ESP-32 CAM y la aplicación de Telegraman.

## Referencias

- [1] Telegram *telegram*  
<https://randomnerdtutorials.com/telegram-esp8266-nodemcu-motion-detection-arduino/>
- [2] photo *PHOTO*  
<https://randomnerdtutorials.com/esp32-cam-pir-motion-detector-photo-capture/>
- [3] ESP32 *ESP*  
<https://randomnerdtutorials.com/esp32-cam-post-image-photo-server/>