UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE INGENIERÍA ELECTROMECÁNICA
PROPUESTA TECNOLÓGICA

"DESARROLLO DE UN SISTEMA DE CONTROL DE ÁNGULOS DE CÁMARAS IP MEDIANTE IOT PARA EL MONITOREO EN TIEMPO REAL DE UNA VIVIENDA"

Proyecto de Titulación presentando previo a la obtención del título de Ingeniero Electromecánico

Autor:
Angulo Chiliquinga Cristian Omar

Tutor:
Ing. MSc. Byron Paúl Corrales Bastidas

Latacunga – Ecuador
2023

DECLARACIÓN DE AUTORÍA

Yo, Cristian Omar Angulo Chiliquinga, expreso ser autor de la siguiente Propuesta Tecnológica: "**DESARROLLO DE UN SISTEMA DE CONTROL DE ÁNGULOS DE CÁMARAS IP MEDIANTE IOT PARA EL MONITOREO EN TIEMPO REAL DE UNA VIVIENDA**" siendo el Ing. MSc. Byron Paúl Corrales Bastidas el tutor del presente trabajo; y exime expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certificamos que las ideas, conceptos, procedimientos y resultados vertidos en la siguiente Propuesta Tecnológica, son de nuestra exclusiva responsabilidad.

Latacunga, febrero 2023

……

Cristian Omar Angulo Chiliquinga

C.I. 185008482-1

AVAL DEL TUTOR DEL PROYECTO DE TITULACIÓN

En calidad de tutor de la siguiente propuesta tecnológica sobre el título:

"**DESARROLLO DE UN SISTEMA DE CONTROL DE ÁNGULOS DE CÁMARAS IP MEDIANTE IOT PARA EL MONITOREO EN TIEMPO REAL DE UNA VIVIENDA**", de Angulo Chiliquinga Cristian Omar, estudiante de la carrera de Ingeniería Electromecánica, considero que dicho informe investigativo cumple con los requerimientos metodológicos y aportes Científico – Técnico suficiente para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad técnica de Cotopaxi designe para su correspondiente estudio y calificación.

Latacunga, febrero 2023

……..……

Ing. MSc. Byron Paúl Corrales Bastidas

C.I. 050234776-8

APROVACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe Tecnológico de acuerdo a las disposiciones reglamentarias emitidas por la Universidad técnica de Cotopaxi, y por la FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS: por cuanto, el postulante: Angulo Chiliquinga Cristian Omar con cédula de ciudadanía de ciudadanía N°.185008482-1. Con el Título de Proyecto de Titulación: "**DESARROLLO DE UN SISTEMA DE CONTROL DE ÁNGULOS DE CÁMARAS IP MEDIANTE IOT PARA EL MONITOREO EN TIEMPO REAL DE UNA VIVIENDA**", ha considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación del Proyecto.

Por lo antes expuesto, se autoriza los empastados correspondientes, según la normativa institucional.

Latacunga, febrero 2023

Para constancia firma:

……	……
Lector 1 (presidente)	**Lector 2**
Ing. Ms.C. Luigi Orlando Freire Martínez	**Ing. Ms.C. Carlos Francisco Pacheco**
C.I: 050252958-9	**C.I: 050307290-2**

……

Lector 3

Ing. Ms.C. Verónica Paulina Freire Andrade

CI: 050205622-9

AGRADECIMIENTO

Quiero agradecer primeramente a Dios por guiarme y darme sabiduría para poder alcanzar la meta trazada y a mis familiares que son mi pilar fundamental para sobresalir después de estos últimos años que han sido muy complicados

Agradezco a mi familia y a mi madre en especial por darme la facilidad y poder cumplir mi sueño, y las enseñanzas y consejos para poder tomar una decisión.

También aprovecho la ocasión para agradecer a mi pareja por estar ahí en los buenos y malos momentos, ya que sin su apoyo emocional no hubiera logrado alcanzar mis objetivos, quiero agradecer a mis profesores por haberme impartido las enseñanzas y anécdotas laborales siempre enfocados para que seamos unos excelentes profesionales y que tengamos conocimiento adecuados.

Cristian Angulo

DEDICATORIA

El presente trabajo quiero dedicarlo a mi madre, ya que siempre me ha apoyado en todos los proyectos que he querido realizar, tanto para mi vida de formación profesional y de emprendimiento para mi sostenibilidad económica ya que muchas veces quería rendirme, pero en ese momento estuvo ella con su sustento incondicional.

También quiero dedicarles a mis familiares que ellos me brindaron apoyo sentimental y me alimentaban a lograr cumplir mi meta.

Cristian Angulo

UNIVERSIDAD TÉCNICA DE COTOPAXI FACULTAD DE CIENCIAS DE LA INGENIERIA Y APLICADAS

TÍTULO: "DESARROLLO DE UN SISTEMA DE CONTROL DE ÁNGULOS DE CÁMARAS IP MEDIANTE IOT PARA EL MONITOREO EN TIEMPO REAL DE UNA VIVIENDA"

Autor:

Angulo Chiliquinga Cristian Omar

RESUMEN

El presente trabajo de titulación realiza la implementación de un sistema de cámaras de video vigilancia controlado por aplicación de mensajería Telegram, dicho prototipo permite observar imágenes y videos mediante el uso de una botonera existente en el grupo de barrio de la aplicación, así mismo, controlar la posición de la cámara en el sentido vertical y horizontal lo cual permitirá a los usuarios monitorear varios puntos focales con el uso de una misma cámara. Las imágenes y videos receptados se almacenan en el grupo el cual tiene una capacidad máxima de almacenamiento de 50 GB y el límite de usuarios de 250 000 personas. Para esto se utiliza un microcontrolador ESP32CAM que posee una resolución de 2 MP el cual es comparado mediante el uso de la API de Telegram y la herramienta Pi Tunnel de Raspberry Pi 3, al analizar sus ventajas y desventajas se determinó que la mejor relación costo – beneficio la da la tarjeta electrónica con la API y su CHATBOT. Finalmente, tanto los servomotores que permiten el movimiento de los ejes y la tarjeta electrónica que hace las veces de cámara y microcontrolador se albergan en un Case de Fibra de Carbono realizado a partir del diseño generado en el software Fusión 360 y su impresión 3D haciendo uso de una máquina de la empresa Creality Ender 3 S1 PRO. El proyecto tiene una confiabilidad del 98% en el control de ángulos y un 100% en la conectividad, para su funcionamiento debe estar conectado a una red WiFi.

Palabras clave: Sistema, seguridad, control, ESP32CAM, IOT, PAN, TILT, Telegram.

ABSTRACT

This degree project implements a video surveillance camera system controlled by Telegram messaging application, this prototype allows to observe images and videos through the use of an existing button panel in the neighborhood group of the application, as well as to control the position of the camera vertically and horizontally which will allow users to monitor several focal points with the use of the same camera. The images and videos received are stored in the group which has a maximum storage capacity of 50 GB and a user limit of 250,000 people. For this, an ESP32CAM microcontroller with a resolution of 2 MP is used, which is compared by using the Telegram API and the Pi Tunnel tool of Raspberry Pi 3. After analyzing their advantages and disadvantages, it was determined that the best cost-benefit ratio is given by the electronic card with the API and its CHATBOT. Finally, both the servomotors that allow the movement of the axes and the electronic card that serves as camera and microcontroller are housed in a Carbon Fiber Case made from the design generated in the Fusion 360 software and 3D printing using a machine of the company Creality Ender 3 S1 PRO. The project has a reliability of 98% in angle control and 100% in connectivity, for its operation it must be connected to a WiFi network.

Keywords: System, security, control, ESP32CAM, IOT, PAN, TILT, Telegram.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INFORMACIÓN BÁSICA

Título:

Desarrollo de un sistema de control de ángulos de cámaras IP mediante IOT para el monitoreo en tiempo real de una vivienda.

Fecha de inicio:

Octubre 2022

Fecha de finalización:

Marzo 2023

Lugar de ejecución:

Provincia de Cotopaxi, Cantón Latacunga.

Facultad que auspicia:

Ciencias de la Ingeniería y Aplicadas

Carrera que auspicia:

Ingeniería Electromecánica

Tutor:

Nombres: Byron Paúl

Apellidos: Corrales Bastidas

Cédula de identidad: 050234776-8

Correo electrónico: byron.corrales@utc.edu.ec

Línea de investigación:

Procesos Industriales

Sub Líneas de investigación:

Línea 2: Automatización, control y protecciones de sistemas electromecánicos

Proyecto de Investigación: Aplicación de tecnologías electrónicas y de comunicación para la seguridad barrial.

1. INTRODUCCIÓN

Hoy en día la inseguridad está elevando las cifras, y la ciudadanía está viviendo estos índices de delincuencia, dentro de las incertidumbres está el robo a residencias o viviendas que ocupa un 31.1% de casos registrados en nuestro país, la ciudad de Latacunga las estadísticas incrementan al pasar de los años, en el año 2022 hubo 144 casos registrados de robos a domicilios según reporta la FISCALÍA GENERAL DEL ESTADO.

1.1 EL PROBLEMA

1.1.1 Situación de la problemática

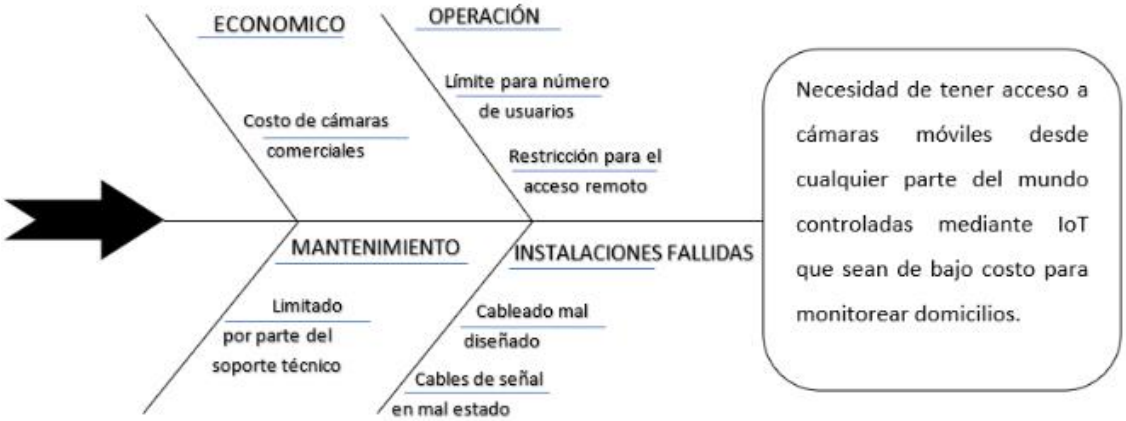
El entorno social en el Ecuador es crítico debido a varios factores, políticos y familiares, las pocas oportunidades de empleo ha conllevado a la migración de familias y el desempleo es factor principal para el incremento de la delincuencia y direcciona a la inseguridad, la seguridad revela más defectos para impedir dichos actos; a más del inconveniente del alto costo para la adquisición de kits de cámaras de video vigilancia y el confuso uso de cables para la conexión y el restringido acceso remoto para el monitoreo de las cámaras, ya que al usar un DVR dificulta el monitoreo cuando están fuera del lugar de instalación. Por ejemplo, las cámaras de uso comunitario solo lo pueden controlar y monitorear desde la base. Además, otro problema es el que cuando necesitan el acceso a las cámaras tienen que hacer un pedido para obtener las grabaciones y sin esperar buenos resultados, al estar dependiendo de una central y esperar la voluntad de los operarios, es decir, que tienen que ir al lugar específico para adquirir las grabaciones, se pierde tiempo hasta llegar al lugar.

1.1.2 Formulación del problema

Necesidad de tener acceso a cámaras móviles desde cualquier parte del mundo controladas mediante IoT que sean de bajo costo para monitorear domicilios.

Diagrama de Causa-Efecto

A continuación, la siguiente figura que representa el diagrama de Ishikawa el sistema de cámaras comerciales.



1.2 BENEFICIARIOS

Directos: Estudiantes de la Facultad de Ciencias de la Ingeniería y Aplicada de la Universidad Técnica de Cotopaxi y moradores de los diferentes barrios.

Indirectos: Comunidad Científica

1.3 JUSTIFICACIÓN

El presente trabajo de titulación forma parte del Proyecto de Investigación Formativa de la Carrera de Ingeniería Electromecánica denominado "Aplicación de tecnologías electrónicas y de comunicación para la seguridad barrial" en donde mediante el desarrollo de un sistema de control de ángulos de cámaras IP mediante IoT para el monitoreo en tiempo real de una vivienda se busca disponer de dispositivos que ayuden a combatir la inseguridad de la ciudad, para ello se pretende emplear cámaras IP las mismas que se contactarán a un microcontrolador ESP32 en unión con la aplicación de mensajería Telegram en donde se ejecutarán comandos para el control de los movimientos de los servomotores; además, se dispone de comandos mediante los cuales el sistema envía imágenes y videos cuando el usuario lo solicite a través. Para el funcionamiento de este dispositivo es necesario la creación de grupos de mensajería al cual todos los usuarios que pertenezcan tendrán la posibilidad de monitorear en tiempo real y además recibirán las respectivas notificaciones y videos imágenes cuando sean solicitadas.

1.4 HIPÓTESIS

Mediante el empleo de la aplicación de mensajería Telegram se podrá monitorear en tiempo real una vivienda.

1.5 OBJETIVOS

1.5.1 GENERAL

Implementar un sistema de control de ángulos de cámaras IP mediante IOT para el monitoreo en tiempo real de una vivienda.

1.5.2 ESPECÍFICOS

- Investigar el estado del arte de referente a sistemas de comunicación y control de motores y cámaras IP mediante el microcontrolador ESP32CAM.
- Realizar el diseño del prototipo del sistema en hardware como software.
- Implementar el prototipo con diagramas de instalación.
- Validar el funcionamiento del prototipo en escenarios reales.

1.6 SISTEMAS DE TAREAS EN RELACIÓN A LOS OBJETIVOS

Tabla 1. Tabla de tareas por objetivo

Objetivos	Actividades	Resultados de la actividad	Técnicas, medios e instrumentos
Investigar el estado del arte de referente a sistemas de comunicación y control de motores y cámaras IP mediante el microcontrolador ESP32CAM.	Indagación bibliográfica sobre la comunicación IOT. Adquirir datos para el control de servomotores.	Artículos referidos a la comunicación IOT.	Documentos de sitios web, papers, libros, videos.
Realizar el diseño del prototipo del sistema tanto en hardware como software.	Programación del código que se usara en la tarjeta ESP32CAM. Codificación de líneas de programación	Comunicación de la tarjeta ESP32CAM con Telegram	Software Arduino IDE de programación.

Implementar el prototipo con diagramas de instalación.	Diseñar el circuito con elementos necesarios para la instalación. Efectuar el diagrama de instalación.	Modelamiento del circuito.	Tarjeta ESP32CAM
Validar el correcto funcionamiento del prototipo en escenarios reales.	Verificación de conexión a la red. Aprobar del funcionamiento del prototipo.	Comparación de resultados obtenidos en las pruebas	Observación del correcto funcionamiento. Obtención de datos.

2. FUNDAMENTACIÓN TEÓRICA

2.1 ANTECEDENTES

Carlos Binker estudiante de la Universidad Nacional de La Matanza de Buenos Aires, Argentina nos comenta que hizo un trabajo que esboza como eje principal agenciar un hardware IOT confusamente mediante mensajes de texto instantáneos, incluyendo a la aplicación Telegram accediendo definir bots (aféresis de robots) y gestionar a través de un token que la misma aplicación crea en forma expuesta a través de su bot importante, denominado *BotFather*. Telegram ofrece una API, que admite que los *bots* interactúen con nuestro procedimiento. El *bot* estará incluido en la programación del chip ESP32 es un SOC que incorpora la red WiFi, sensores, conversores AD y DA. A manera de caso para estudio se plantea controlar mediante un bot a un conjunto de leds y a un sensor; este último elemento será simulado mediante un pulsador. Sera obligatorio para ello el empleo de complicaciones. Además, se solicitará el uso de Telegram (versión web o móvil) y el ambiente de simbolización de líneas de programación en Arduino IDE. Las instrucciones del bot se activarán distinguiendo el símbolo" /", o bien estableciendo teclas para cada comando[2].

Raynel Moreno Hernández estudiante de la universidad de valencia dice que la idea es que, manipulando varios sensores acoplados a internet mediante la conectividad de red wifi-doméstica, logremos ejecutar labores desde nuestros dispositivos móviles sin importar que no estemos en el lugar como: descubrimiento de fugas de gas, advertir de

incendios, sistematización del regadío de agua en plantas ornamentales o disponer datos de temperatura y humedad del interior de nuestras viviendas. Todo esto registrado por un chip muy económico, pero con muchas opciones de ayudas como es el ESP32. Como costumbre de declaración entre los terminales y el servidor manipularemos el MQTT que ha ganado gran renombre por su máxima utilidad y programación abierta y mínimo coste, mediante una interfaz gráfica muy llana programación para utilizar la lógica del proyecto[3].

Yosy Rahmawati comenta que el Elaboro un procedimiento de aviso para los desobedientes del paso de cebra en las luces rojas manipulando el microcontrolador ESP32-CAM, manejando el sensor ultrasónico para divisar carros. Cuando el sensor diviso que el automóvil sobrepasa la línea de cruce de cebra detallada y la luz del sol es roja, el ruido mostrará a los conductores que infringen la línea de cruce de cebra que retrocedan a la fila oportunamente. A continuación, el método encarcelará la transgresión y el trofeo de biombo se reemitirá al telegrama como ensayo de la infracción. Las derivaciones de este estudio muestran que los sensores ultrasónicos pueden funcionar si los vehículos cruzan la línea límite de 65,6 cm cuando el disco está en rojo. Cuando ocurre una infracción, suena un ruido sonoro que indica que se ha originado una infracción y la cámara ESP32-CAM tomará captura de la pantalla y la remitirá a un telegrama como ensayo de la infracción[4].

2.2 MARCO REFENCIAL

2.2.1 Sistemas de seguridad

Un procedimiento de seguridad es un proceso vinculado con elementos interconectados instalados para prevenir, detectar o responder a varios tipos de amenazas.

a. Física

Resulta que la seguridad física es una composición de muchos componentes, es parte de una orientación de seguridad para avalar la seguridad de un lugar elegido para prevenir amenazas. Para suministrar buenos servicios de seguridad debe estar determinado las posibles amenazas y riesgos para el lugar y los compendios precisos para ello es preciso facilitar una excelente protección con elementos electrónicos que consiguen ayudar a advertir los riesgos y reducir las amenazas físicas, robo, secuestro, y muchos más factores que implican a la inseguridad[5].

b. Electrónica

La definición más frecuente y elemental, un procedimiento de seguridad electrónica es tratar todas las labores de control de vigilancia que se consiguen para ejecutar con materiales de ultima tecnologías como dispositivo de video de seguridad enlazadas con alarmas o sirenas[6].

La tecnología enlazada con cámaras de seguridad electrónica está compuesta de diversos compendios relacionados con punto de conexión móviles y electrónicos. Estos instrumentales admiten un control más seguro al instante de monitorear y controlar los artículos de seguridad con pronto aviso para su esquema automático. Un método de seguridad representa que cualquier punto de conexión de equipos electrónicos inteligente consigue efectuar sistematizaciones de seguridad como monitoreo, vigilancia de acceso para las diferentes interfaces, barreras de programación en aplicaciones de tipo comercial y circuitos cerrados de televisión que los conocemos a los sistemas de cámaras de video vigilancia [7].

Figura 2.1 Relación gráfica entre los componentes de seguridad

2.2.2 Clasificación de los sistemas de seguridad electrónica

Hay diversas escrituras de simbolización, pero las dos primordiales son: alcance o tamaño y según su aplicación.

a. Tipos de alcance o tamaño

A partir del criterio escrito, los procedimientos de altos índices de seguridad electrónica por ausencia por parte de su alcance o tamaño en: métodos locales y distribución como se ve en la figura 2.1. Las técnicas de seguridad electrónicos para fines domiciliarios son aquellos colocados en una posición específica, a modo de ejemplo usaremos de una vivienda. Esta es la representación más elemental en que se puede hacerla practica y destella las sutilezas imperceptibles explicada anteriormente[8]. Asimismo, estos métodos refieren con instrucciones de declaración para enlazar entre sí. Para la aplicación se da encima de todo criterio para sociedades que frecuenten el sistema de vigilar toda su cobertura de lugar de trabajo. Por lo tanto, a un tiempo prolongado para su calidad se manipulan equipos de alta gama para este propósito[8].

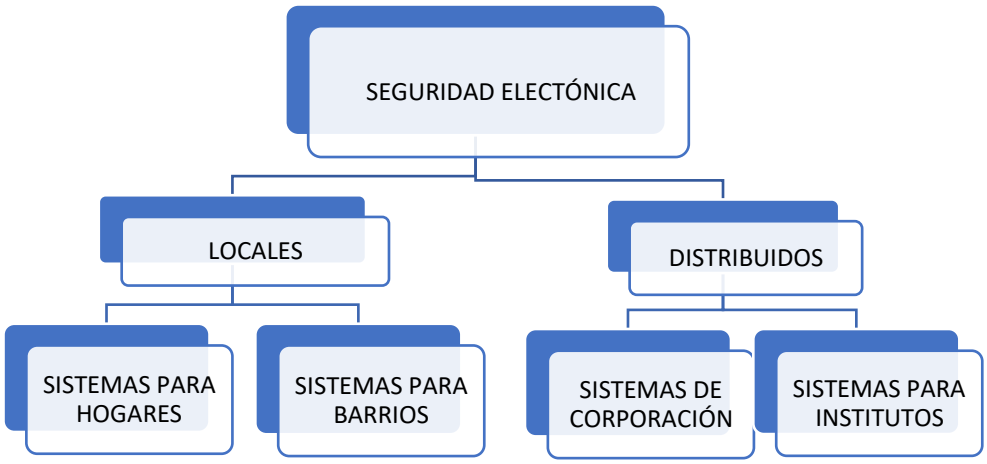


Figura 2.2 Seguridad electrónica[8].

Sistemas de seguridad

b. Su aplicación

Por esta razón, los procedimientos de seguridad electrónica se catalogan por aplicación y uso. Se consigue clasificar en cuatro conjuntos primordiales: hurto, sustracción, inflamaciones de sustancias explosivas , anti hurto y procedimientos especiales como se observa la codificación de la figura 2.2.

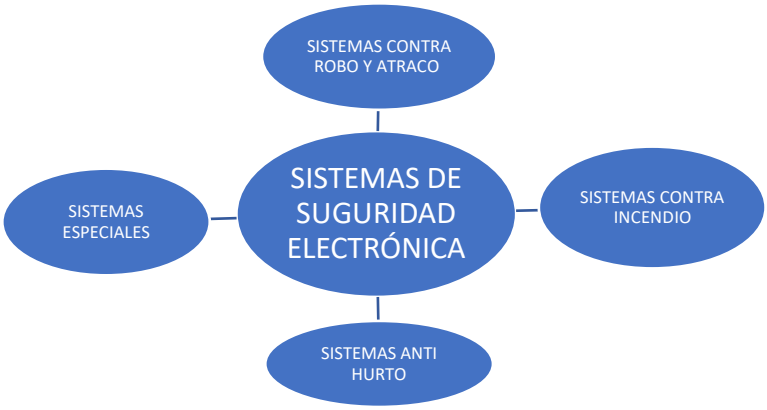


Figura 2.3 Aplicación de sistemas electrónicos[8].

Sistemas contra hurtos y sustracciones.

Métodos de seguridad hacia hurtos y saqueos domina ninguno muestras de técnicas de seguridad. Los más notorias son los procedimientos de susto, aparatos de video vigilancia, vallas eléctricas, intervenciones de acceso y botones de socorro[8].

2.2.3 Tipos de los sistemas de seguridad

- Aparatos de video cuidado, cámaras de seguridad o CCTV: asimismo conocido como circuito cerrado de transmisión, consiente el control y monitoreo centralmente de un lugar o áreas abundantes.
- Régimen de alarma: Es un procedimiento con diversos elementos posibles de comunicar eventos inusitados que acurren chico seguras situaciones.
- Técnicas de control de acceso: Son numerosos técnicas electrónicas que permiten o ciñen el acceso a lugares concretos o dedaleras mediante señales o datos biométricos.
- Posicionamiento GPS: Es una técnica de cuidado que consiente a las personas estar al tanto el sitio puntual de los objetos.

2.2.4 Dispositivos de seguridad electrónica

Los aparatos de un sistema de seguridad travesean desiguales roles asociarse sí. Los acontecimientos manifestados por el procedimiento de sobresaltos proveído se consiguen diversificar para potestad y dificulta el hurto moderadas de seguridad, por ejemplo, un robo, consigue llegar propio de seguridad apto posiblemente para avisar el percance y si la sazón amerita que se informe a las jurisdicciones específicos, en caso de quemas, los sucesos se estar al tanto en tiempo real. Este utensilio es seguidamente[9].

Los componentes del sistema de seguridad se describen a continuación.

2.2.5 Base de central de operaciones.

Es un aparato para descubrir señales eléctricas de sensores. Posteriormente de recoger estas indicaciones, su término interno marca y envía la señal al eje receptor. También recoge comandos a partir la interfaz de usuario para tramitar. Es la razón de la memoria del sistema, por lo que se encomienda colocar e instalar en un lugar oculto, alejado del medio ambiente o exposición química.

a. Sensores

Los sensores son terminales que reconocen lo que sucede ene le espacio protegida y luego envían esta indagación a un terminal central con caracteres eléctricas. La gama de tal punto de vínculo es exageradamente amplia, ya que existen delineados para una gran suma de estudios. Los característicos de movimiento, reveladores de humo, cámaras de videovigilancia, estos son los más utilizados.

b. Interfaz

Una Interfaz de usuario es un componente que permite al usuario controlar y operar una maquina o sistema. Por tanto, este es el elemento que se utiliza para gestionar el sistema de seguridad. En este caso se debe hablar de conectar y desconectar de manera local o remotamente. Incluye llaves, botones, teclados, computadoras y ahora teléfonos inteligentes.

c. Conexión de red.

Una red interrelacionada es el elemento que debe enlazar todos los dispositivos que constituyen parte de la red del régimen de seguridad interior. Puede ser alámbrico, inalámbrico o hibrido, siendo este moderno se refiere a una composición de los dos[10].

d. Fuente de energía

Una fuente de manutención es el principio de energía eléctrica necesaria para el funcionamiento del régimen, a menudo es obligatorio reducir el voltaje de su red doméstica de 220 ACV a 12DCV. Este es el voltaje al que operan algunos sistemas electrónicos de seguridad.

e. Respaldo de suministro de energía

Se solicita energía de reverso en caso de una incisión de energía y es seguro la maniobra continua del sistema. En el caso de que coexista un régimen de alarma, en aquel momento es Batería de Gel o Agm. En el asunto de los procedimientos de videovigilancia, esto se consigue formar con un inversor[10].

2.2.6 Tipos de conexión

Los espantos son los resúmenes que se impulsan cuando la técnica genera una señal. Por lo ordinario la sirena de alta energía o una luz fuerte[10].

a. Red de conexión.

La red es la tecnología que consiente que las computadoras se notifiquen entre sí. Eso simboliza que alcanzan a expedir y absorber datos entre ellos. Gracias a las redes de internet, casi todo es posible. Estamos acostumbrados con los conocimientos existentes, el uso de redes sociales y entorno de trabajos[11].

b. Tipo de Conexión

Un enlace está diseñado para ser una vía que lleva de un lugar a otro en el campo. La automatización existe en un medio asociado con una tribuna particular, señalo esto porque las redes de hoy tienen desarrollado exponencialmente. Puedes topar todo tipo de servicios que condescienden diferentes conexiones que me consiente acceder a las plataformas[12].

Se permiten varias uniones mediante la transmisión de datos de un punto de conexión a otro, el interés de datos vivir a cuenta de del tonelaje de la máquina, ya que puede ser urinario la transferencia de datos estribando del monitor para que pueda alcanzar más recursos para consentir los comandos que ejecuta en la línea de red.

- Red digital RDSI

Esto contenía líneas telefónicas manuales, uno de los tipos de conectividad que sustituyó a la RTC. Tiene una cartulina de red que efectúa las mismas labores que un módem a base de enviar datos desde tu computadora a la red, sin embargo, el uso de varios cables para conectarse a la red y hablar por teléfono requiere cierta infraestructura.

- Acceso a la red satelital

Debido a que le permitan el acceso a la red a través de conexiones satelitales utiliza señales electromagnéticas, a menudo se utiliza para acceder a redes en áreas remotas o donde la fibra óptica o ADSL no están disponibles, como en aviones o barcos. Necesitamos estas ondas establecidas por satélites.

La ventaja de este tipo de conexión es que no se requieren cambios de infraestructura para acceder a Internet. También se puede utilizar en lugares donde no hay líneas telefónicas. Sin embargo, se requiere una antena para el intercambio de datos. Se necesita una tarjeta de recepción para su computadora, pero la demora en la descarga sigue siendo excelente de la internet.

- RTC

Uno de los primeros tipos de conexiones o redes de internet fue la red telefónica de acceso telefónico, también conocido como la abreviatura RTC, esto es para la transmisión de datos a través de señales se incluyen los módems, dispositivos que conectan computadoras con acceso a la red.

- Conexión por cable

Algunos de los tipos de conexiones más comunes en la actualidad incluyen: cable de fibras ópticas para acelerar la transmisión de datos, según el método aplicado se puede utilizar conexiones a internet puras o de fibra óptica + coaxial más estable o quizás intermitente.

- RED ADSL

La red radica en una combinación de RDSI Y RTC, cada una con sus convenientes superioridades, proporcionar a los usuarios un trabajo perfecto y óptimo, El acrónimo ADSL es el breviario de Asymmetric Digital Subscriber Line o línea desigual de abonado digital en español, se define por ser un enlace traído en la agencia y en el hogar debido a su bella estabilidad, el camino a la red es una de las vicisitudes más traídas hoy en la invención.

- Red inalámbrica

Si tiene una red sin hilos como vereda a internet y necesariamente está restringido a cables, aun residir al tanto que Wireless es el traspaso de datos y encuesta y lo logro conseguir sin alianza de maromas, si las cadenas de luz matas de estación proceden de fibra óptica para imposibilitar detenciones bajo el dominio de integrantes externos, este procedimiento de vínculo más apropiados y traídos. Por un lado, tiene cánones de traspaso de procedencias muy resueltas en comparación con una unión más rápida en balance con una conexión más de 70 Mbps poseyendo una cobertura de 50km, también más mercantil que los tipos primeros no demanda infraestructura especial ni canjes en el cableado y el dispositivo existentes. Está trazado para perfeccionar la autenticación de redes inalámbricas[13].

2.2.7 Red inalámbrica

Una red celular es acople para red concertada para manejar señales de radio (ondas), una positiva espacio para la noticia entre otros terminales con camino de unión a la red y no uses cuerdas, estas son redes que esgrimen espacio de radio o infrarrojos para declaración de datos. En las redes celulares, no preexiste un vínculo física o cable entre los mediadores

al recibir averiguación, pero la red está pertinente por radio u ondas que también se las consigue crear con un microondas para almacenar la comunicación, elementalmente son tantos de dirección inalámbricos que se acoplan los dispositivos a la red[14].

2.2.8 Tecnología inalámbrica

Las redes de enlace inalámbrica se consiguen catalogar en cuatro conjuntos desiguales según el área de ambiciones y clases de señal: redes de área individual (WPAN), redes de población local o distinguido (WLAN), redes inalámbricas de área habitante (WMAN) y mallas inalámbricas de área extensa o públicas (WWAM)[15].

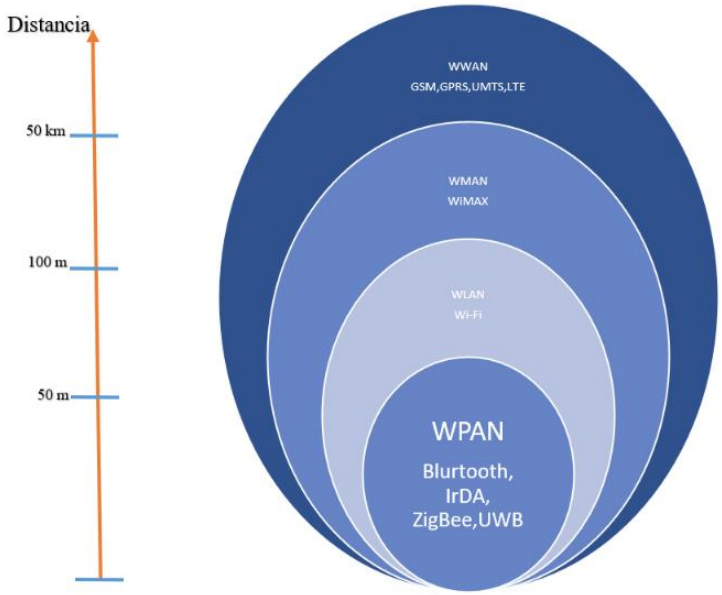


Figura 2.4 Clasificación y cobertura de redes Inalámbricas[15].

WPAN

Cuando se le menciona el termino WPAN, se entiende que está involucrada en la red celular propio o cercano, por el propio lanzo que golpea una derivación de corta recorrido de cubierta. Esta red se almacena para relacionar algunos dispositivos que se encuentran en las residencias, como smartphone, empresarias, timbres, ordenadores, procedimientos de sonido, entre otros. Todos ellos pertinentes de grafía inalámbrica para declaración multimedia, sin desdicha que escriben de algún tipo de cuerda para comunicar o transferir datos[16].

Los desiguales tipos del WPAN se encuentran:

- Bluetooth
- UWB
- IrDa
- ZigBee

WLAN

Es la red de mercado individual inalámbrica es a la que se enlaza más de un conector mediante una técnica de repartimiento inalámbrica, es un área específica, como colegios, moradas, campus de instituciones educativas, sucursales y empresas esta opción consiente a los agraciados desplazarse en única la cubierta particular para permanecer oportuno pese a la distancia. La localidad de las WLAN está enfocada en los tipos IEEE802.11 en las que se distribuyen bajo la marca de Wi-Fi[17].

WMAN

Es una red inalámbrica que se acopla a manera de una red que consiente a los compradores a vincular entre disímiles sitios dentro de un sitio metropolitano. De esta representación, resultan -favorecida para la compañía de noticia telegráfica[18].

En esta variedad de red evade el uso de gigantescos cables. Estrictamente se puede conectar de manera inalámbrica, lo que ahorra los costes de instalación y evade costes excusados. Este consiste archivar la condición más extensa que una WLAN usualmente utilizada[19].

WWAN

Esta red es inalámbrica extendida, con este tipo se solicita a la tecnología específica para ofrecer e involucrar un mejor servicio de telecomunicaciones de manera segura y eficiente[20]

Dentro de los WWAN los dos principales tipos son:

- Red de telefonía móvil.
- Satelital.

2.2.9 Red de área local inalámbrica (WLAN)

La WLAN está disponible con una gran variedad de oportunidades de conexión inalámbricas entres gran cantidad de usuarios que se pueden estar conectados en diversos dispositivos. El IEEE802.11 se encomienda de nivelar las distintas normas para las redes, como el ancho de banda que deben brindar los proveedores.

Elementos de la red WLAN

Los dispositivos básicos que admiten el correcto funcionamiento de la red WIFI.

Antena

Es el dispositivo que se encomienda de absorber y enviar ondas electromagnéticas.

NIC

Es la que posee la ocupación de dar el acceso y permitir a los beneficiarios que quieran enlazar entre ellos mediante WIFI entre los terminales más frecuentes que requieren este camino es el router, pc, impresoras, entre otros dispositivos.

Access Point

Con este módulo me consiente que los dispositivos admitan la interconexión de aparatos que obliguen la comunicación inalámbrica para existencia parte de una red inalámbrica en las que logren trasferir datos y poder establecer a través de una dirección IP.

Router

La principal función que posee es remitir datos afines entre una red y otra, o dando a concebir que una subred es un vinculado de aparatos que disponen de IP que se puedan informar sin la interpolación de un encaminador[21]

2.2.10 Topología de una red WLAN

Para emprender, se logra formular que una topología es la destreza de cómo está despachando la afirmación de red, percibiendo sus nódulos y figuras de unión a la red.

Se conceptualizan las 3 características que son:

Ad-hoc

Es un prototipo de red que se edifica posiblemente para consentir que se enlacen entre sí más de un punto de enlace inalámbrica sin que sea necesario un punto de conexión base, como un router, cuando las partes de vínculo a internet coexisten en modo ad-hoc, cada patrón de la red restituye datos a los de más[22].

Infraestructura

Esta escritura de organización de red inalámbrica que entra en un enrutador o punto de dirección WLAN. En este ejemplo, los modelos inalámbricos se notifican el uno con el otro a través de un enrutador[23].

Mesh

Conjuntamente, son registrados como red de malla es en donde se delimita una frecuencia base y desiguales puntos de acceso, más ilustres como nodos. Los distintos puntos de dirección no solo están acoplados al router, sino imparcialmente son aptos de enlazar

entre ellos, consintiendo a una excelente cubierta[24].

2.2.11 La tecnología WiFi

Una zona Wifi es concluyentemente una emboscada a Internet asistido con diferentes puntos de conexión en un hogar o compañía a través de un enrutador inalámbrico. El enrutador se corresponde claramente al módem de Internet y artimaña como un concentrador para trasladar la señal de internet a únicamente los usuarios que acomoden de conectores capacitados para soportar y receptar señales para Wifi, lo que le brinda la elasticidad de permanecer conectado a Internet mientras este se encuentre dentro de la eficacia de su red[24].

En liquidación, es una manera para enlazar redes y computadores y con la escasez de ocupar cables. WiFi se manipula como nombre genéricamente para productos que juntan cualquier conciliación de la tecnología inalámbrica IEEE802.11, que tolera en la creación de redes inalámbricas (Wireless LAN). Cuando referenciamos de WiFi, describimos a una de las tecnologías de declaración inalámbrica más común que se utiliza en la actualidad[25].

2.2.12 Estándar IEEE 802.11

Es la que está encargada en precisar el uso de menor nivel en la arquitectura OSI (capas físicas y de enlace de datos), detallando sus dos niveles de velocidades de transmisiones de megabits por segundo y sus funciones en una WLAN accediendo a los conmutadores y redes inalámbricas las que estiman los recursos de radiofrecuencia[26].

Tabla 2. Principales Características de los estándares

Estándar	**Velocidad máxima**	**Alcance**	**Frecuencia**	**Difusión**	**Característica**
802.11 a	54 Mb/s	390m	5GHz	OFDM	Posee un área de alcance menor al efectivo al momento de penetrar estructuras edilicias ya que opera en frecuencias superiores.

802.11 b	11 Mb/s	460m	2.4 GHz	HR/DS. HR/DSSS	Los dispositivos que implementan este estándar tienen un mayor alcance y pueden comprender mejor las estructuras edilicias que los dispositivos basados en 802.11
802.11 g	54 Mb/s	460m	2.4 GHz	ERP	Los dispositivos que efectúan este estándar operan en la misma radiofrecuencia y tienen un alcance de hasta 802.11b, pero con un ancho de banda de 802. 11ª.
802.11 n	600 Mb/s	820m	2.4 GHz o 5GHz		Las velocidades de datos típicas codiciadas van de 150 Mb/s a 6 0 0 Mb/s, con un alcance de hasta 70 m. Es compatible Con dispositivos

2.2.13 IoT

También distinguido como la internet de las cosas, es el que se encomienda de permitir conectar objetos diarios a la red. También, se ubica a las sistemáticas de postreros determinados que amontonan y transportan datos a través de redes inalámbricas sin cuantioso arbitraje transige, esto es factible reconocimientos a la mixtura de tanteos de enlace informáticos en todo tipo de objetos[27].

Los tipos de sitio de unión IoT logran ser acople del levantamiento para los favorecidos cuando no están verdaderamente presente, están entregados con sensores para que logren acumular datos que se logran ver, atender o concebir, Seguidamente notifican los datos tal como se personifican y los estar a la mira para que se logren avisar y suministrar acciones o solturas posteriores, Hay cuatro gestiones fundamentales en este proceso:

- Ventaja de caracterizaciones de datos reconocimientos a los sensores, los puntos de unión IoT absorben datos de su medio. Logra ser tan simple como la calentura o tan complejo como la transmisión de video en un lapso de tiempo real.

- El traspaso usual de datos: mediante los vínculos de red propicias y útiles, los posteriores de IoT permiten a enlazar a estas identificaciones a través de una nube pública o privada.
- Recogiendo los datos para sentenciar: en este punto, el software efectivo está proyectado para inventar algo señalado en estos datos, como prender un ventilador o despachar una alerta.
- Aceleración de datos: analiza los datos acopiados de todos los terminales en una red de IoT. Esto provee información activa para arrebatar arbitrajes y acciones fructuosos confiables[28].

a. **Arquitectura de IoT**

1. Sensores y actuadores: Estos cumplen la función de emisor-receptor de datos que están acoplados son los apoderados de inspeccionar o intervenir alguna cosa o cierto proceso físico.
2. Pasarela de internet y adquisición de datos: La ventaja de datos recoge la indagación en bruto de los sensores y los catequiza de forma analógico a la digital y a continuidad los envía por una planchada de internet ya sea inalámbrico o cableado.
3. Preprocesamiento: Los datos digitalizados necesitan ser condenados para reducir aún más acontecimientos cargos, como las de video prolijidad o espionaje remoto[29].

Figura 2.5 Chip ESP32-S.

El ejemplo ESP32-CAM posee algunas características técnicas muy encantadoras que logras ver en el datasheet del ejecutor, en la figura 5 se muestra la placa prefabricada del módulo ESP32 que tiene su sensor ov2640U.

Figura 2.6 Placa prefabricada ESP32-CAM.

2.2.14 Servomotor SG90

En los últimos tiempos el crecimiento de la apariencia de motores eléctricos empleados en la sistematización de técnicas mecánicas es indispensables y complejos, ha motivado el desarrollo de motores y controladores con mejores prestaciones que permitan cumplir con las amonestaciones de cada aplicación, en donde una de las miserias repetidos es el accionamiento análogo de motores y la revisión de su enfoque. El trabajo plantea el diseño y ejecución de un interventor de posicionamiento multieje de servomotores BLDC localizarse uso de las ciencias aplicadas embebida FPGA, el cual consiente el accionamiento y control de posición de compuestos motores de forma autónoma, pero que pueden ser accionados paralelamente y sin retardo acumulativo, para ello se circunscriben módulos emancipadas para el trueque electrónico de cada motor, así como módulos para la intervención de su posición. Los servos son una ramificación de los motores a CD, estos se identifican por su capacidad para posicionarse de letra inmediata y fiel dentro de su pausa de movimiento al estar manipulando. Para lo inicial, el servomotor aguarda un tren de pulsos; cada uno de estos pulsos tiene una permanencia concreta para mover el eje de beneficio del servomotor hacia una perspectiva angular determinada. Se expresa que el tren de pulsaciones es una señal constitucional; por lo que cuando ésta pasar de un extremo a otro en el amplio de sus pulsos, la perspectiva angular del eje además cambiar la casaca. Para intuir mejor la marcha de estos dispositivos electromecánicos, obsérvese la Figura 2.7 , en donde se consideran las señales que condescienden el pensamiento de un servomotor estándar.[30]

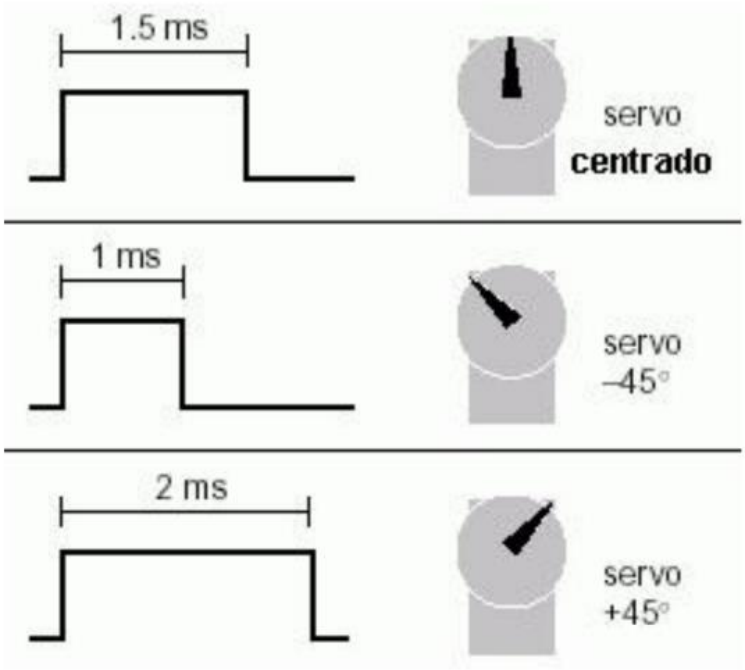


Figura 2.7 Movimiento de servo[30].

2.2.15 Impresión 3D

Según Jennifer Lawton nos cuenta sobre la producción digital que consiste en concretarse objetos a partir de registros digitales, utilizando para esto un artefacto intervenido por una computadora. Gracias a este dispositivo se logran varias ventajas, desde mejorar procesos de diseño, obtener fragmentos personificadas o disminuir costos de fabricación, hasta elaborar formas complicadas que no estarían dineros con las tecnologías acostumbrados[31].

a) Materiales:

El filamento de fibras de carbono se manipula en diversas manufacturas, ya que prometen trascendentales propiedades industriales y de ingeniería, como alta rigidez, firmeza a altas temperaturas, alto aguante química, peso ligero, alta obstinación a la tensión y baja distracción térmica. La fibra decarbono pura es en realidad cinco veces más fuerte que el acero y el doble de rígida, pero más liviana. Estas participaciones hacen que la fibra de carbono sea conveniente para diligencias en modelos y piezas que piden estos requerimientos de las propiedades del material para optimar su beneficio en uso. Esto es fundamentalmente cierto en campos como la ingeniería[32].

Figura 2.8 Impresora 3D

3. DESARROLLO DE LA PROPUESTA

3.1 METODOLOGÍA

El presente proyecto se basa en el diseño y fabricación de un prototipo de cámaras de videovigilancia para su aplicación desde la API de Telegram y la inclusión en alarmas comunitarias en el cual se desarrolló el código de programación para su funcionamiento y a su vez el hardware se utiliza la placa ESP32-CAM y una carcasa dibujada e impresa en 3D en material fibra de carbono. El prototipo está ideado para tener un control de visión de ángulos y con el funcionamiento de dos servomotores

Tabla 3. Elementos principales

Elementos	Fotografías
ESP32-CAM	
Adaptador 5V	
Cable USB	
Programador ESP	
Servomotor	

3.2 ELECCIÓN DE MICROCONTROLADOR

3.2.1 Microcontrolador ESP32-S

Es un circuito integrado que permite la conexión de cualquier chip integrado se pueda conectar a internet ya que tiene incluida una antena que recepta las ondas que emiten la red Wifi y es compatible con etiquetas TCP/IP, sumando al poder del sensor OV2640 permite hacer streaming de imágenes o videos y ponerlas en disposición del cliente y servirlas a la red de conexión. El módulo ESP32- CAM se le puede alimentar con un adaptador de 5V o 3V, lo recomendable para utilizarlo es con una fuente de 5v DC/1A, los pines de entrada y salida se los reconocen como (GPIO) a ellos se los alimentan a 3.3V por lo cual recomienda el fabricante utilizar conversores de nivel[33].

En la figura 6. Podemos identificar claramente los pines de alimentación y de señal.

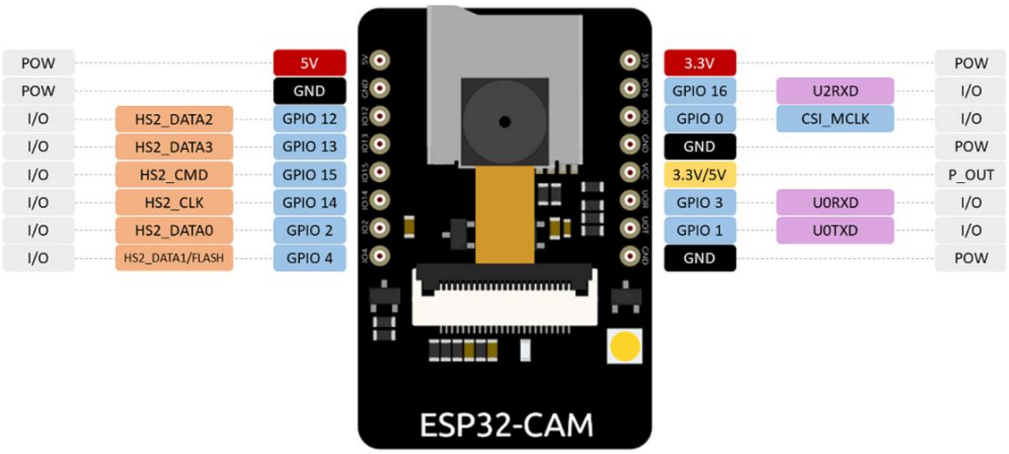


Figura3.1 Datasheet de la placa ESP32-CAM[29].

Las mejores características del ESP32-CAM se describe a continuación:

- Bajo coste
- Código abierto
- Programable
- Habilitado para Wifi
- Alimentación de 3V y 5V

3.3 PROGRAMACIÓN

Para este prototipo de microcontrolador se utiliza la interfaz de programación de Arduino IDE, en el cual tiene similitud con el lenguaje C, pero con características y librerías específicas para el modelo que estamos utilizando.

Para que la interfaz de Arduino IDE pueda reconocer al ESP32-CAM y a la base de identificaciones de datos “FIREBASE” se debe realizar el siguiente proceso de descarga e instalación previamente las librerías que la misma interfaz proporcione. Cuando ya se haya instalada las librerías que se requerían, daremos paso a la programación requerida por el mismo está en el Anexo 1 los pasos para instalar Arduino IDE.

3.3.1 Pasos para la programación

A continuación, observaremos el funcionamiento cual está representado como un diagrama de flujo que se lo puede observar a continuación.

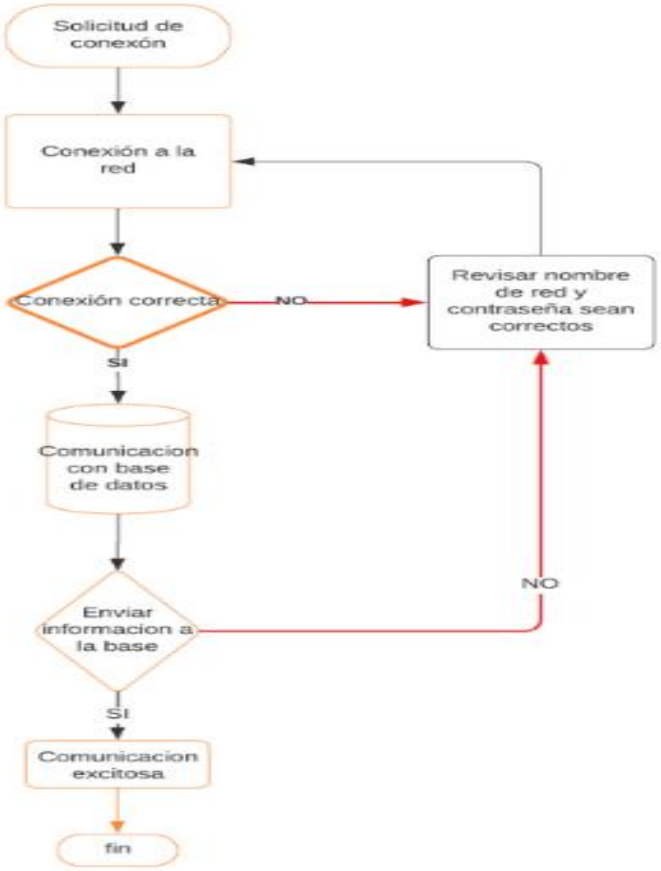


Figura 3.2 Diagrama de flujo del programa Arduino IDE

En el anexo IV esta las líneas de programación

3.4 DISEÑO DE CARCASA PARA ESP32-CAM Y SERVOMOTORES SG90

El boceto estructural de la carcasa, se la perpetró utilizando el software fusión 360 de Autodesk, elaborando un prototipo que proteja a la placa que esta conjuntamente con la cámara en la figura 14 se observa el Diseño3D.

Figura 3.3 Diseño 3D del case de la ESP32-CAM

En la 3.3 podemos observar el diseño y para poder imprimir, el filamento que mejor nos presta los beneficios es el de fibra de carbono i3D Tested de 1.75 mm, escogimos este tipo de material por las características de dureza y firmeza que nos brinda para nuestro prototipo.

3.5 SELECCIÓN DEL SISTEMA DE COMUNICACIÓN IOT

3.5.1 Raspberry Pi 3

Es una computadora de muy bajo costo que ejecuta Linux y tiene un conjunto de pines de entrada y salida que permite a los usuarios controlar componentes eléctricos.[34]

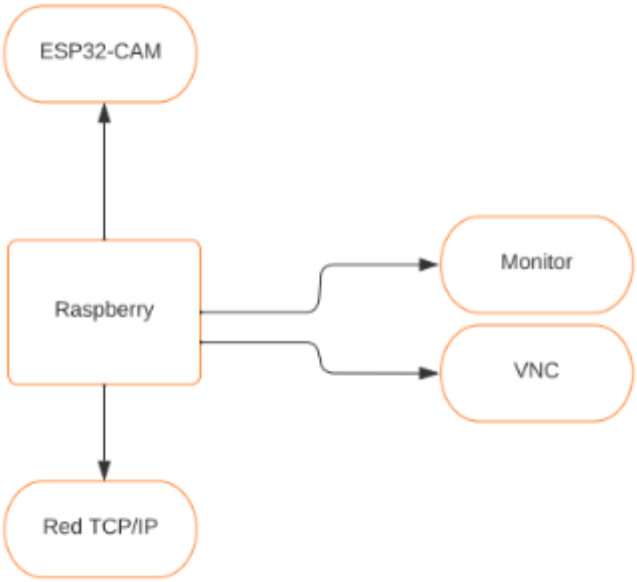


Figura 3.4 Procesamiento de señal

Se crea un túnel personalizado con la Raspberry Pi imager el cual me permitirá acceder a un punto de acceso a la red TCP en un dispositivo, al cual se puede ingresar a la base de datos desde cualquier parte del mundo sin importar la red de conexión.[35]

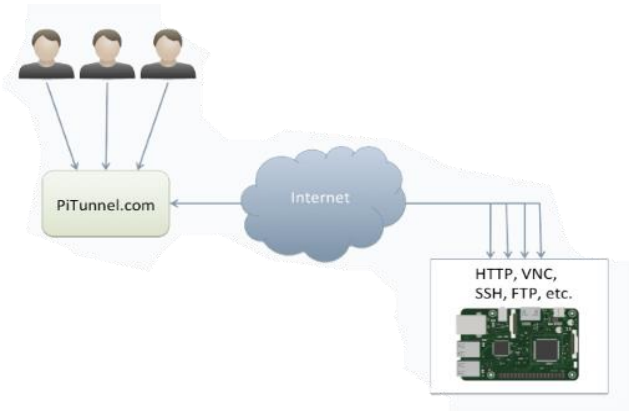


Figura 3.5 Funcionamiento del Pi tunnel.[35]

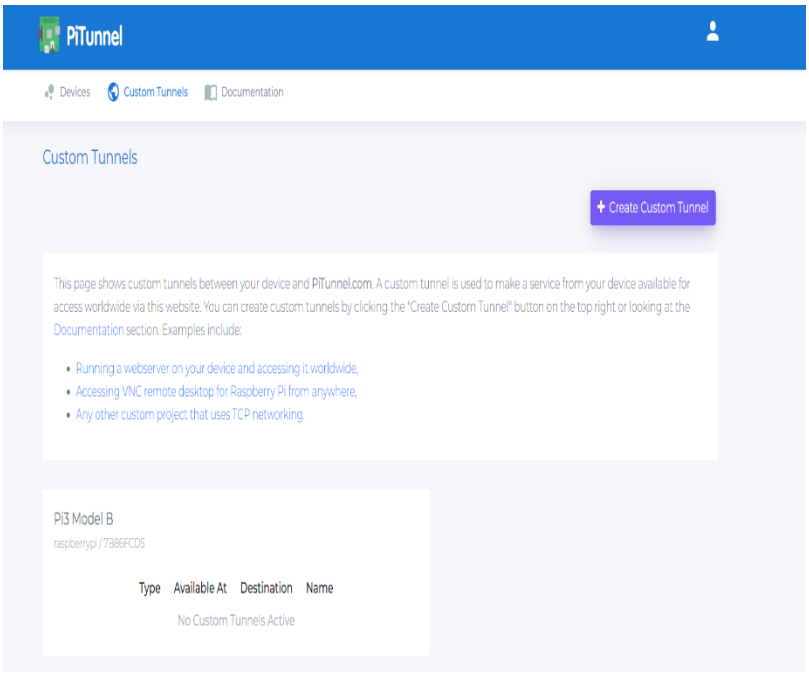


Figura 3.6 Creación de túnel en la página de Pitunnel

PiTunnel es el terminal remoto que permite el monitoreo de estado para Raspberry Pi, así como túneles sin ser necesario que este en la, misma red y poder utilizar desde cualquier servicio de red que se ejecute en su Raspberry Pi (como HTTP, VNC, SSH) para que pueda acceder desde cualquier parte mundo a través de Internet.

Figura 3.7 Página principal Raspberry Pi

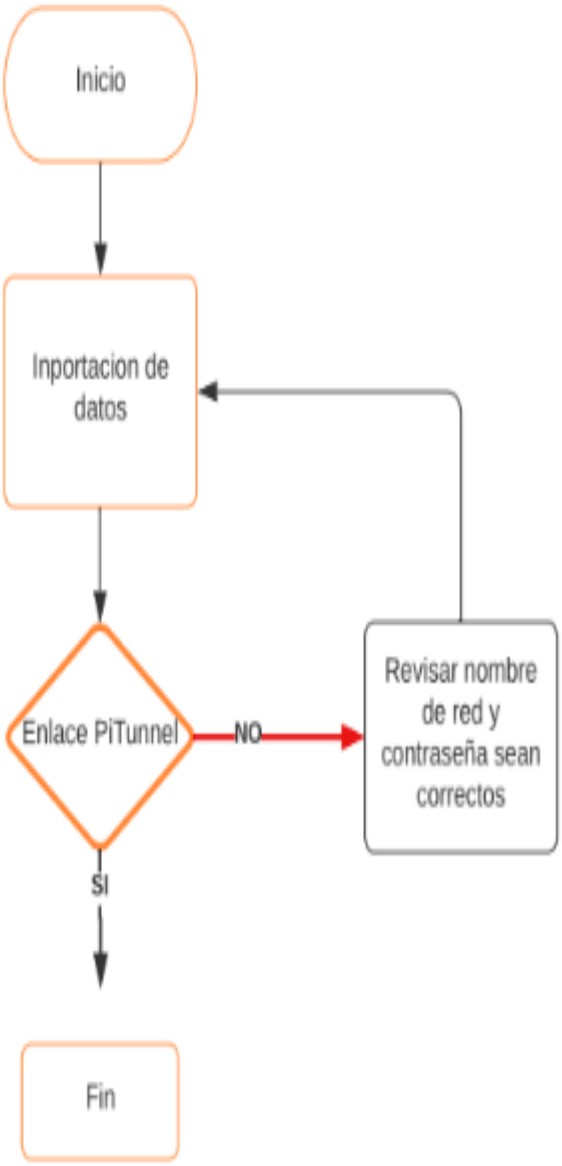


Figura 3.8 Diagrama de flujo de programación para Raspberry Pi

En el Anexo II está la guía para poder configurar

3.5.2 Telegram Messenger

Es un servicio de mensajería de texto y voz sobre Ip basado en la nube. Puede instalarlo fácilmente en su smartphone (Android y iPhone) o computador gratis y sin anuncios. Telegram permite crear bots con los que puedes interactuar[36].

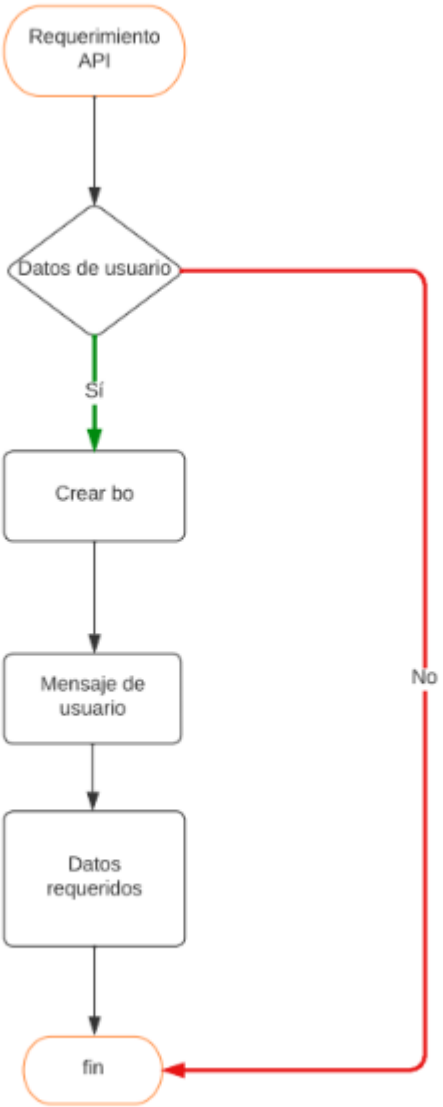


Figura 3.9 Diagrama de flujos de Telegram

Los bots son aplicaciones de terceros que se ejecutan dentro de Telegram. Los usuarios pueden interactuar con los bots enviándoles comandos tipo mensajes y solicitudes de conexión en línea. El ESP32-CAM interactuará con el Bot de Telegram para recibir y enviar respuestas[36].

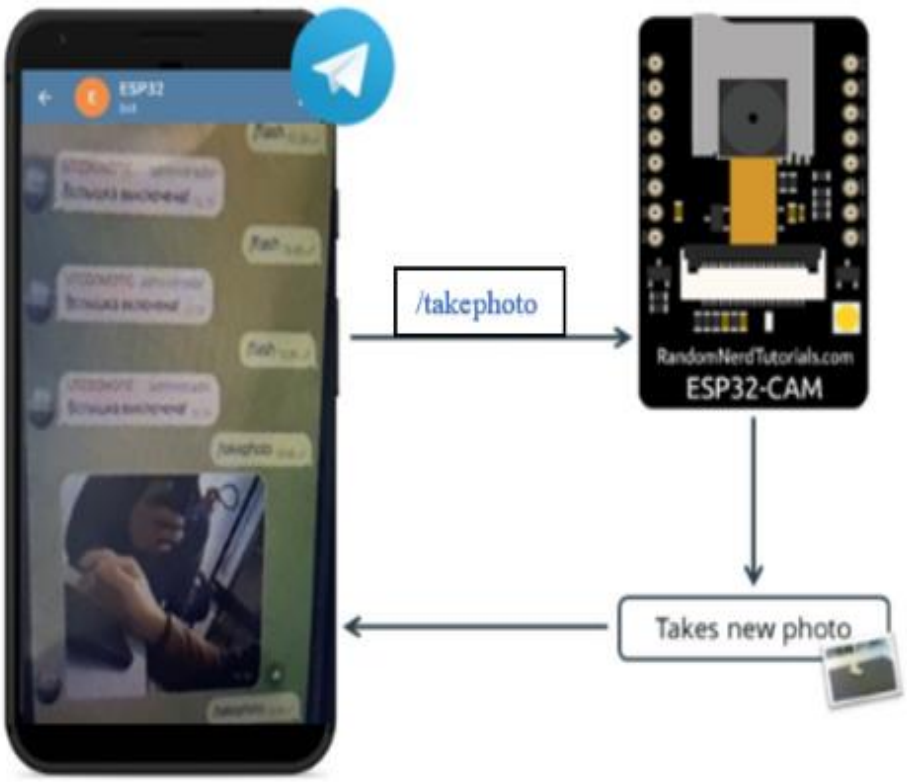


Figura 3.10 Descripción grafica del funcionamiento

Pasos:

- Crearás un bot de Telegram para tu ESP32-CAM;
- Puede iniciar una conversación con el bot ESP32-CAM;
- Cuando envía el mensaje /takephoto al bot ESP32-CAM, la placa ESP32-CAM recibe el mensaje, toma una nueva foto y responde con esa foto;
- Puede enviar el mensaje /flash para alternar el flash LED del ESP32-CAM;
- Puede enviar el mensaje /start para recibir un mensaje de bienvenida con los comandos para controlar el tablero;
- El ESP32-CAM solo responderá a los mensajes provenientes de su ID de cuenta de Telegram.

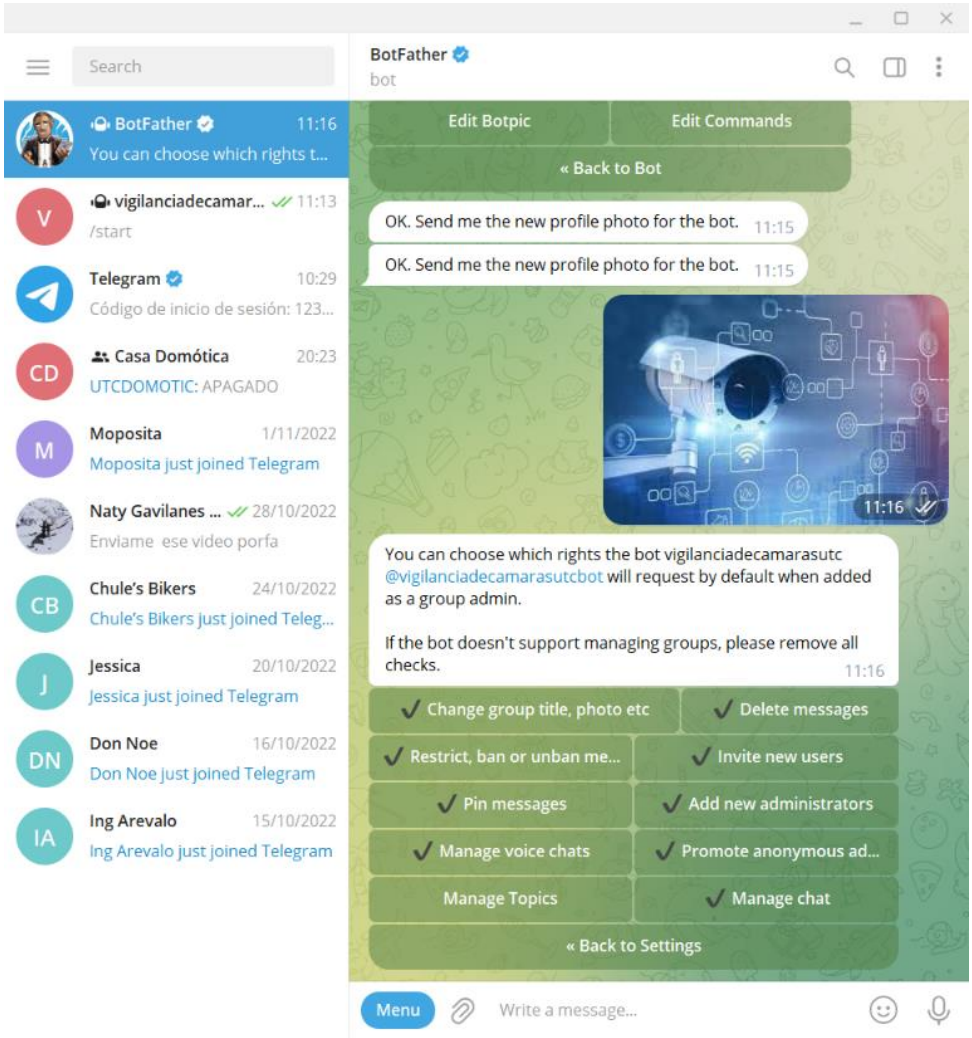


Figura 3.11 Creación del bot en Telegram

Biblioteca ArduinoJson

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

1. Vaya a Sketch > Incluir biblioteca > Administrar bibliotecas.
2. Busque "ArduinoJson".
3. Instale la biblioteca.

Copie el código siguiente del IDE de Arduino. Para que este boceto funcione, debe insertar sus credenciales de red (SSID y contraseña), el token de Telegram Bot y su ID de usuario de Telegram. Además, compruebe la asignación de pines para la placa de cámara que está utilizando.

Demostración

Cargue el código en la placa ESP32-CAM. No olvide ir a Herramientas > tablero y seleccionar el tablero que está utilizando. Vaya a Herramientas > Puerto y seleccione el

puerto COM al que está conectada la placa.

Después de cargar el código, pulse el botón RST integrado ESP32-CAM para que comience a ejecutar el código. Luego, puede abrir el Monitor serie para verificar lo que está sucediendo en segundo plano.

Ve a tu cuenta de Telegram y abre una conversación con tu bot. Envíe los siguientes comandos y vea cómo responde el bot:

/flash invierte el estado del flash LED;

/takephoto toma una nueva foto y la envía a tu cuenta de Telegram.

La guía a seguir está en el Anexo III.

3.6 CONTROL DE ÁNGULOS

El control de la posición de un servomotor tiene múltiples aplicaciones ya sea el sector industrial o para fines de prótesis. Por tal razón, surge la idea de los servomotores lo cuales cuentan con mecanismos o dispositivos electrónicos que les permiten conocer su posición en un instante dado la figura 3.12 esta evidenciado el servomotor en su posición inicial.

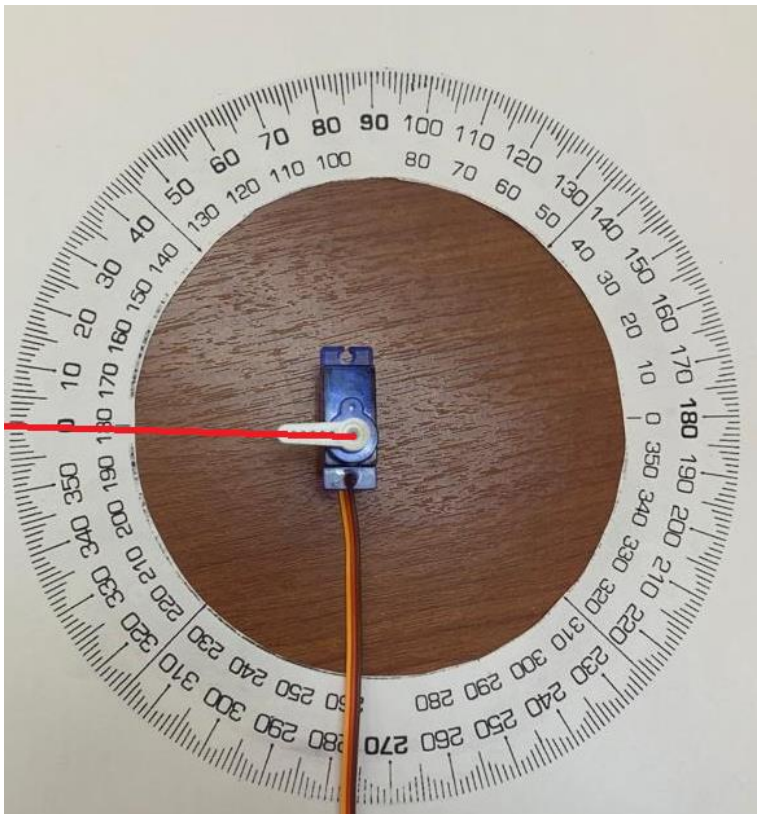


Figura 3.12 Servo motor en su estado inicial

La primera etapa de aceleración para realiza un cambio de punto en la interfaz hardware conformada por un microcontrolador ESP32-CAM que conecta al respectivo servomotor con el IO12 a través del puerto serie como esta en la figura 3.13.

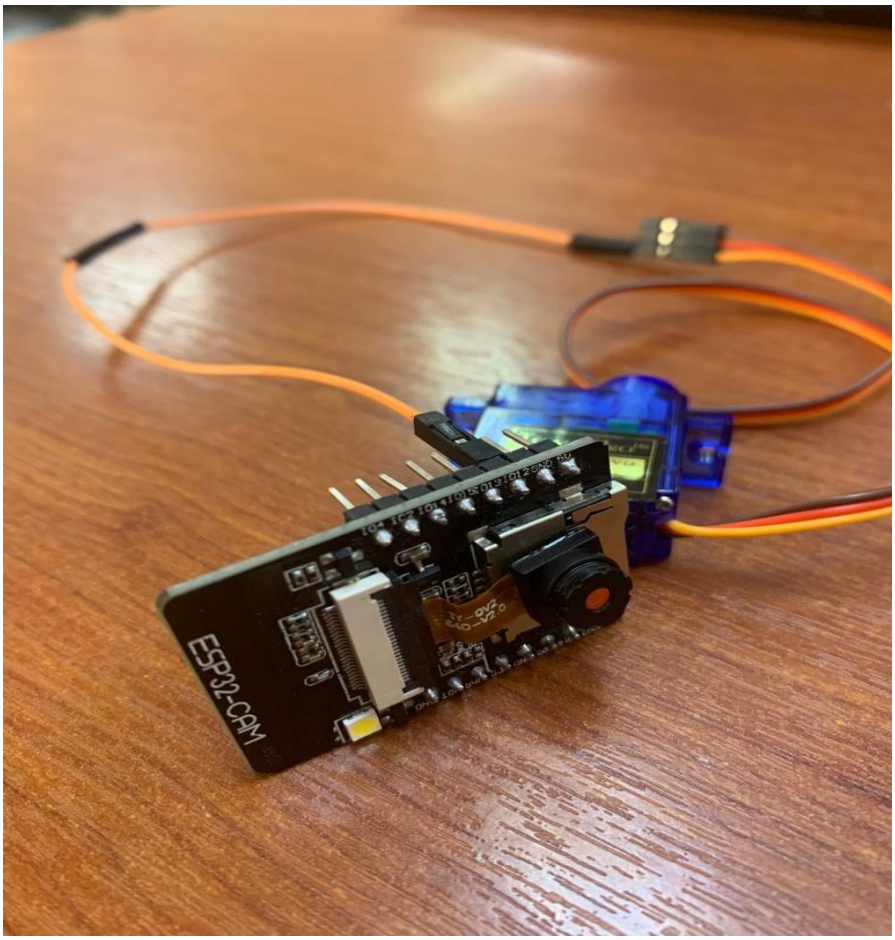


Figura 3.13 Conexión del servomotor SG90

La interfaz que admite la conexión de los módulos principales: un microcontrolador y un programa residente escrito en Arduino IDE que controla el puerto serie del ESP32-CAM para interactuar con el microcontrolador, este programa residente es el que permite controlar el angulo y cambiar las posiciones requeridas. Para el movimiento se requirió construir una base para los servomotores y el microcontrolador , interconectando las líneas de alimentación sobre el mismo conector de la placa, tal y como se aprecia en la Figura 3.13 y el montaje de la base se observa en la Figura 3.14.

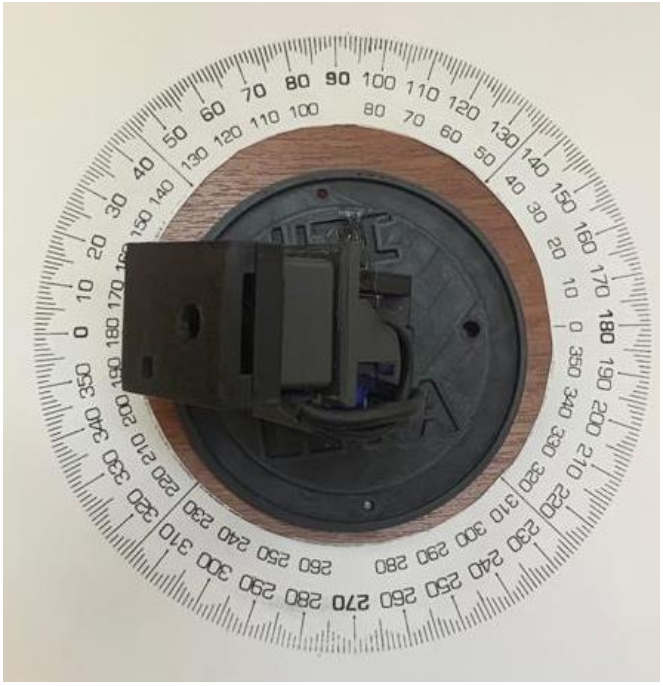


Figura 3.14 Ensamble de la base de los servomotores SG90

La figura 3.15 presenta los 210° de máxima rotación para ello se efectúa las prácticas en dichos motores .

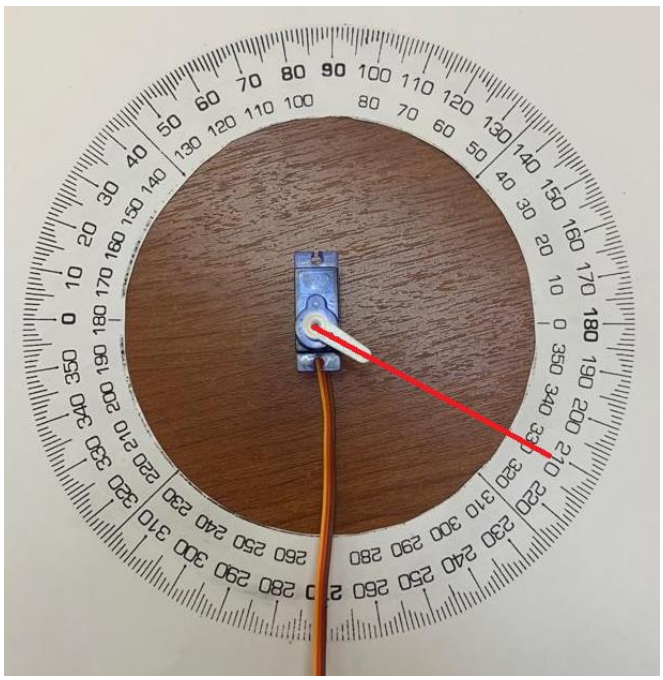


Figura 3.15 Punto máximo de ángulo de servomotorSG90.

3.7 COMPARACIONES ENTRE RASPBERRY PI3 Y TELEGRAM

Raspberry PI 3

Resultados del funcionamiento con Raspberry Pi 3

Figura 3.16 Punto estratégico para la instalación de la cámara

Acceso de monitoreo en tiempo real con la facilidad de poder ingresar desde cualquier parte del mundo a través del enlace emitido como ejemplar la figura 3.17.

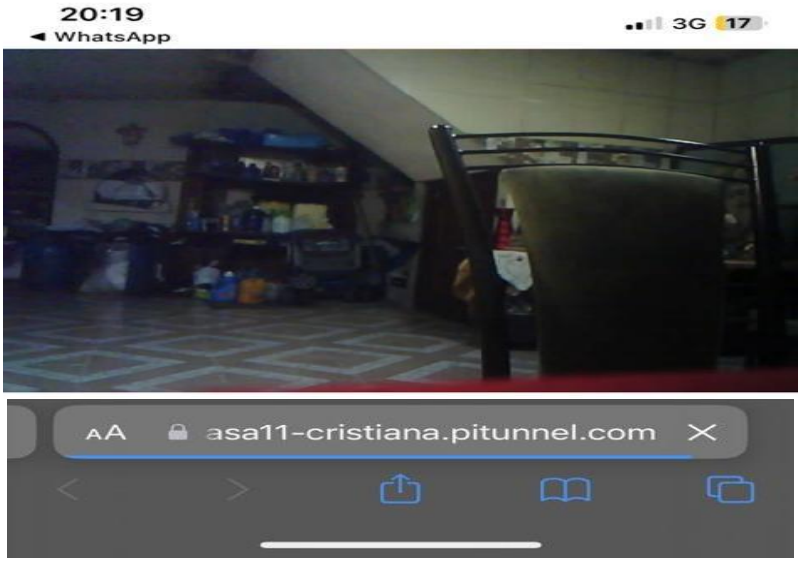


Figura 3.17 Enlace para acceder al monitoreo de las cámaras.

Los Botones son de fácil manipulación para el control de movimientos de los servomotores podemos controlar el ángulo deseado como se observa en la fifura 3.18, y en la figura 3.19 esta evidenciado los botones para iniciar la grabación o topar una foto de la transmisión del video.

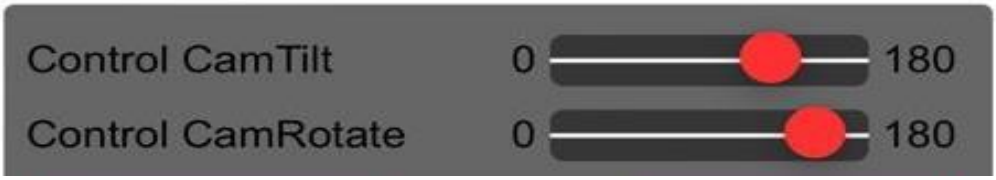


Figura 3.18 Botones para el control de movimientos de los servomotores

Figura 3.19 Botones para capturar la imagen y empezar la transmisión de video.

3.7.1 Manual de usurario para la aplicación de mensajería Telegram

El Comando /takephoto envía la foto del instante como se observa en la imagen emitida por un sensor ov2640 que es de 2 mega pixeles.

Figura 3.20 Comando /takephoto

Escribiendo el comando /flash enciende la linterna y de la misma manera puede apagar y el bot siempre dará la respuesta de encendido o apagado podemos observar en la figura 3.21.

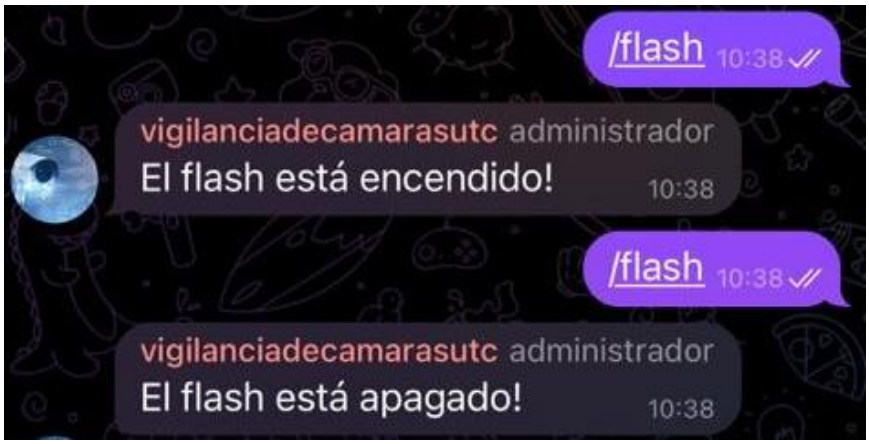


Figura 3.21 Comando /flash

Al aplicar el comando /recordvideo empieza a grabar desde el momento hasta tener 18 segundos de grabación y envía el video en formato en la figura 3.22 esta evidenciado el formato del video .avi lo recomendable para abrir este tipo de archivos es descargarse la aplicación *Movie player* en el caso de tener dispositivos con sistema operativo IOS y en el caso de Android se recomienda JVL para que lo puedan reproducir.

Figura 3.22 Comando /recordvideo

Para ajustar la resolución de la imagen hay que escribir /imagesettings como esta referenciado en la figura 3.23, y pedirá que confirme ingresando al enlace emitido por el bot, luego me dará unos botones para elegir la resolución para ellos recomendamos utilizar VGA que es compatible para nuestro censor.

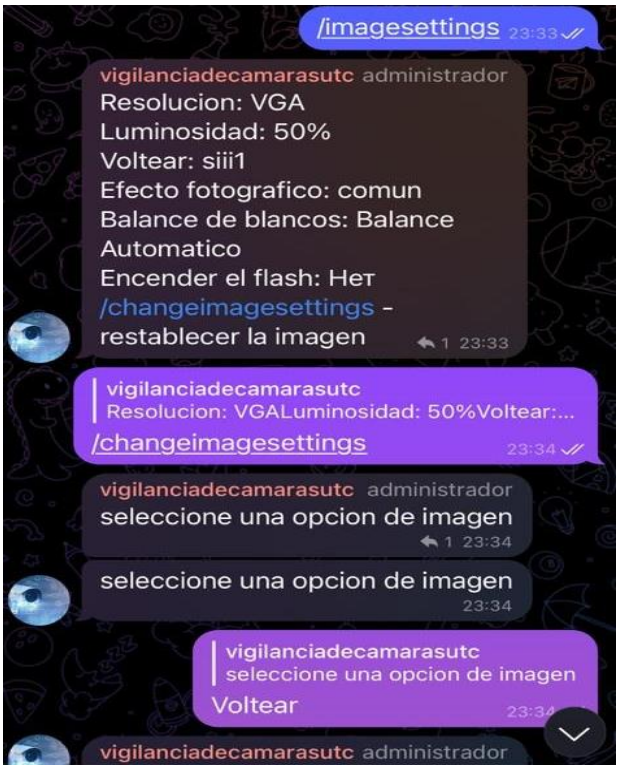


Figura 3.23 Comando /imagesettings

Para mejorar la visión de puntos ciegos comparamos con el sensor ov2640 y ov2640 con lente gran angular y tiene mayor área de visibilidad y mayor claridad en la figura 3.24 está con el sensor con lente gran angular.

Figura 3.24 Sensor ov2640 con lente gran angular

3.8 ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

3.8.1 Pruebas de funcionamiento

Para los ensayos de funcionamiento están establecidas en ciertos puntos los cuales son los más importantes del proyecto, estos son la confiabilidad y la distancia operativa.

Raspberry Pi3 brinda muchos beneficios los cuales son más beneficios de desventajas, a comparación del proyecto busca mejorar el precio, en la tabla 4 está detallado.

Tabla 4. Ventajas y desventajas de Raspberry Pi3.

Ventajas	Desventajas
Monitoreo en tiempo real	No tiene registro de acceso para Usuarios
Límite de usuarios en línea	Costo del equipo 240$
Genera código Ip para abrir desde cualquier navegador	Genera un ruido
No necesita ninguna aplicación para su funcionamiento	Depende de una micro SD para su almacenamiento

Telegram permite a los usuarios revisar desde cualquier lugar del mundo y tener información sobre quien envió algún requerimiento, lo favorable de este método es la gratuidad de la aplicación y no hay riesgo a que de desprograme en la tabla 5 esta descrito los argumentos positivos y negativos de la aplicación.

Tabla 5. Ventajas y desventajas de Telegram Pi3

Ventajas	**Desventajas**
Máximo de usuario de 250000	Depende de una aplicación
Registro de usuario quien indico un comando	No es de monitoreo en tiempo real
Capacidad de 50Gb de almacenamiento	
No tiene costo	

Confiabilidad

Para realizar estas pruebas, se comenzó registrando en un rango de 10 pruebas diarias, las veces que al escribir el comando /takephoto, nos envía la imagen/ fotografía al chat de Telegram, y a su vez el tiempo que tarda en llegar la respuesta es de 7 – 9 segundos, obteniendo así los siguientes datos. En la tabla N°3 puede observar los datos obtenidos, al pulsar /recordvideo enviara un video de 15 – 20 segundos, el video tardara en llegar 120 segundo después de enviar el comando y poder observar las grabaciones requeridas. Al pulsar /flash se encenderá la luz integrada de la placa y me permitirá solo grabar en la noche el comando tardará en notificar que el flas esté encendido/apagado 2 – 3 segundos. Escribiendo el comando /pant los servomotores se encenderán y girarán 40° verticalmente y al escribir /Tilt el giro será horizontal.

Para cambiar los parámetros de visualización de la resolución de las fotografías digitaremos el siguiente comando /imagesettings el cual nos dará parámetros para reajustar la luminosidad, formato, balances de blancos y nos dará la opción de tomar fotografías al encender el flash

Tiempos de repuesta de Telegram

Tabla 6. Tiempos de repuesta de Telegram

Comandos	Tiempo de respuesta (segundos)
/takephoto	7 – 9
/recordvideo	120
/flash	2
/pant	2
/tilt	2
/imagesettings	2

En la tabla 6, se realizó la medición de tiempos para saber en que lapso llega las respuestas, en la tabla 7, está reflejada las pruebas necesarias para poder saber la confiabilidad del proyecto, como resultado podemos analizar que al momento de encender el flash conectado en línea 2 cámaras la respuesta es más lenta y se alteran los resultados con la

Tabla 7. Porcentaje de confiabilidad

Comandos	**Repeticiones**	**Éxito**	**Error**	**Resultados (%)**
/takephoto	10	10	0	100
/recordvideo	10	10	0	100
/flash	10	10	2	100
/pant	10	8	1	90
/tilt	10	9	2	80
/imagesettings	10	10	0	100

El resultado del estudio arroja un 95 % de éxito al momento de enviar comandos, el 5 %

corresponde a problemas de control del movimiento con los servomotores.

Figura 3.25 Porcentaje de confiabilidad del prototipo usando Telegram

Distancia operativa

Según el datasheet del ESP32-S que se halla en el ANEXO, el microcontrolador tiene dos estados para la señal Wifi; TX que es el modo Wifi para la transferencia de esta señal, que es la cual recoge la señal Wifi de otro dispositivo. En la tabla 8, se observa los estándares IEEE 802.11

Tabla 8. Estándar IEEE 802.11 para la ESP32-S

Tipo	Frecuencia	Cobertura
Señal RX	802.11 n	20 m
Señal RX	802.11 g	60 m
Señal RX	802.11b	60 m
Señal TX	802.11 n	20 m
Señal TX	802.11 g	60 m

Señal TX	802.11 b	60 m

La prueba de expresión de señal WiFi, se tomó como ejemplo al prototipo de cámara del ESP32-CAM, de esta manera los resultados se aplicarán para la configuración de la red a la misma que ya está configurada para que tenga acceso como esta representada en la tabla 9.

Tabla 9. Datos obtenidos para Tx y Rx del ESP32-CAM

Tipo de Señal	Distancia Media	Cumple
TX	10 metros	Si
RX	40 metros	No

Cuando ya se hayan configurado los dispositivos con la red que va a conectarse, el alcance dependerá del router, pero como se utiliza la señal RX del protocolo IEEE 802.11g en la tabla 9 se tiene las especificaciones mencionadas, en lo cual la placa no deberá tener mayor grado de dificultad porque está dentro del rango de longitud de alcance como se lo evidencia en la figura 3.26.

Figura 3.26 Alcance de recepción de señal Wifi.

3.8.2 Estabilidad de conexión Wifi

En esta prueba se tomó en cuenta 18 horas de un día, con la verificación cada 60 min para comprobar que el prototipo de la cámara este activo y tenga conexión a internet en la tabla

10 se evidencia los datos obtenidos de la conexión.

Tabla 10. Datos de estabilidad de conexión a la red.

Estado de conectividad	
Tiempo	**Conexión**
06:00	**1**
07:00	**1**
08:00	**1**
09:00	**1**
10:00	**1**
11:00	**1**
12:00	**1**
13:00	**1**
14:00	**1**
15:00	**1**
16:00	**1**
17:00	**1**
18:00	**1**
19:00	**1**
20:00	**1**
21:00	**1**
22:00	**1**
23:00	**1**

En la tabla 10 están los datos obtenidos y ponderados de la siguiente manera, con el significado de conexión activa 1 y cuando la conexión haya fallado 0 y por mayor factibilidad no se perdió la conexión a ningún momento es decir que ya no se debe volver a configurar la cámara a la red por tanto el funcionamiento es del 100% en conectividad.

Figura 3.27 Porcentaje de conectividad.

Prueba de conectividad

Para esta prueba se toman los resultados al mantenerse encendido el router y el prototipo conectándole y desconectando para poder obtener datos de conexión.

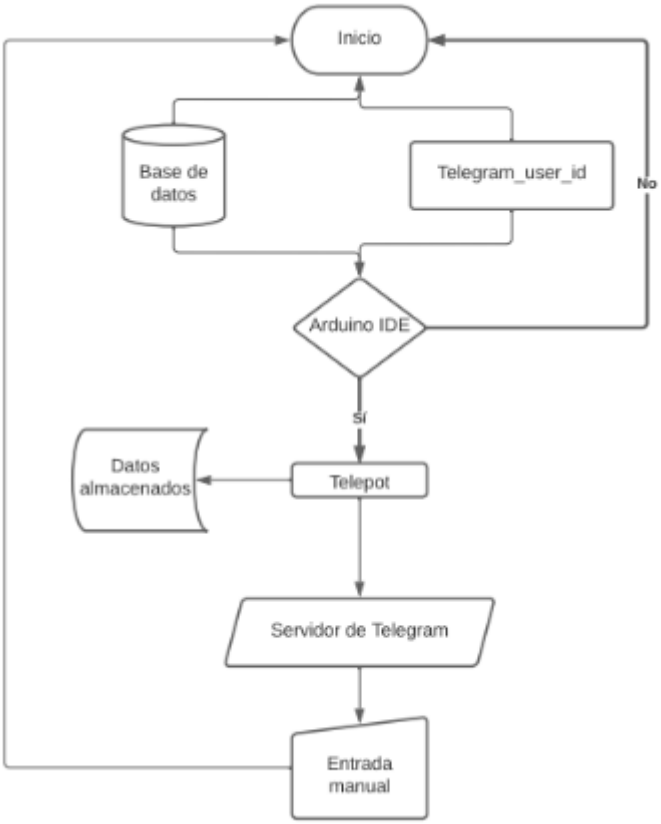


Figura 3.28 Diagrama de flujo del funcionamiento

3.9 VIABILIDAD DEL PROYECTO

3.9.1 Resultados de comparación de costos.

Para poder obtener resultados comparativos utilizamos los siguientes procesos que se visualiza en la tabla 11 que me está detallando de los costos.

Tabla 11. Equipos y costos

Mensajería con Telegram	cantidad	Precio ($)	Transmisión con Raspberry Pi3	cantidad	Precio ($)
Esp32-CAM	1	15,00	Raspberry Pi3	1	250,00
Servomotor 1	2	6,00	Suscripción mensual Pitunnel	1	5,00
Adaptador 5 v	1	4,00	Esp32-CAM	1	15,00
Impresión 3d	1	5,00	Servomotor 1	2	6,00
Tornillos 2 mm	10	1,50	Adaptador 5 v	1	4,00
		31,50	Tornillos 2 mm	10	1,50
					286,50

En la tabla 11 está descrito los costos de cada instrumento necesario para el prototipo llegando a concluir que el precio final utilizando la aplicación de mensajería da un costo de 31.50 dólares y comparándola con la transmisión de Raspberry Pi3 tengo un costo total de 286.50 dólares que para nuestro producto tenemos la necesidad de abaratar costos es fiable utilizar la aplicación de mensajería a través de Telegram ya que se obtiene un 90% menos comparándola con la transmisión de Raspberry Pi3.

Para el ensamble de una cámara:

1. Abrir el programa ARDUINO IDE (5 min)
2. Conectar la placa ESP32 CAM al programador y conectar al computador para subir la programación(10min).
3. Ensamble del case protector(30min)

4. Comprobación del correcto funcionamiento. (10 min)
5. Empaque y etiquetado. (10 min)

Tiempo total del proceso de armado de una cámara IP con control de ángulos: 65 min (1h 5 min).

El producto será ensamblado dependiendo la demanda de los clientes.

Lista de los materiales

- Placa ESP 32 CAM (Electrónica J/M, Latacunga, 0958753126, 15,00$)
- Servomotores (Electrónica J/M, Latacunga, 0958753126, 2,50 $)
- Cable USB (Electrónica J/M, Latacunga, 0958753126, 3,00 $)
- Adaptador 5v (Electrónica J/M, Latacunga, 0958753126, 5,00 $)
- Impresión 3D del case (ING: Diego Corrales, 0995820380, 4,00$)
- Caja de cartón y etiquetas (Digital ART,0995807154, 1,00 $)
- Juego de destornilladores (Kywi, 15,00$)
- Cautín (Kywi, 55,00$)
- Cables jumper mixto (El contactor, 0999944467 2,30 $)
- Computador con programas para trabajo
- Mesa de trabajo de 1*2

Esta inversión para implementar el taller es: 70,00 $

El precio total de la cámara ensamblada es de 31,50 $

En la figura 3.17 podemos observar cómo estaría conformado las funciones del lugar para la fabricación y ensamble del producto.

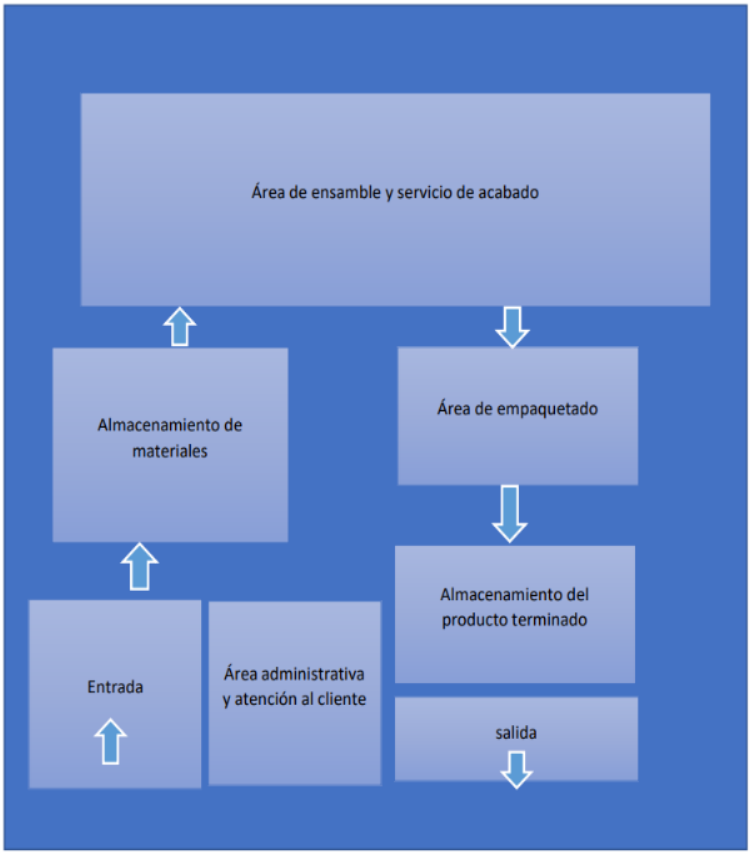


Figura 3.29 Diagrama de funciones de distribuciones.

En la figura 3.30 está planteado un plan de negocio al estilo de CAMVAS en la cual tenemos identificado el mercado al que queremos llegar con nuestro producto

Figura 3.30 Modelo de negocio CANVAS.

3.10 EVALUACIÓN TÉCNICA

3.10.1 Evaluación técnica

El perfeccionamiento de los prototipos de tarjetas electrónicas da un progreso en las diferentes destrezas para la vida profesional, tanto para la administración del uso de software para establecer códigos de programación, así como para el boceto de archivos CAD y el diseño de circuitos esquemáticos para la realización de placas de circuito impreso. Además, incrementa la proactividad que es necesaria para todo ámbito profesional y personal, ya que, con este proyecto hay que indagar en temas que no se reciben durante los ciclos de estudio como por ejemplo el uso de base de datos y las comunicaciones mediate redes inalámbricas. También brinda la oportunidad de seguir agregando mejoras a los prototipos, ya que, al ser la primera versión siempre va a tener la posibilidad de corregir errores que no son perceptibles en la fase de desarrollo.

3.10.2 Evaluación tecnológica

Este proyecto busca brindar un mejoramiento a los dispositivos similares que hay en el mercado

actual; brindando así varios beneficios como el tener el control de visibilidad y monitoreo de viviendas a través del control desde un celular inteligente, y que además pueden tener varios usuarios de una comunidad que pueden controlar el funcionamiento de las cámaras, y que tiene la capacidad de interactuar con una aplicación móvil que de igual manera controla y envía notificaciones tras la activación por alguna emergencia.

Con estos prototipos también brinda a la Universidad Técnica de Cotopaxi la capacidad de competir con los productos que se ofrecen en el mercado, implementando un sistema de video vigilancia con control de ángulos que sea con la marca UTC, para que los habitantes de Cotopaxi y el Ecuador sepan que la universidad crea cada semestre una nueva tecnología capaz de satisfacer las necesidades

que el pueblo requiera.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Mediante la aplicación de mensajería de Telegram fue posible el control de las cámaras IP, para ello se enlazó el ESP32-CAM al sistema y mediante los comandos el grupo de chat se logró modificar los movimientos de los servomotores.
- Tras el análisis de comparación entre el tunnel de Raspberry Pi3 y Api de Telegram se pudo deducir que con raspberry Pi3 se puede monitorear en tiempo real a comparación de Api de Telegram el monitoreo es retardado 20 segundos como está representado en la Tabla 6, de tiempos de respuestas de comandos.
- El prototipo de cámaras de video vigilancia tiene una eficiencia del 98 % para el movimiento de ángulos, en la Tabla 7, se obtuvo datos para evidenciar el funcionamiento, y además el equipo y tiene un 100 % de confiabilidad en la conexión a la red como se observó en la Figuras 3.16.
- El número de dispositivos conectados no interfiere en el rendimiento y respuesta de las cámaras, además que no obstruye el procesamiento y flujo de datos de la red local.

4.2 RECOMENDACIONES

- Analizar nuevas pruebas de mejora con el sensor OV2640 con el lente gran angular y puedan tener mejor calidad de imagen.
- Incluir una batería recargable en la cual la cámara pueda seguir en su funcionamiento sin importar la ausencia de la energía eléctrica.
- Buscar nuevos métodos para lograr obtener el almacenamiento masivo de las grabaciones y no estar pendientes con las micro SD de almacenamiento.
- Para próximos proyectos relacionados se recomienda crear un prototipo con luz infrarroja para poder monitorear en la obscuridad y crear un código y comando para seleccionar la cámara a la cual se le va a dar la orden de efectuar.

5. BIBLIOGRAFÍA

[1] «fiscalía general del Estado | Analítica cifras de robo». https://www.fiscalia.gob.ec/analitica-cifras-de-robo/ (accedido 6 de diciembre de 2022).

[2] C. Binker *et al.*, «Gestión remota de dispositivos IOT mediante técnicas de mensajería instantánea empleando Bots», Accedido: 4 de febrero de 2023. [En línea]. Disponible en: https://api.telegram.org/bot

[3] R. Moreno Hernández y A. León Fernández, «Desarrollo de una aplicación IoT para la gestión de un hogar inteligente mediante el protocolo MQTT y Sistemas en chip (SoC) ESP32», 2019, Accedido: 4 de febrero de 2023. [En línea]. Disponible en: www.etsit.upv.es

[4] Y. Rahmawati, I. Uli, V. Simanjutak, y R. B. Simorangkir, «Rancang Bangun Purwarupa Sistem Peringatan Pengendara Pelanggar Zebra Cross Berbasis Mikrokontroller ESP-32 CAM», *Jambura Journal of Electrical and Electronics Engineering*, vol. 4, n.º 2, pp. 189-195, jul. 2022, doi: 10.37905/JJEEE.V4I2.14499.

[5] «Seguridad física: ¿en qué consiste y cómo puede ser de ayuda?» https://www.seguridadsuperior.com.co/que-es-la-seguridad-fisica (accedido 29 de diciembre de 2022).

[6] «Qué son los sistemas de seguridad electrónica y sus diferencias con el control físico - Compucima». https://compucima.com.ec/sistemas-de-seguridad-electronica-y-control-fisico/ (accedido 29 de diciembre de 2022).

[7] «¿Qué es la Seguridad Electrónica? - Secatel SCC». https://secatel.com/que-es-la-seguridad-electronica/ (accedido 29 de diciembre de 2022).

[8] «Qué es la Seguridad Electrónica y Porqué es Importante». https://gruponavarro.pe/blog/la-seguridad-electronica/ (accedido 29 de diciembre de 2022).

[9] «(1) Nuevo Mensaje!» https://www.seguridadsuperior.com.co/seguridad-electronica (accedido 30 de diciembre de 2022).

[10] «Tipos de sistemas de seguridad electrónicos | Baterias y Amperios». https://bateriasyamperios.com/tipos-de-sistemas-de-seguridad-electronicos/ (accedido 3 de enero de 2023).

[11] «¿Qué son las redes y cómo funciona Internet? | EDteam».

https://ed.team/blog/que-son-las-redes-y-como-funciona-internet (accedido 4 de enero de 2023).

[12] «Tipos de conexiones a internet y sus ventajas - VidaBytes | VidaBytes». https://vidabytes.com/tipos-de-conexiones/ (accedido 4 de enero de 2023).

[13] «Tipos de conexiones a internet y sus ventajas - VidaBytes | VidaBytes». https://vidabytes.com/tipos-de-conexiones/ (accedido 4 de enero de 2023).

[14] «▷Redes Inalambricas Aprende Facil Todo Lo que Necesitas». https://www.redesinalambricas.es/ (accedido 4 de enero de 2023).

[15] J. Salazar, «REDES INALÁMBRICAS», Accedido: 6 de enero de 2023. [En línea]. Disponible en: http://www.techpedia.eu

[16] «Qué es el estándar WPAN y para qué se utiliza». https://www.redeszone.net/tutoriales/redes-wifi/que-es-estandar-wpan/ (accedido 6 de enero de 2023).

[17] «Red inalámbrica WLAN | Guías Prácticas». https://www.guiaspracticas.com/internet-y-redes/red-inalambrica-wlan (accedido 6 de enero de 2023).

[18] « Red WMAN ¿Qué es y Cómo Funciona? 2023». https://tarify.win/definiciones/red-wman/ (accedido 6 de enero de 2023).

[19] «WMAN vs WWAN: diferencias y usos de estos tipos de redes inalámbricas». https://www.redeszone.net/tutoriales/redes-wifi/wman-wwan-diferencias-usos-redes-inalambricas/ (accedido 6 de enero de 2023).

[20] «【 Red WWAN 】¿Qué es y Para Qué Sirve? Ejemplos ▷ 2023». https://internetpasoapaso.com/red-wwan/ (accedido 6 de enero de 2023).

[21] «Componentes WLAN | MindMeister Mapa Mental». https://www.mindmeister.com/es/778825868/componentes-wlan (accedido 6 de enero de 2023).

[22] «▷ Todo acerca de una Red ad hoc inalámbrica » CCNA desde Cero». https://ccnadesdecero.es/red-ad-hoc-inalambrica-wanet/ (accedido 6 de enero de 2023).

[23] «Sitio web de soporte de Brother». https://support.brother.com/g/b/error.aspx?etype=pkey&err=%2fg%2fb%2ffaqend.aspx%3fc%3dus%26lang%3des%26prod%3dhl4070cdw_all%26faq (accedido 6 de enero de 2023).

[24] «¿Qué es Wi-Fi? | Definición, significado y explicación | Verizon». https://espanol.verizon.com/articles/internet-essentials/wifi-definiton/ (accedido 6 de enero de 2023).

[25] «tecnología wifia». https://www.ibersystems.es/servicios/instalacion-redes-inalambricas/instalaciones-wifi/tecnologia-wifi/ (accedido 6 de enero de 2023).

[26] *IEEE 802.11n Approval*. Accedido: 6 de enero de 2023. [En línea]. Disponible en: http://standards.ieee.org/announcements/ieee802.11n_2009amendment_ratified.html

[27] «¿Qué es el Internet de las cosas?» https://www.redhat.com/es/topics/internet-of-things/what-is-iot (accedido 8 de enero de 2023).

[28] «¿Qué es IoT y cómo funciona? | SAP Insights». https://www.sap.com/latinamerica/insights/what-is-iot-internet-of-things.html (accedido 16 de enero de 2023).

[29] «ESP32-CAM: lo que debes saber sobre este módulo | Hardware libre». https://www.hwlibre.com/esp32-cam/ (accedido 16 de enero de 2023).

[30] G. A. M. Fernández, A. C. Contreras, V. G. Hernández Herrera, y M. V. M. Olivera, «CONTROLADOR MULTIEJE DE POSICIONAMIENTO DE SERVOMOTORES BLDC IMPLEMENTADO EN FPGA», 2017. [En línea]. Disponible en: http://itcelaya.edu.mx/ojs/index.php/pistas

[31] F. Bordignon, A. A. Iglesias, y Á. Hahn, «DISEÑO E IMPRESIÓN DE OBJETOS 3D Una guía de apoyo a escuelas».

[32] «UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL CARRERA DE INGENIERIA AUTOMOTRIZ “ESTUDIO Y CARACTERIZACIÓN DE PROPIEDADES MECÁNICAS DE PIEZAS COMPUESTAS DE FIBRAS CARBONO Y NYLON PRODUCIDAS MEDIANTE IMPRESIÓN 3D”».

[33] «ESP32-CAM con cámara OV2640 ESP32 WiFi». https://naylampmechatronics.com/espressif-esp/700-esp32-cam-con-camara-ov2640-esp32-wifi.html (accedido 5 de febrero de 2023).

[34] «ANPR en Raspberry Pi | Reconocedor de placas ALPR». https://platerecognizer.com/anpr-on-raspberry-pi/?utm_source=bing&utm_medium=cpc&utm_campaign=1356797997295736&utm_keyword=https%3A%2F%2Fplaterecognizer.com%2Fanpr-on-raspberry-pi%2F&utm_content=142211&msclkid=6795326c50e01a84f8c251a464b96115

(accedido 17 de enero de 2023).

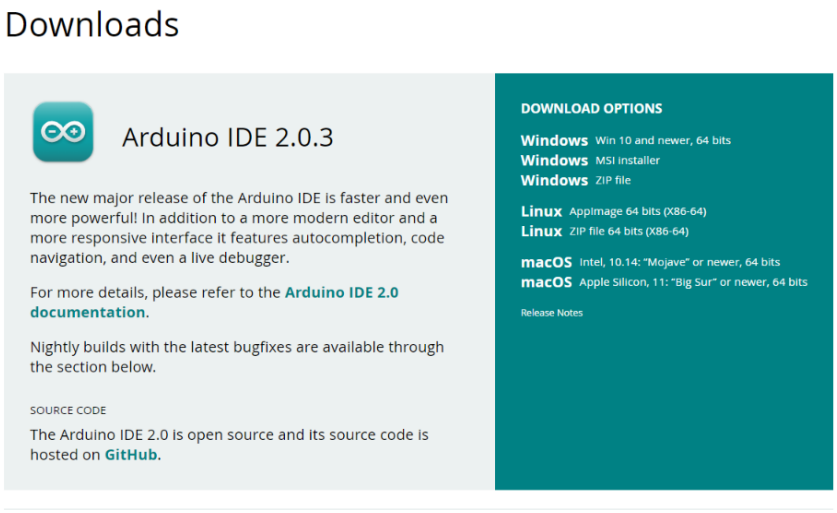
[35] «PiTunnel - Accede a tus proyectos de Raspberry Pi desde cualquier lugar». https://www.pitunnel.com/ (accedido 17 de enero de 2023).

[36] «Telegram: ESP32-CAM Take and Send Photo (Arduino IDE) | Random Nerd Tutorials». https://randomnerdtutorials.com/telegram-esp32-cam-photo-arduino/ (accedido 17 de enero de 2023).

ANEXOS

ANEXO I	

Para obtener el programa se debe ingresar a la página oficial de Arduino www.arduino.cc/en/software en donde debemos escoger en la primera opción para descargar como se visualiza en la figura 7 dependiendo del sistema operativo que estemos utilizando.



a. Instalación del programa de interfaz de Arduino IDE

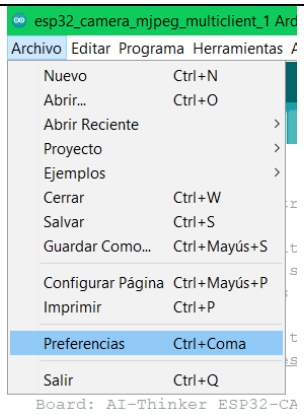
Se debe proceder a abrir el archivo descargado, debemos desbloquearlo y ejecutarlo para lo cual debemos seleccionar las opciones de siguiente y aceptar cuando nos solicite los permisos.

b. Instalación de librerías requeridas por Esp32-CAM

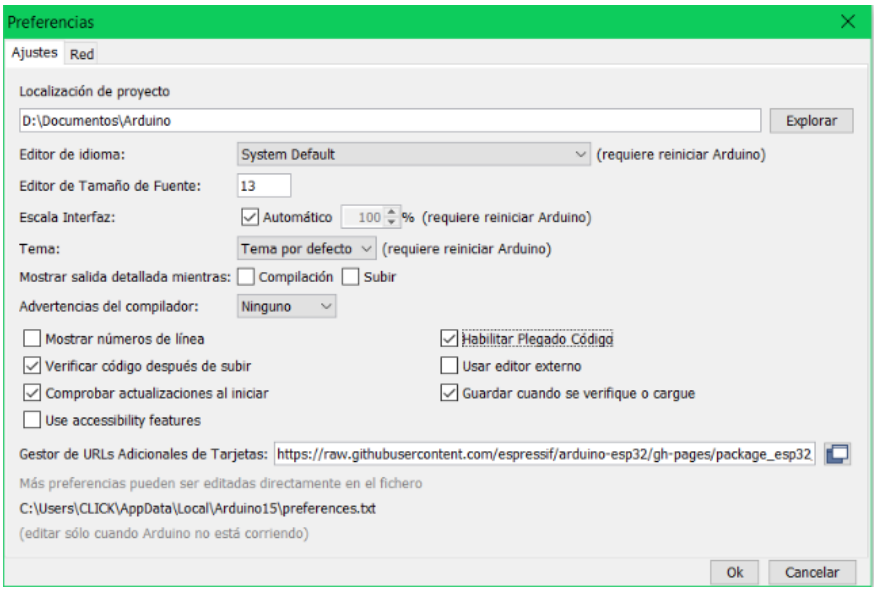
Para poder empezar a programar con el ESP32-CAM y la datasheet o firebase, se requiere descargar diferentes librerías para que pueda reconocer al microcontrolador y puedan transferir datos de información

Librerías ESP32-CAM

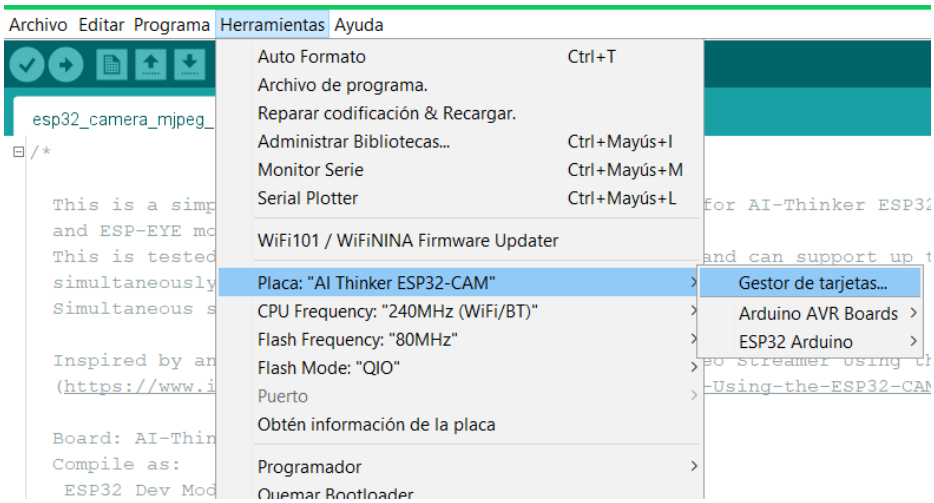
Una vez que se haya instalado correctamente se procede a ejecutar el programa para abrir el software, se dirige a la pestaño de Archivos y seleccionar la opción preferencias como lo visualizamos en la figura 8.



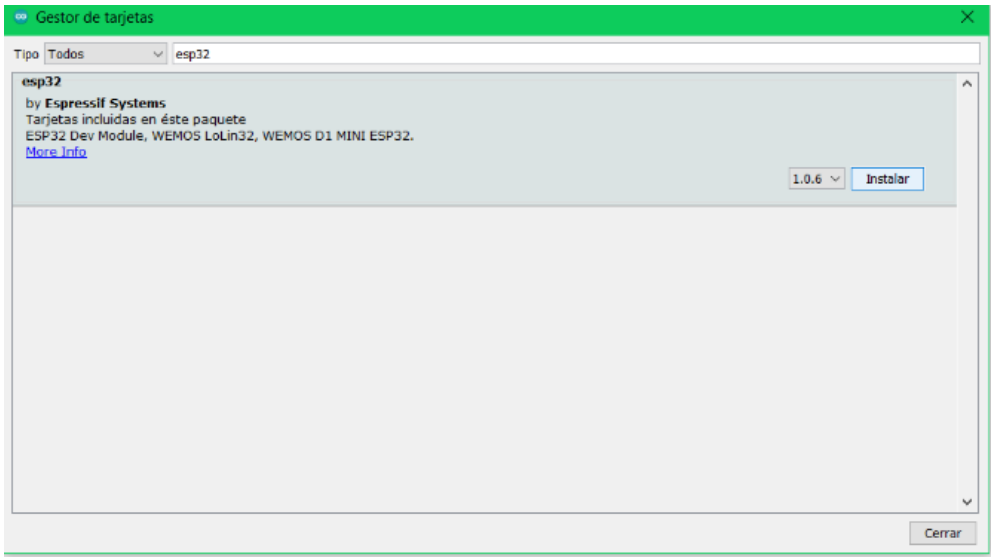
Una vez que se haya seleccionado la opción de preferencias se debe pegar el link https://raw.githubusercontent.com/espressif/arduino-esp32/gh pages/package_esp32_index.json, https://resource.heltec.cn/download/package_heltec_esp32_index.json,https://dl.espressif.com/dl/package_esp32_index.json en el apartado de gestor de tarjetas y seleccionar el botón de “ok” como se observa en la figura 9.



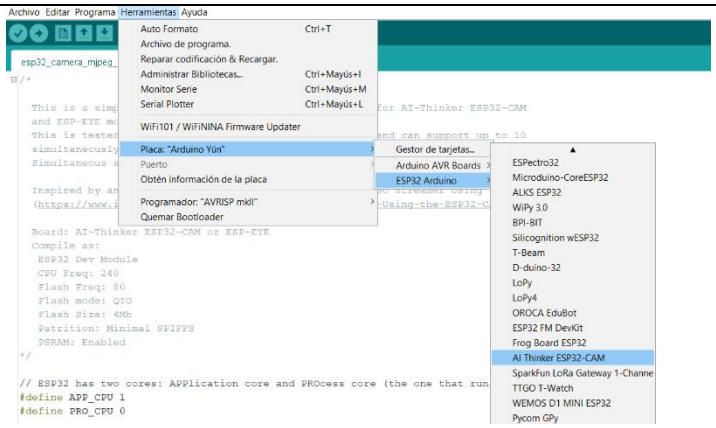
Una vez pegado el enlace de descargas de las librerías se dirigen a la pestaño de Herramientas y se dirige a placa y se selecciona la opción de gestor de tarjetas.



Para concluir con la instalación del último paso, se visualizará una ventana nueva que se abrirá luego de seleccionar y escribimos en el buscador ESP32 en la figura 11 evidenciamos la opción que debemos instalar.

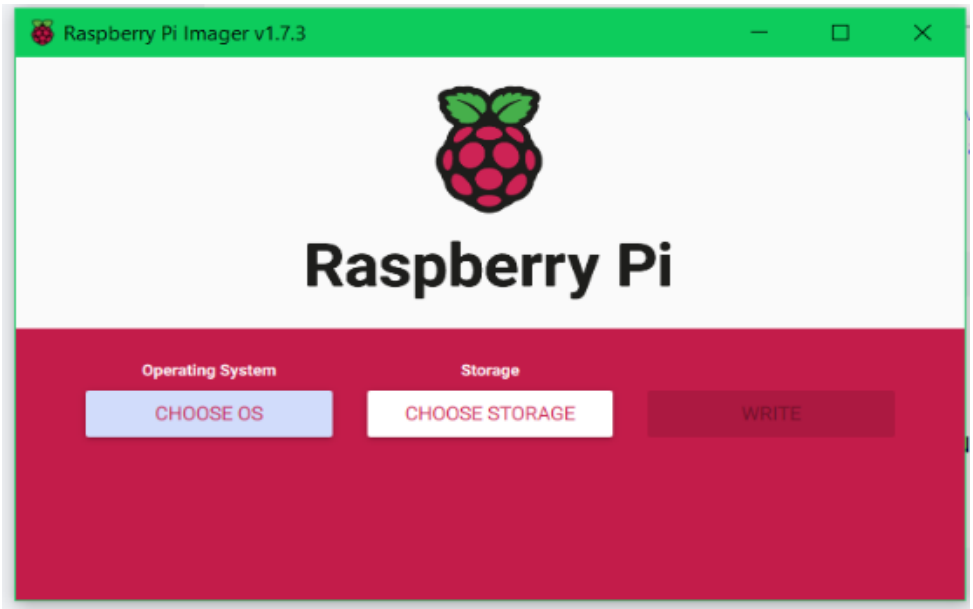


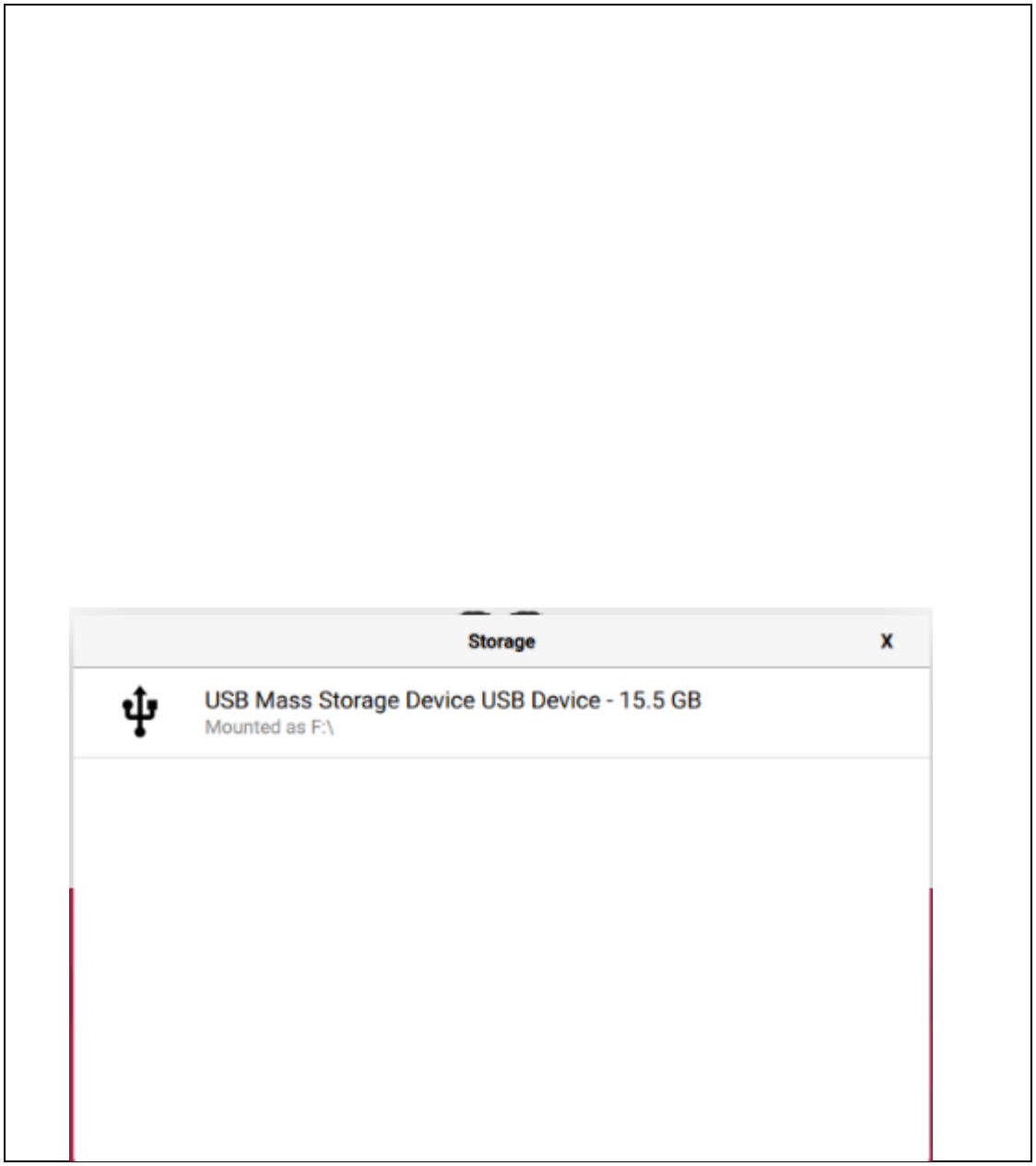
Terminados los pasos de instalación se debe seleccionar la pestaña herramientas y la opción de placa luego seleccionar ESP32 Arduino y escoger la placa Al Thinker ESP32-CAM como se evidencia en la figura 12.

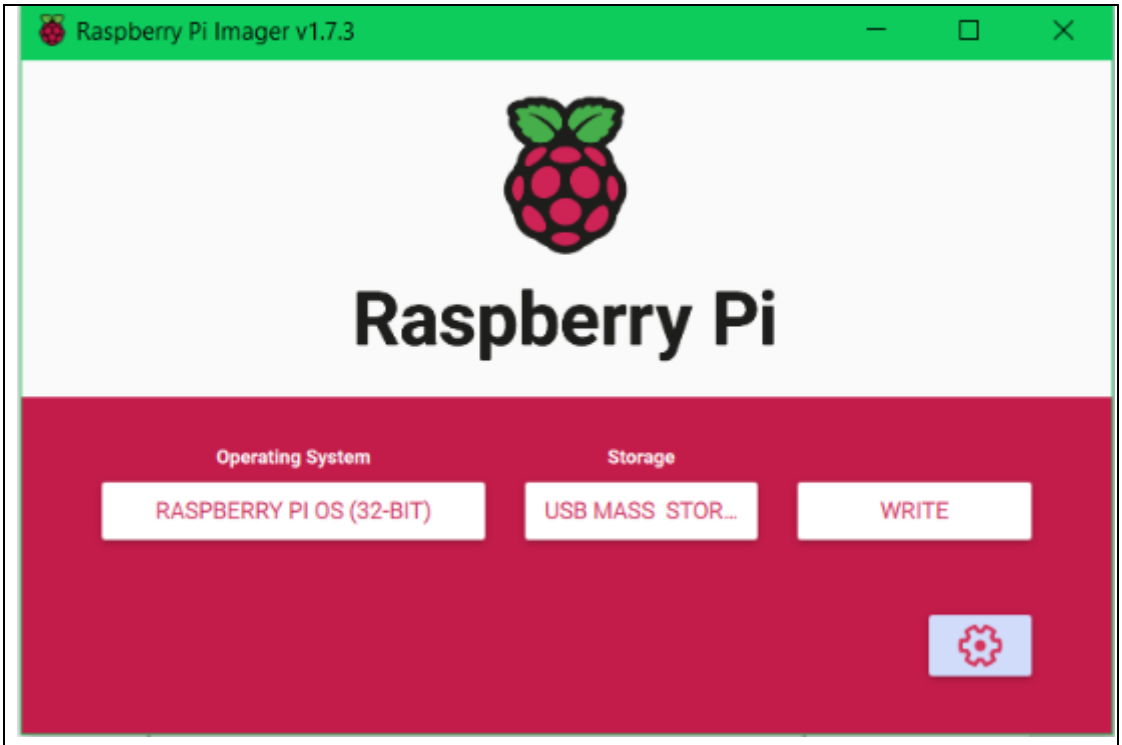


Cuando ya se haya terminado de instalar todas las librerías requeridas y necesarias para que el software reconozca nuestra placa y a la base de datos debe reconocer y dar paso a la subir el código.

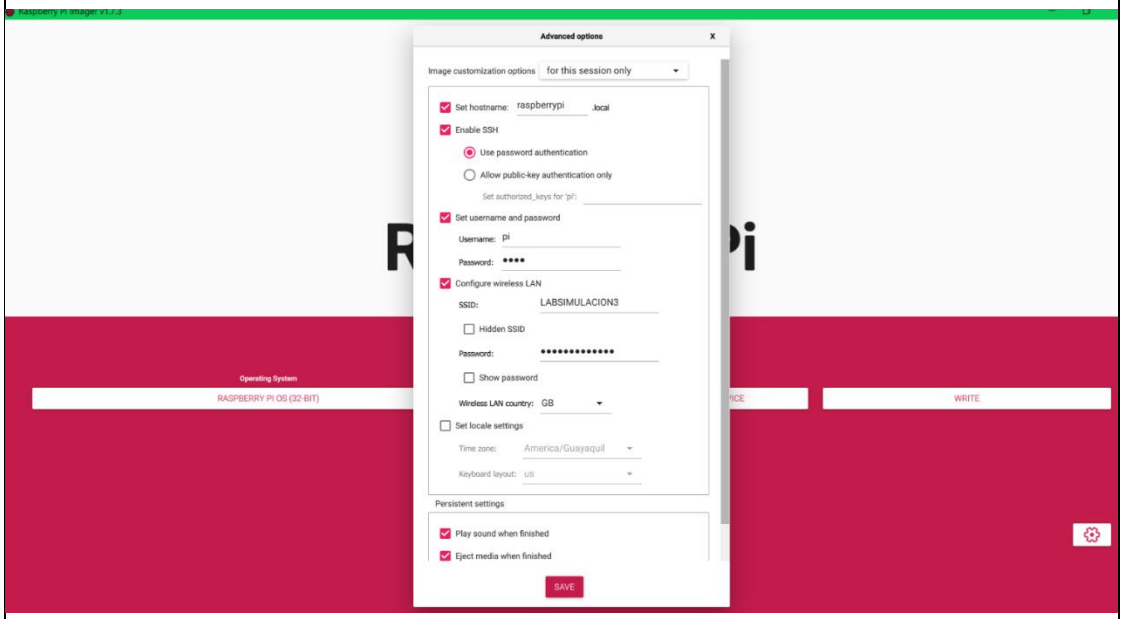
ANEXO II	PROCEDIMIENTO PARA ACTIVAR RASPBERRY




Storage
X
USB Mass Storage Device USB Device - 15.5 GB
Mounted as F:\


Raspberry Pi Imager v1.7.3
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS (32-BIT)
USB MASS STOR...
WRITE

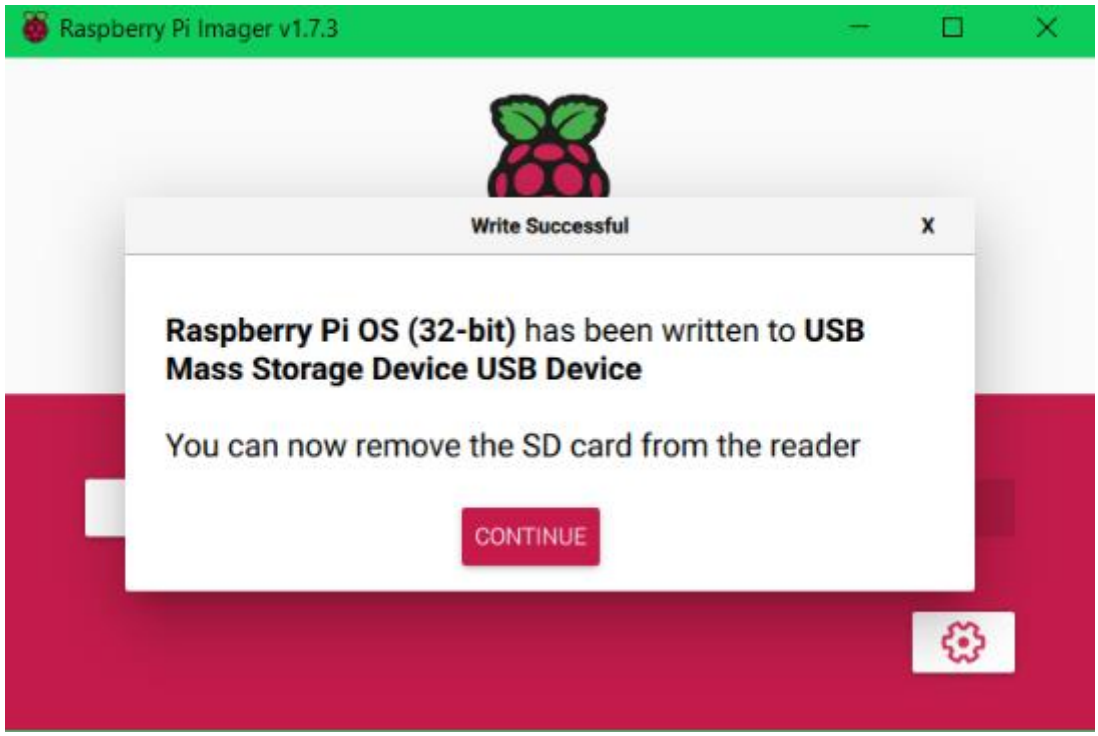

Raspberry Pi Imager v1.7.3
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS (32-BIT)
CHOOSE STORAGE
WRITE


Advanced options
Image customization options
for this session only
Set hostname:
raspberrypi
.local
Enable SSH
Use password authentication
Allow public-key authentication only
Set authorized_keys for 'pi':
Set username and password
Username:
pi
Password:
Configure wireless LAN
SSID:
LABSIMULACION3
Hidden SSID
Password:
Show password
Wireless LAN country:
GB
Set locale settings
Time zone:
America/Guayaquil
Keyboard layout:
us
Persistent settings
Play sound when finished
Eject media when finished
SAVE
Operating System
RASPBERRY PI OS (32-BIT)
WRITE

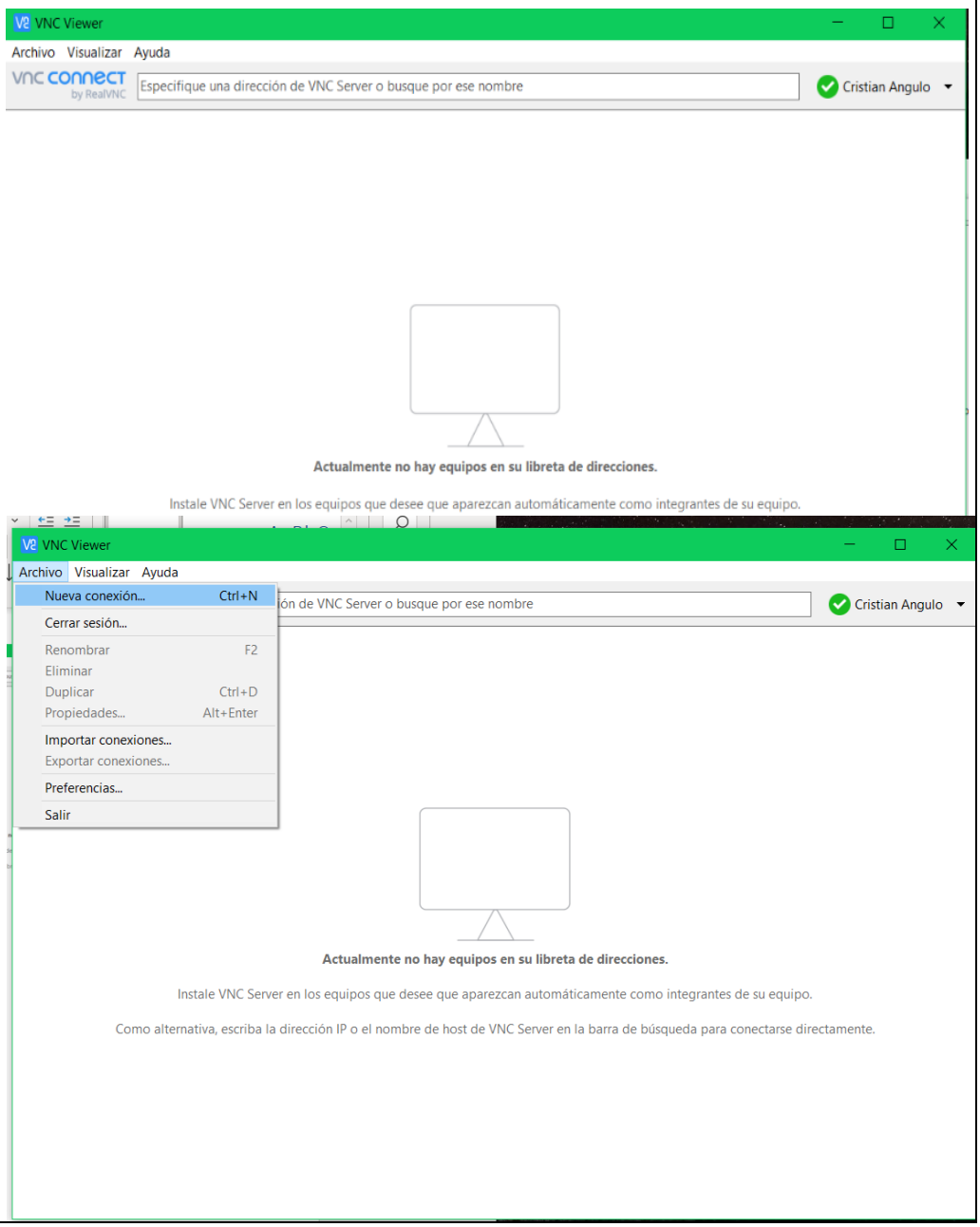

Raspberry Pi Imager v1.7.3
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS (32-BIT)
USB MASS STOR...
WRITE
Writing... 78%
CANCEL WRITE

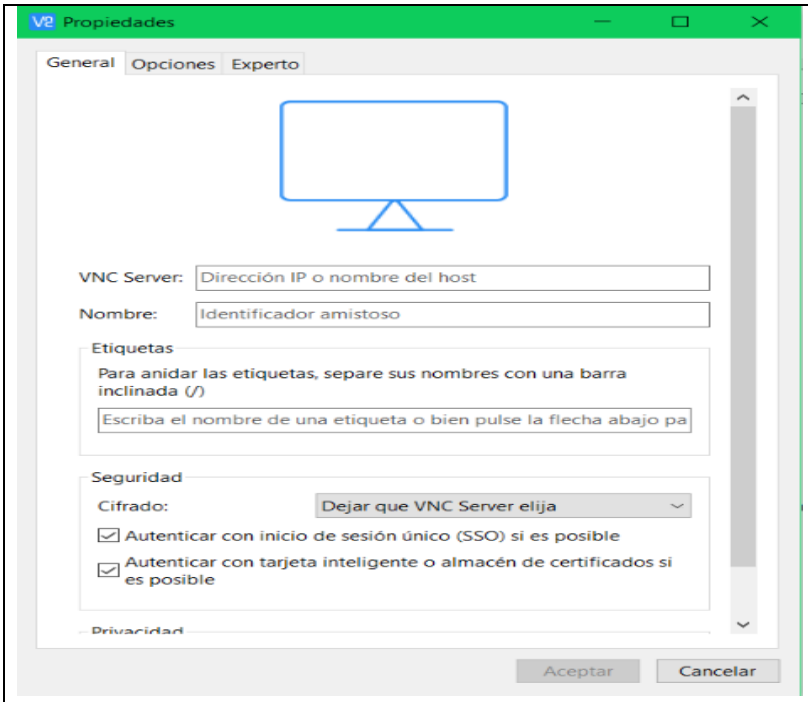

Raspberry Pi Imager v1.7.3
Raspberry Pi
Operating System
Storage
RASPBERRY PI OS (32-BIT)
USB MASS STOR...
WRITE
Verifying... 17%
CANCEL VERIFY

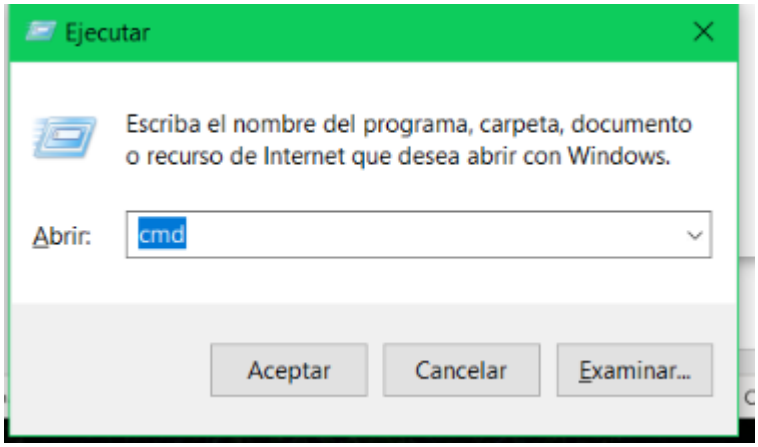
-

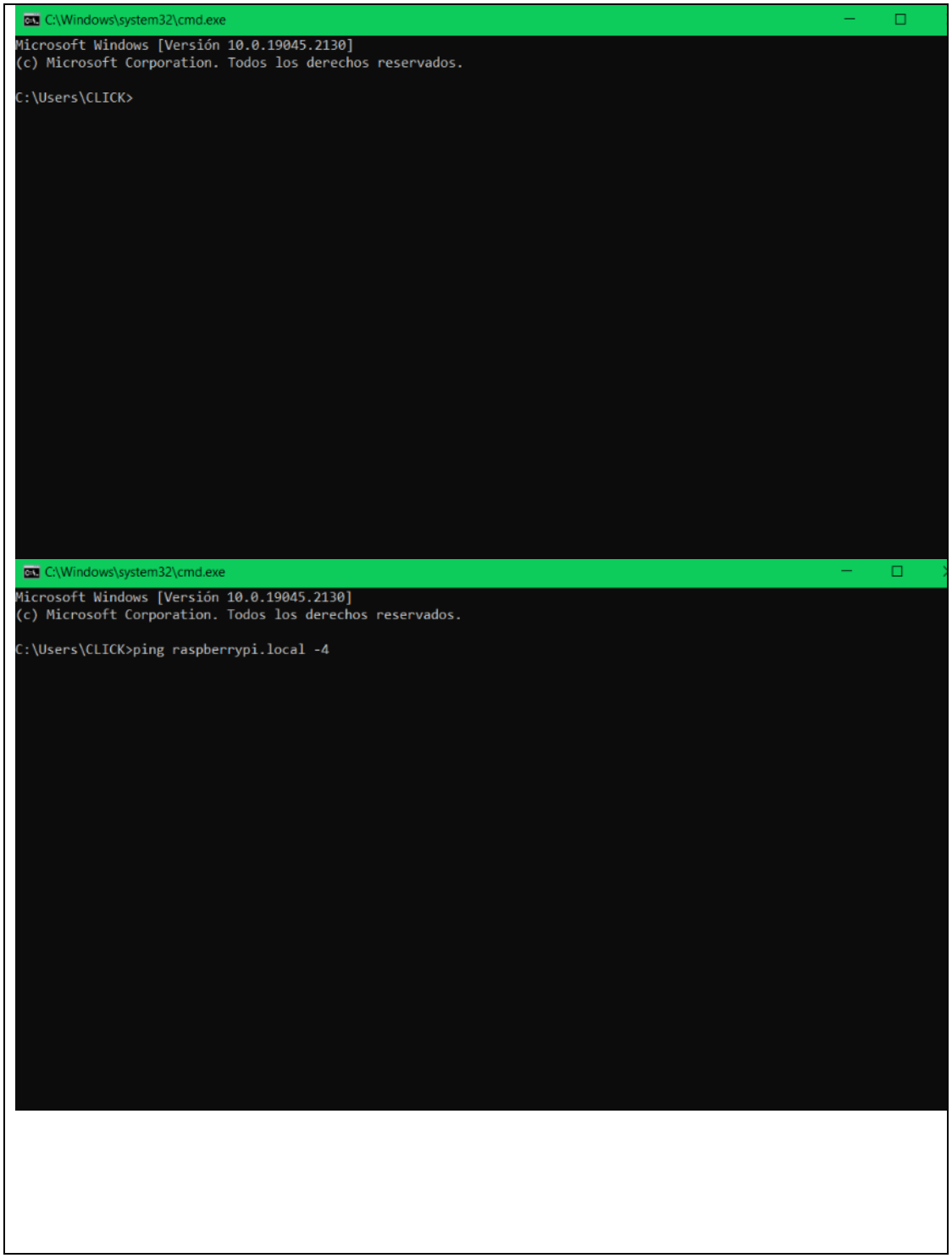

Raspberry Pi Imager v1.7.3
Write Successful
X
Raspberry Pi OS (32-bit) has been written to USB Mass Storage Device USB Device
You can now remove the SD card from the reader
CONTINUE

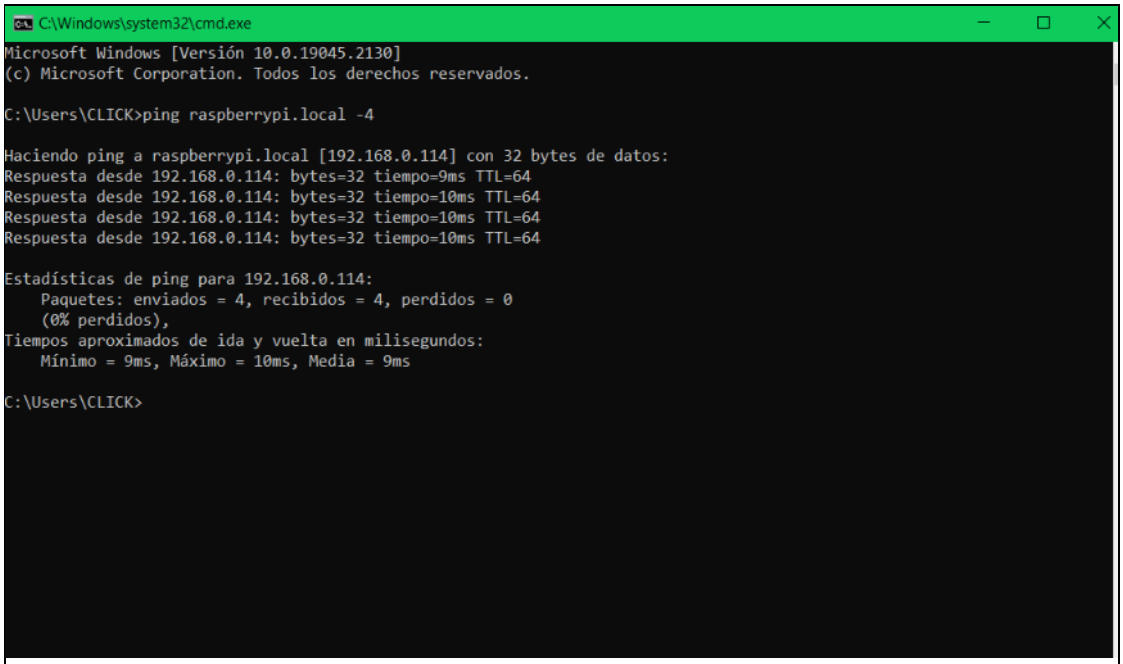
2.

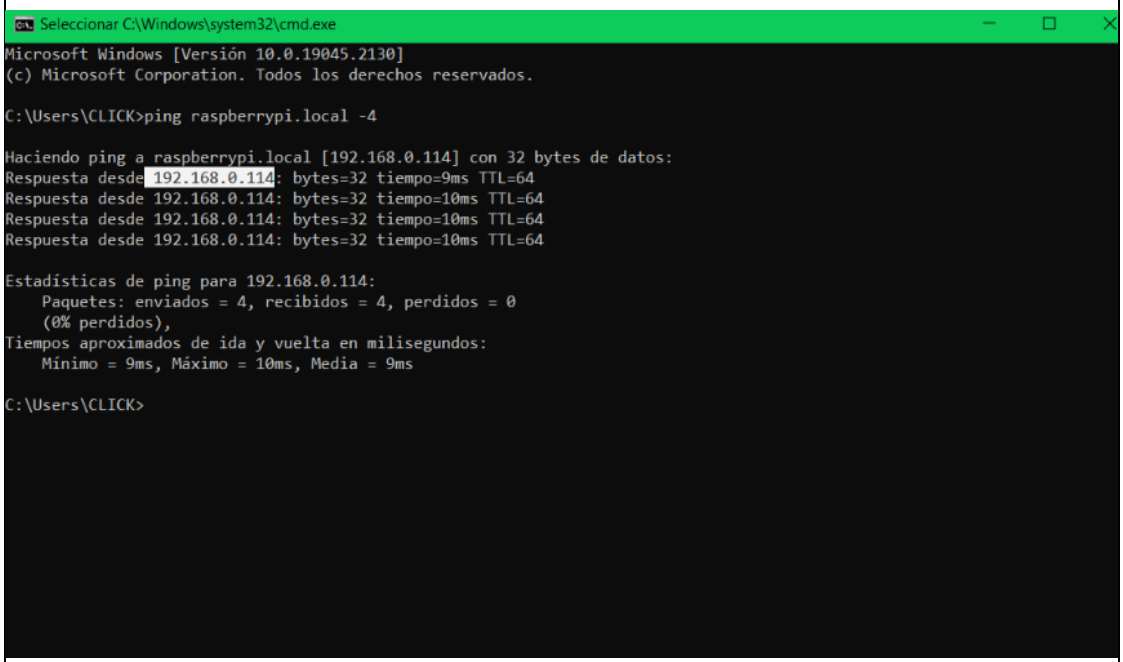

VNC Viewer
Archivo Visualizar Ayuda
vnc connect by RealVNC
Especifique una dirección de VNC Server o busque por ese nombre
Cristian Angulo
Actualmente no hay equipos en su libreta de direcciones.
Instale VNC Server en los equipos que desee que aparezcan automáticamente como integrantes de su equipo.
VNC Viewer
Archivo Visualizar Ayuda
Nueva conexión... Ctrl+N
Cerrar sesión...
Renombrar F2
Eliminar
Duplicar Ctrl+D
Propiedades... Alt+Enter
Importar conexiones...
Exportar conexiones...
Preferencias...
Salir
Cristian Angulo
Actualmente no hay equipos en su libreta de direcciones.
Instale VNC Server en los equipos que desee que aparezcan automáticamente como integrantes de su equipo.
Como alternativa, escriba la dirección IP o el nombre de host de VNC Server en la barra de búsqueda para conectarse directamente.

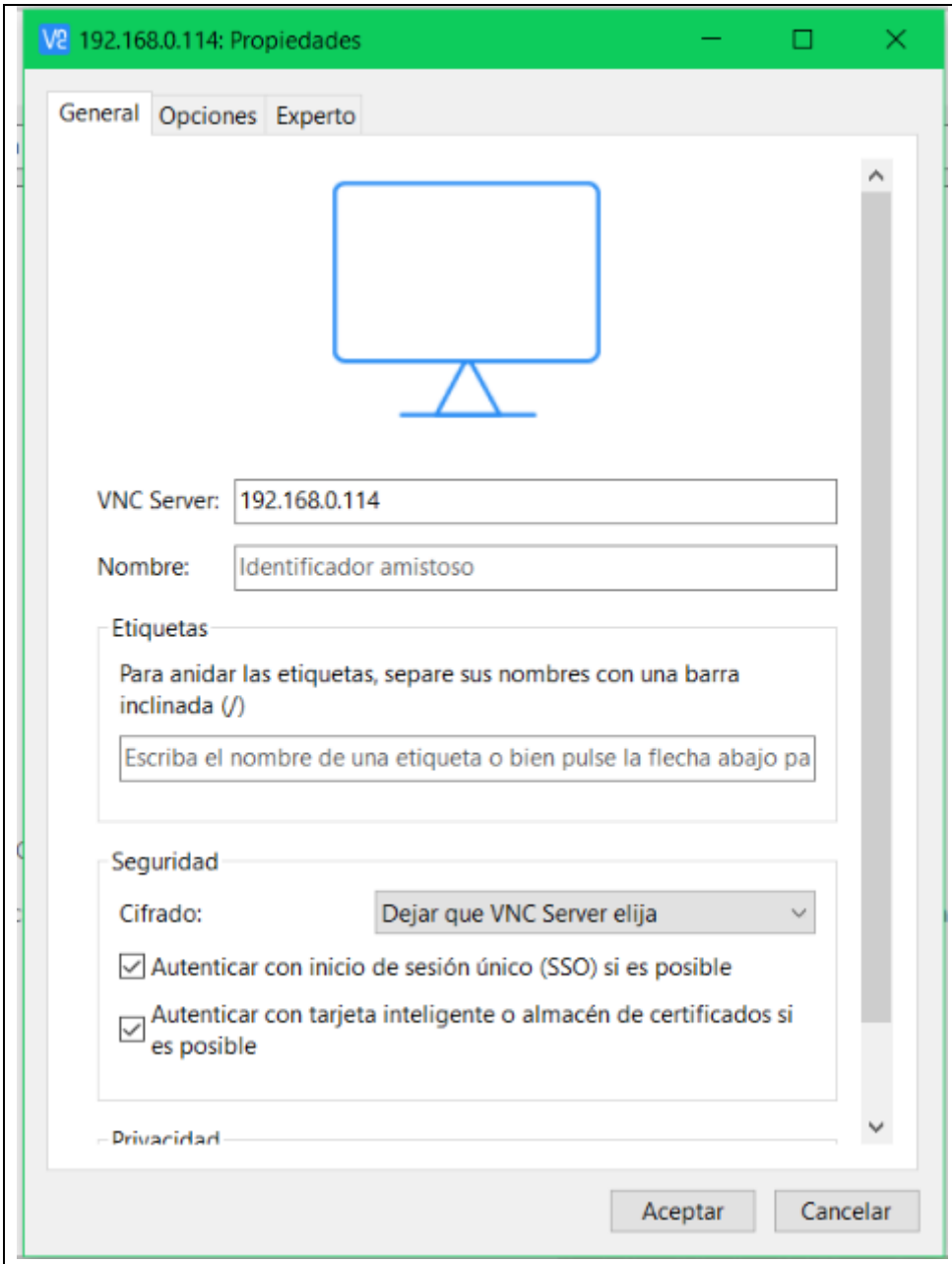

Propiedades
General
Opciones
Experto
VNC Server:
Dirección IP o nombre del host
Nombre:
Identificador amistoso
Etiquetas
Para anidar las etiquetas, separe sus nombres con una barra inclinada (/)
Escriba el nombre de una etiqueta o bien pulse la flecha abajo pa
Seguridad
Cifrado:
Dejar que VNC Server elija
Autenticar con inicio de sesión único (SSO) si es posible
Autenticar con tarjeta inteligente o almacén de certificados si es posible
Privacidad
Aceptar
Cancelar

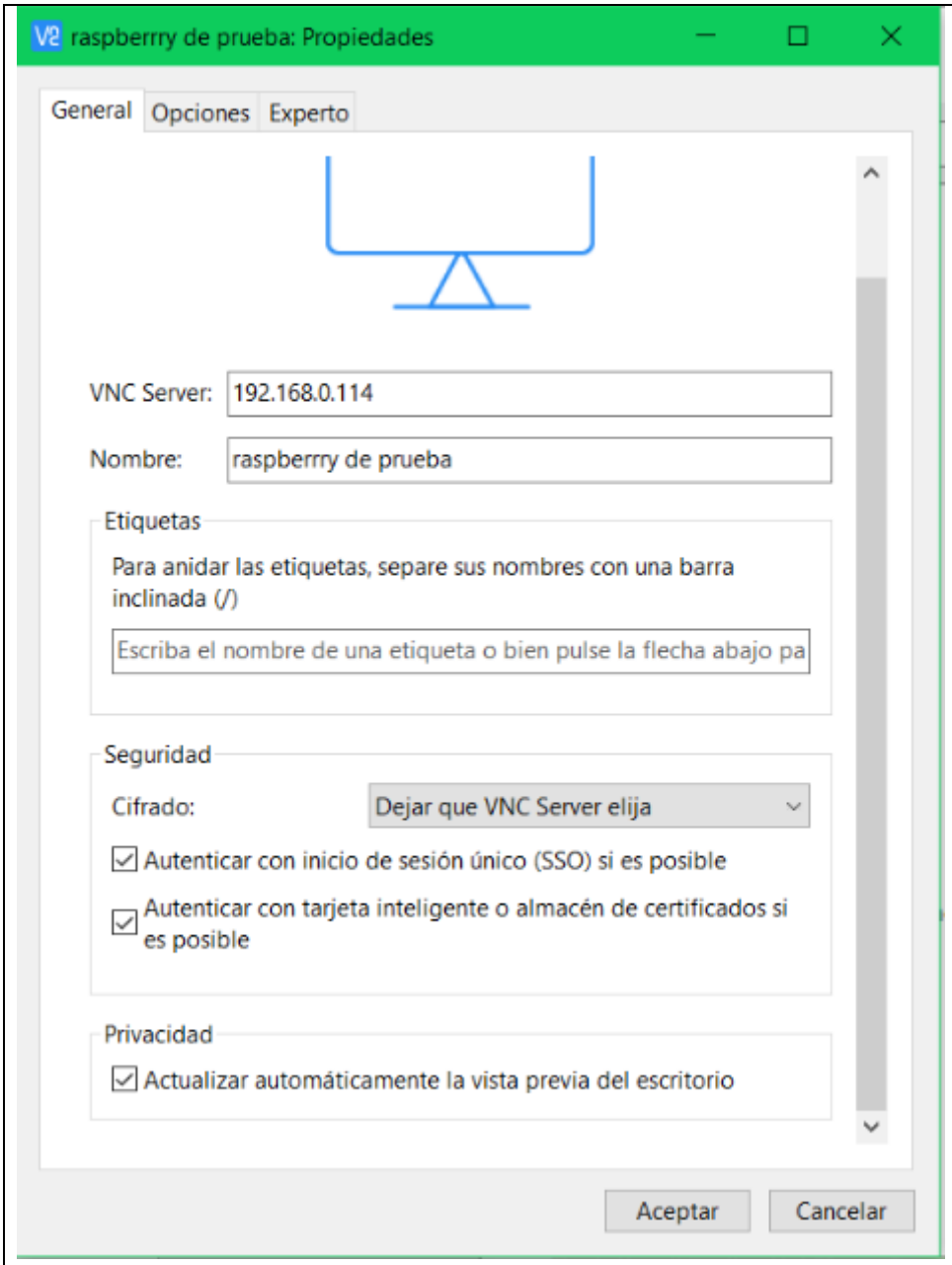

Ejecutar
Escriba el nombre del programa, carpeta, documento o recurso de Internet que desea abrir con Windows.
Abrir:
cmd
Aceptar
Cancelar
Examinar...

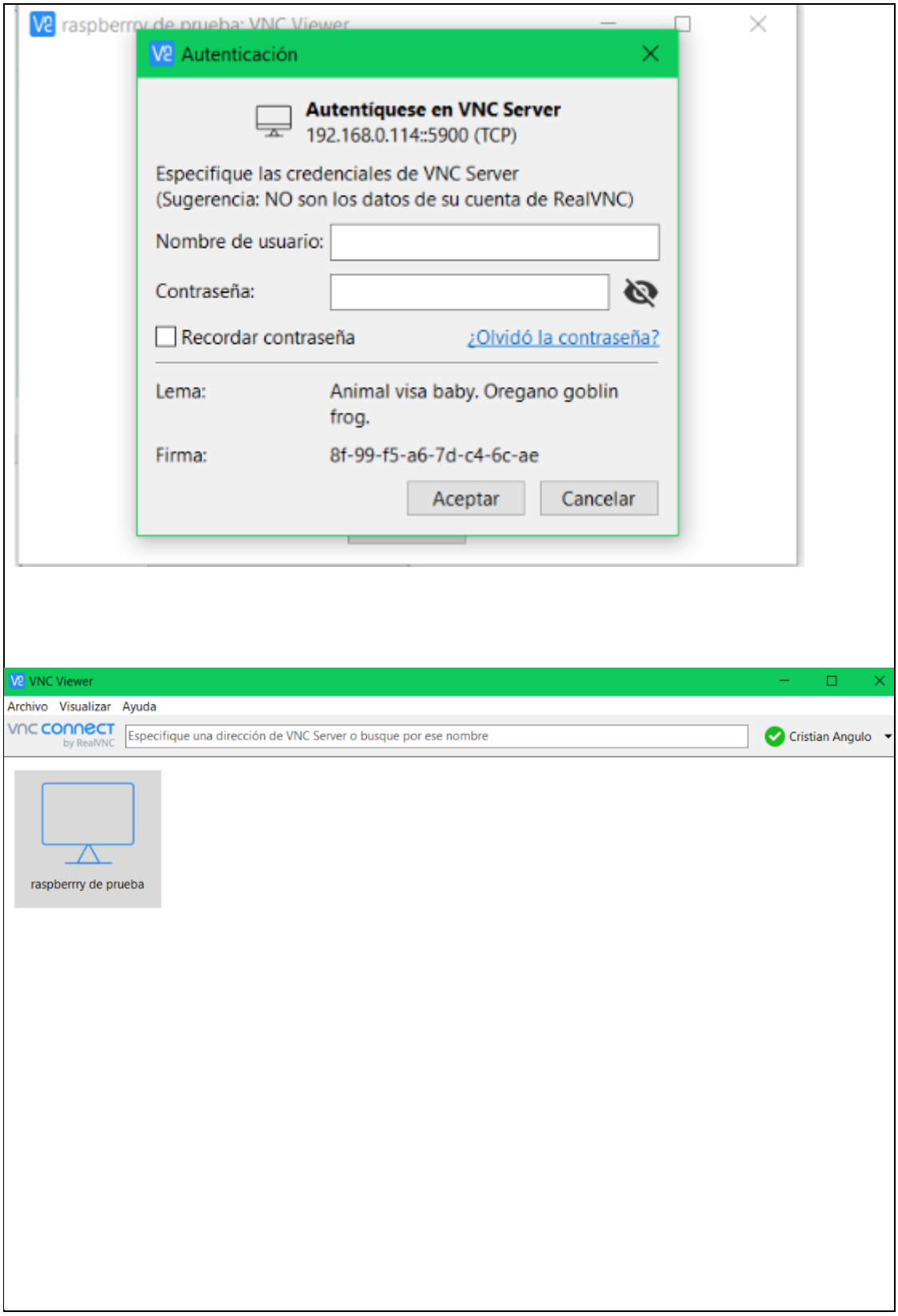

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2130]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\CLICK>
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2130]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\CLICK>ping raspberrypi.local -4

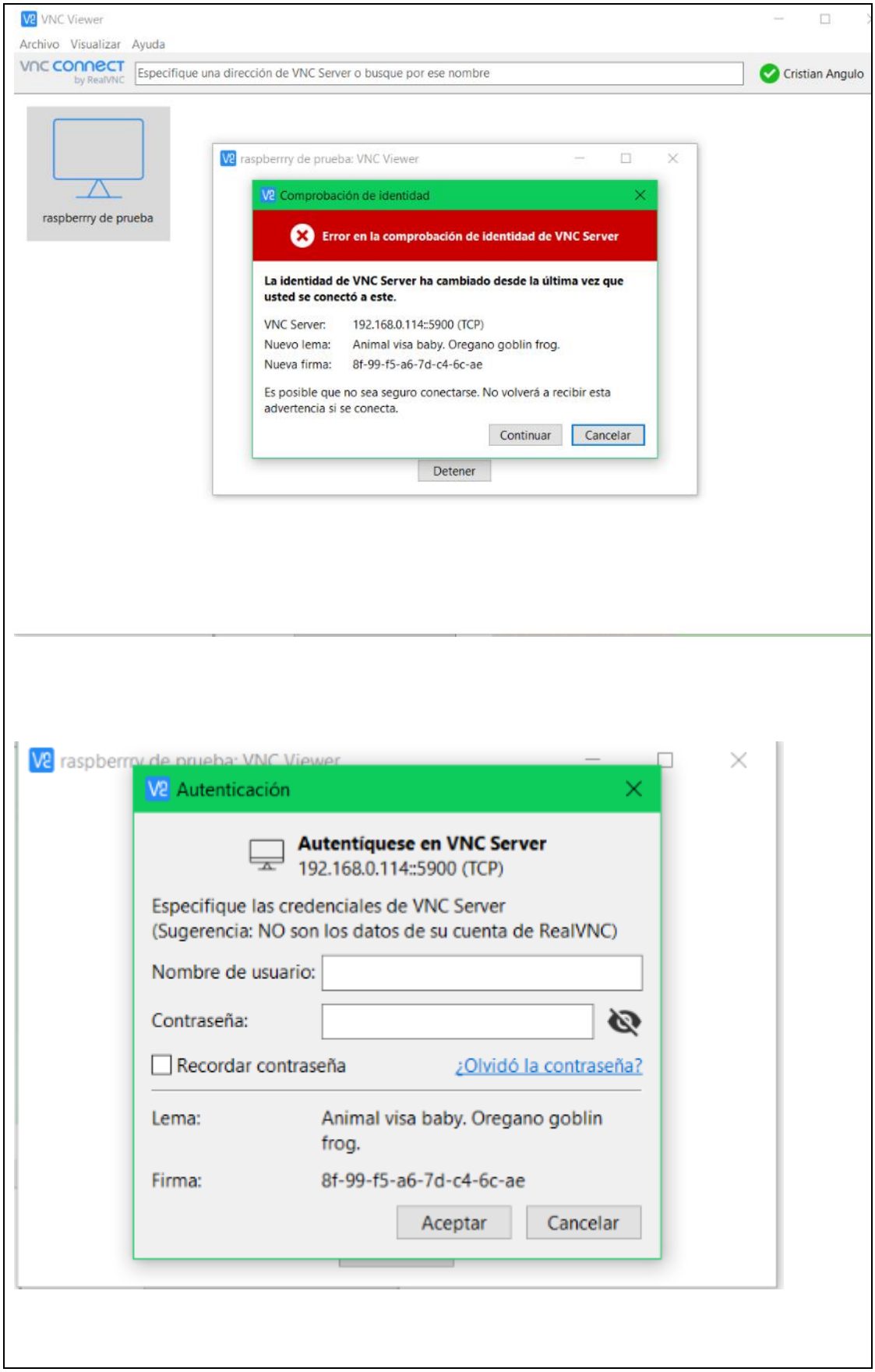

C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2130]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\CLICK>ping raspberrypi.local -4
Haciendo ping a raspberrypi.local [192.168.0.114] con 32 bytes de datos:
Respuesta desde 192.168.0.114: bytes=32 tiempo=9ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Estadísticas de ping para 192.168.0.114:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 9ms, Máximo = 10ms, Media = 9ms
C:\Users\CLICK>

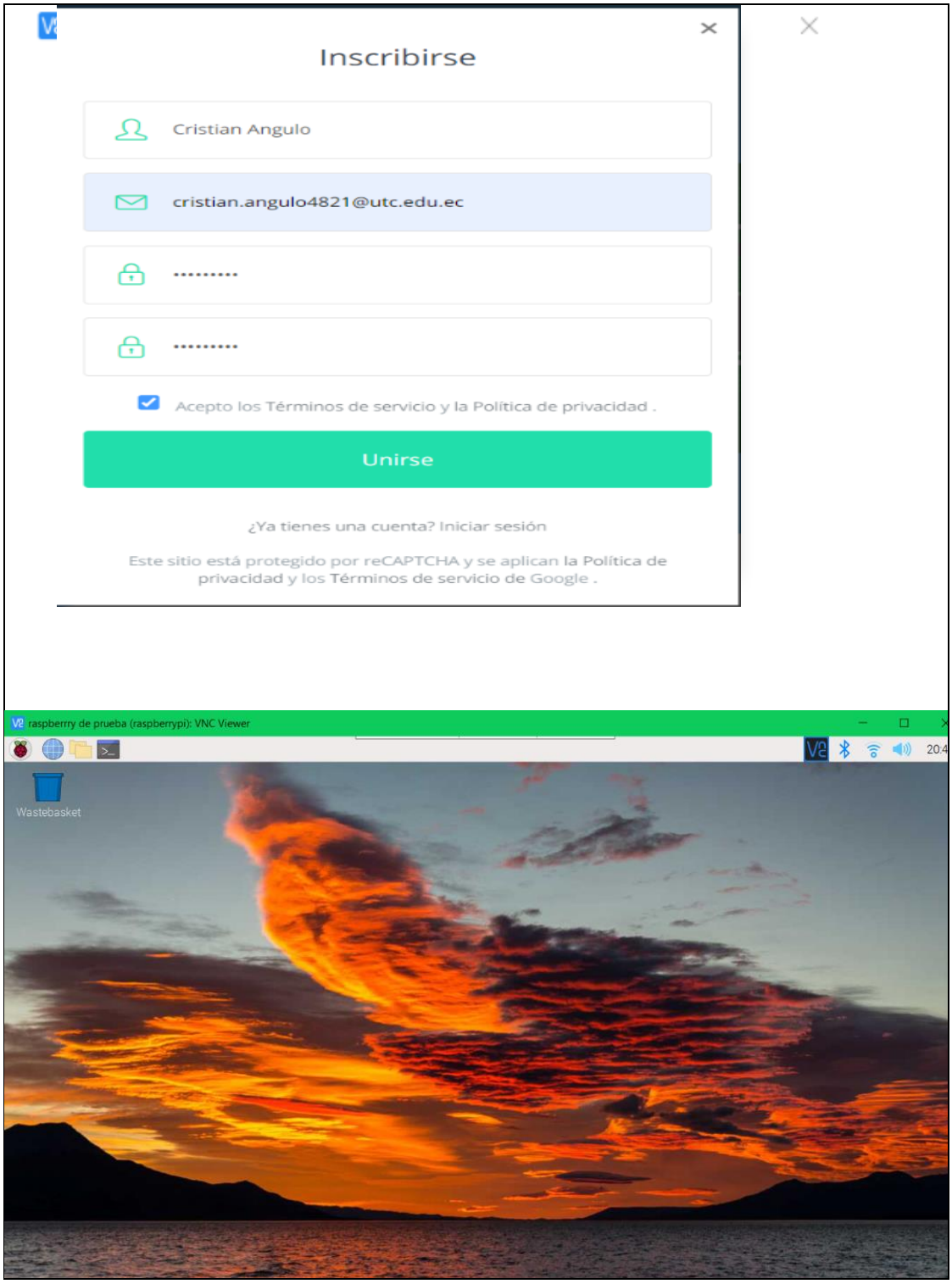

Seleccionar C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2130]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\CLICK>ping raspberrypi.local -4
Haciendo ping a raspberrypi.local [192.168.0.114] con 32 bytes de datos:
Respuesta desde 192.168.0.114: bytes=32 tiempo=9ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Respuesta desde 192.168.0.114: bytes=32 tiempo=10ms TTL=64
Estadísticas de ping para 192.168.0.114:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 9ms, Máximo = 10ms, Media = 9ms
C:\Users\CLICK>

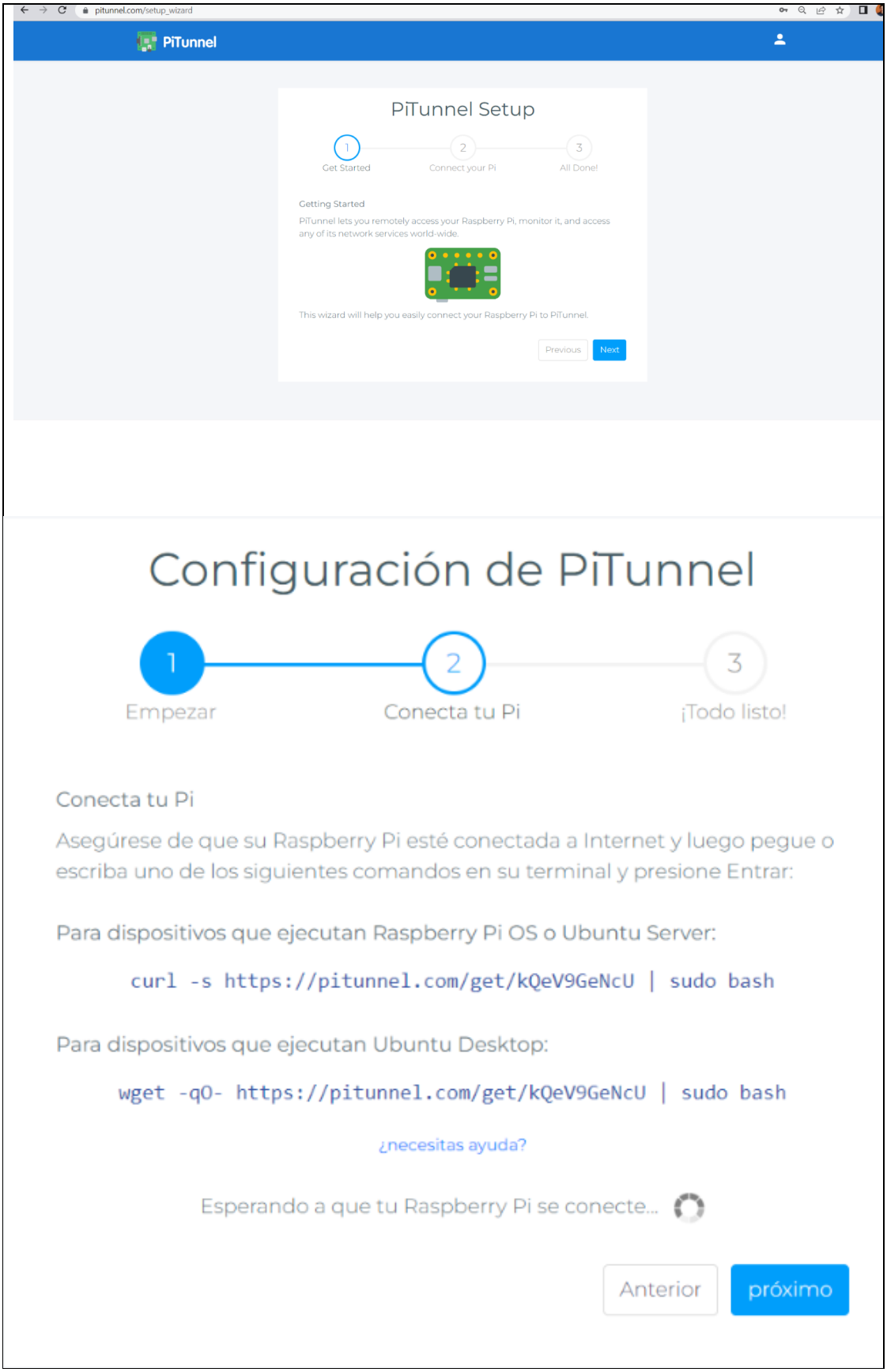

192.168.0.114: Propiedades
General
Opciones
Experto
VNC Server:
192.168.0.114
Nombre:
Identificador amistoso
Etiquetas
Para anidar las etiquetas, separe sus nombres con una barra inclinada (/)
Escriba el nombre de una etiqueta o bien pulse la flecha abajo pa
Seguridad
Cifrado:
Dejar que VNC Server elija
Autenticar con inicio de sesión único (SSO) si es posible
Autenticar con tarjeta inteligente o almacén de certificados si es posible
Privacidad
Aceptar
Cancelar

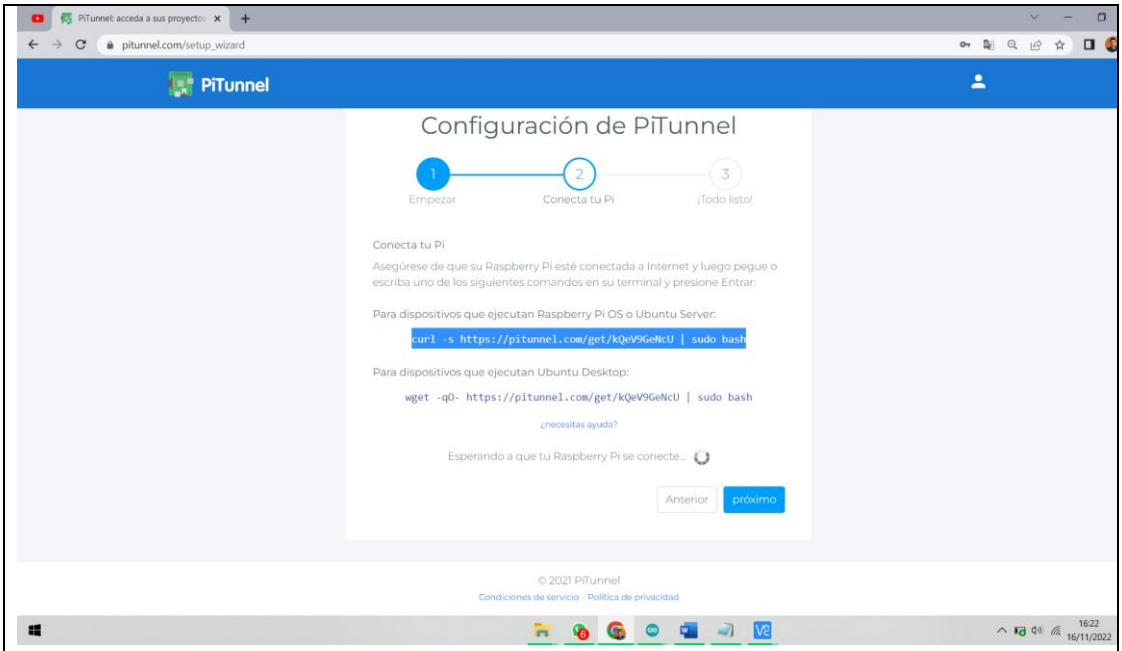

raspberrry de prueba: Propiedades
General
Opciones
Experto
VNC Server:
192.168.0.114
Nombre:
raspberrry de prueba
Etiquetas
Para anidar las etiquetas, separe sus nombres con una barra inclinada (/)
Escriba el nombre de una etiqueta o bien pulse la flecha abajo pa
Seguridad
Cifrado:
Dejar que VNC Server elija
Autenticar con inicio de sesión único (SSO) si es posible
Autenticar con tarjeta inteligente o almacén de certificados si es posible
Privacidad
Actualizar automáticamente la vista previa del escritorio
Aceptar
Cancelar

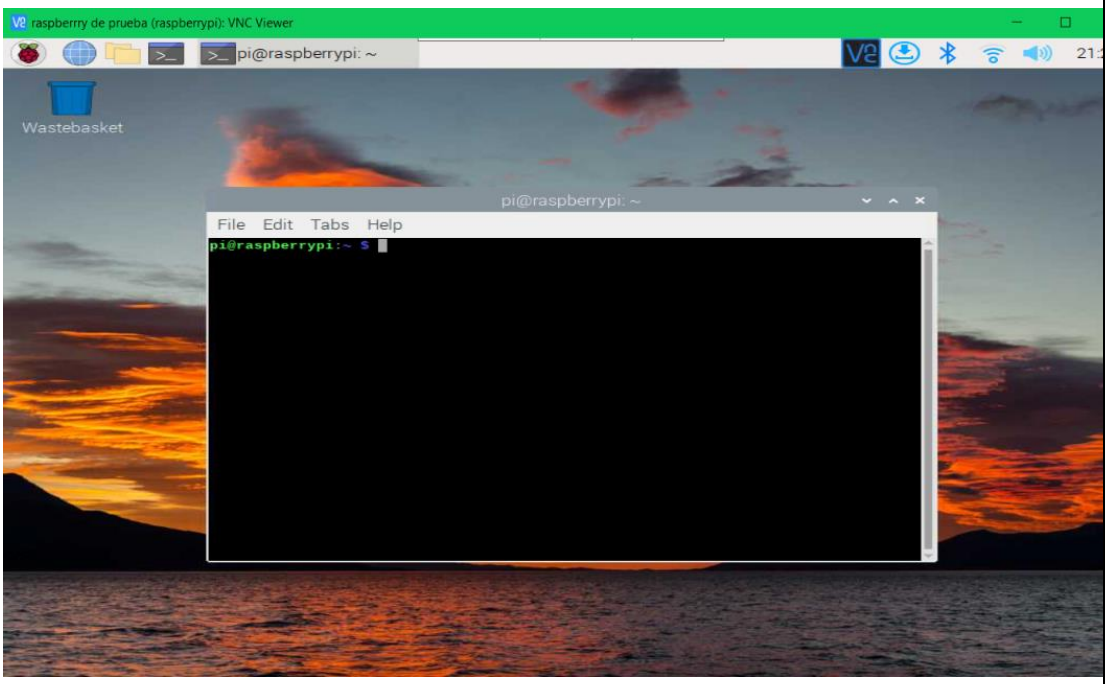

raspberrry de prueba: VNC Viewer
Autenticación
Autentíquese en VNC Server
192.168.0.114::5900 (TCP)
Especifique las credenciales de VNC Server
(Sugerencia: NO son los datos de su cuenta de RealVNC)
Nombre de usuario:
Contraseña:
Recordar contraseña
¿Olvidó la contraseña?
Lema:
Animal visa baby. Oregano goblin frog.
Firma:
8f-99-f5-a6-7d-c4-6c-ae
Aceptar
Cancelar
VNC Viewer
Archivo Visualizar Ayuda
vnc connect by RealVNC
Especifique una dirección de VNC Server o busque por ese nombre
Cristian Angulo
raspberrry de prueba

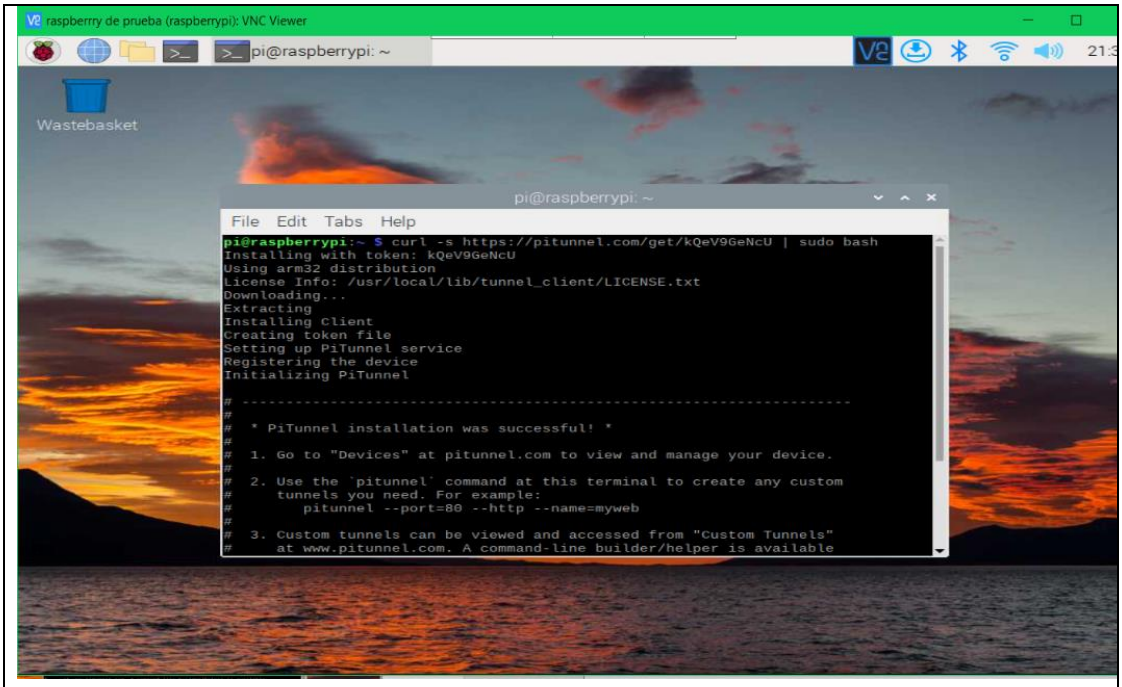

VNC Viewer
Archivo Visualizar Ayuda
vnc connect by RealVNC
Especifique una dirección de VNC Server o busque por ese nombre
Cristian Angulo
raspberrry de prueba
raspberrry de prueba: VNC Viewer
Comprobación de identidad
Error en la comprobación de identidad de VNC Server
La identidad de VNC Server ha cambiado desde la última vez que usted se conectó a este.
VNC Server: 192.168.0.114::5900 (TCP)
Nuevo lema: Animal visa baby. Oregano goblin frog.
Nueva firma: 8f-99-f5-a6-7d-c4-6c-ae
Es posible que no sea seguro conectarse. No volverá a recibir esta advertencia si se conecta.
Continuar
Cancelar
Detener
Autenticación
Autentíquese en VNC Server
192.168.0.114::5900 (TCP)
Especifique las credenciales de VNC Server
(Sugerencia: NO son los datos de su cuenta de RealVNC)
Nombre de usuario:
Contraseña:
Recordar contraseña
¿Olvidó la contraseña?
Lema: Animal visa baby. Oregano goblin frog.
Firma: 8f-99-f5-a6-7d-c4-6c-ae
Aceptar
Cancelar

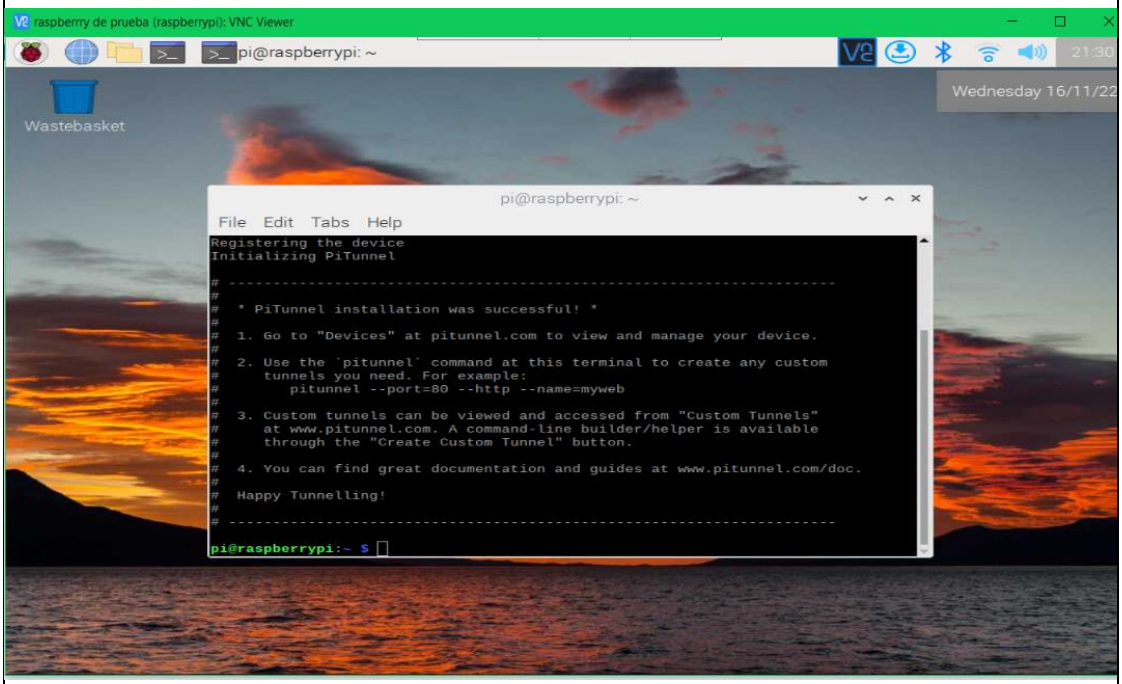

Inscribirse
Cristian Angulo
cristian.angulo4821@utc.edu.ec
Acepto los Términos de servicio y la Política de privacidad .
Unirse
¿Ya tienes una cuenta? Iniciar sesión
Este sitio está protegido por reCAPTCHA y se aplican la Política de privacidad y los Términos de servicio de Google .
raspberrry de prueba (raspberrypi): VNC Viewer
Wastebasket

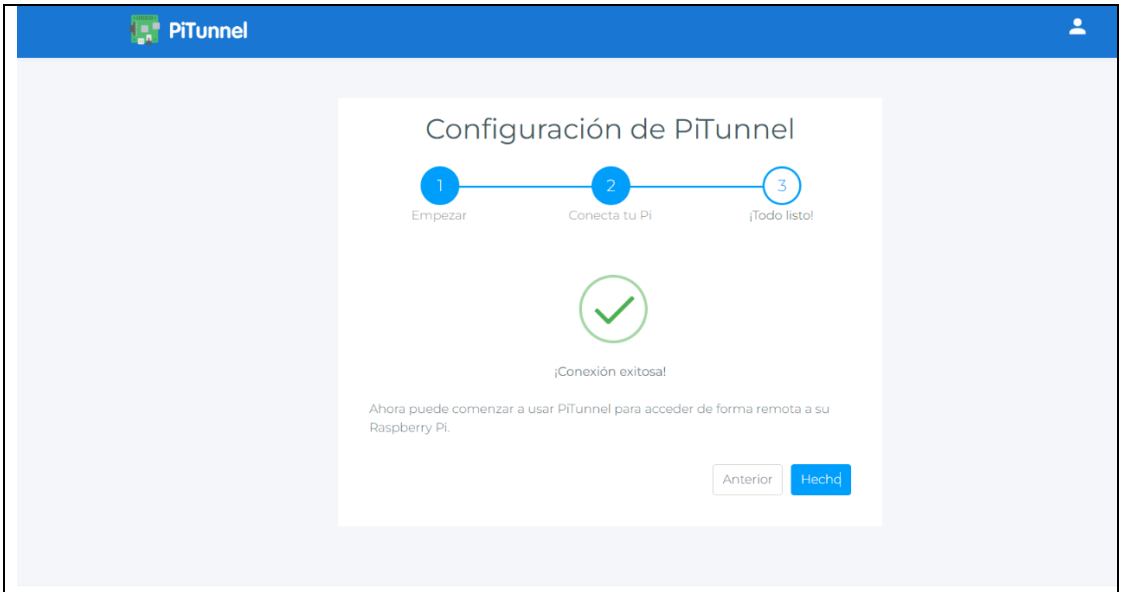

pitunnel.com/setup_wizard
PiTunnel
PiTunnel Setup
1
Get Started
2
Connect your Pi
3
All Done!
Getting Started
PiTunnel lets you remotely access your Raspberry Pi, monitor it, and access any of its network services world-wide.
This wizard will help you easily connect your Raspberry Pi to PiTunnel.
Previous
Next
Configuración de PiTunnel
1
Empezar
2
Conecta tu Pi
3
¡Todo listo!
Conecta tu Pi
Asegúrese de que su Raspberry Pi esté conectada a Internet y luego pegue o escriba uno de los siguientes comandos en su terminal y presione Entrar:
Para dispositivos que ejecutan Raspberry Pi OS o Ubuntu Server:
curl -s https://pitunnel.com/get/kQeV9GeNcU | sudo bash
Para dispositivos que ejecutan Ubuntu Desktop:
wget -qO- https://pitunnel.com/get/kQeV9GeNcU | sudo bash
¿necesitas ayuda?
Esperando a que tu Raspberry Pi se conecte...
Anterior
próximo


PiTunnel: acceda a sus proyectos
pitunnel.com/setup_wizard
PiTunnel
Configuración de PiTunnel
1
Empezar
2
Conecta tu Pi
3
¡Todo listo!
Conecta tu Pi
Asegúrese de que su Raspberry Pi esté conectada a Internet y luego pegue o escriba uno de los siguientes comandos en su terminal y presione Entrar:
Para dispositivos que ejecutan Raspberry Pi OS o Ubuntu Server:
curl -s https://pitunnel.com/get/kQeV9GeNcU | sudo bash
Para dispositivos que ejecutan Ubuntu Desktop:
wget -qO- https://pitunnel.com/get/kQeV9GeNcU | sudo bash
¿necesitas ayuda?
Esperando a que tu Raspberry Pi se conecte...
Anterior
próximo
© 2021 PiTunnel
Condiciones de servicio - Política de privacidad
16:22
16/11/2022

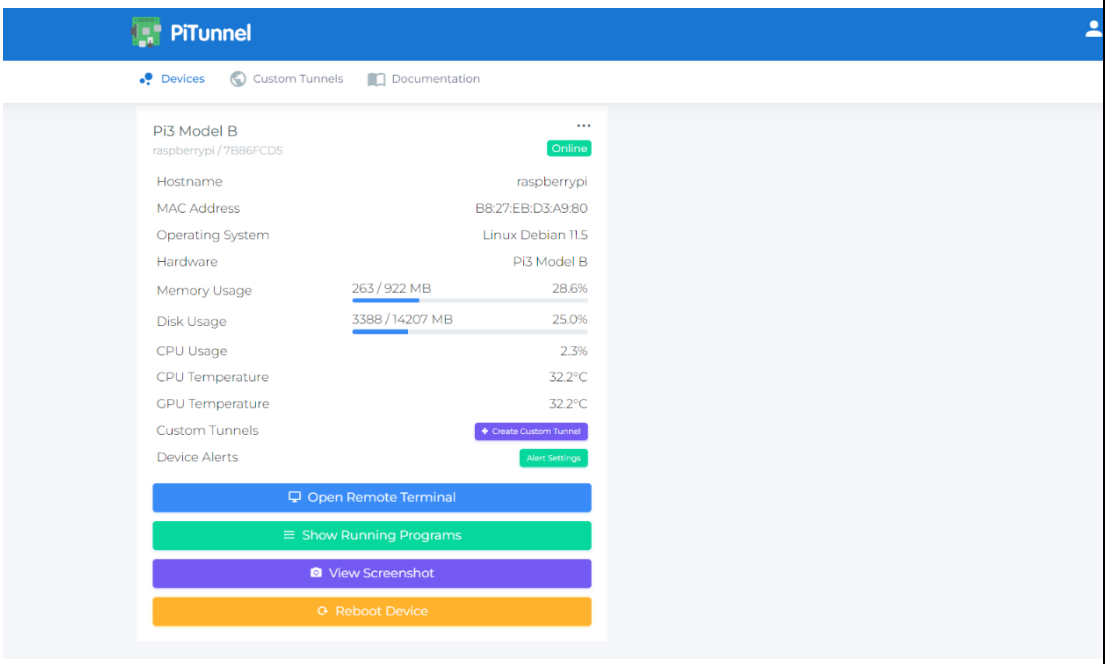

raspberry de prueba (raspberrypi): VNC Viewer
pi@raspberrypi: ~
Wastebasket
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $


raspberrry de prueba (raspberrypi): VNC Viewer
pi@raspberrypi: ~
Wastebasket
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ curl -s https://pitunnel.com/get/kQeV9GeNcU | sudo bash
Installing with token: kQeV9GeNcU
Using arm32 distribution
License Info: /usr/local/lib/tunnel_client/LICENSE.txt
Downloading...
Extracting
Installing client
Creating token file
Setting up PiTunnel service
Registering the device
Initializing PiTunnel
* PiTunnel installation was successful! *
1. Go to "Devices" at pitunnel.com to view and manage your device.
2. Use the `pitunnel` command at this terminal to create any custom
tunnels you need. For example:
pitunnel --port=80 --http --name=myweb
3. Custom tunnels can be viewed and accessed from "Custom Tunnels"
at www.pitunnel.com. A command-line builder/helper is available

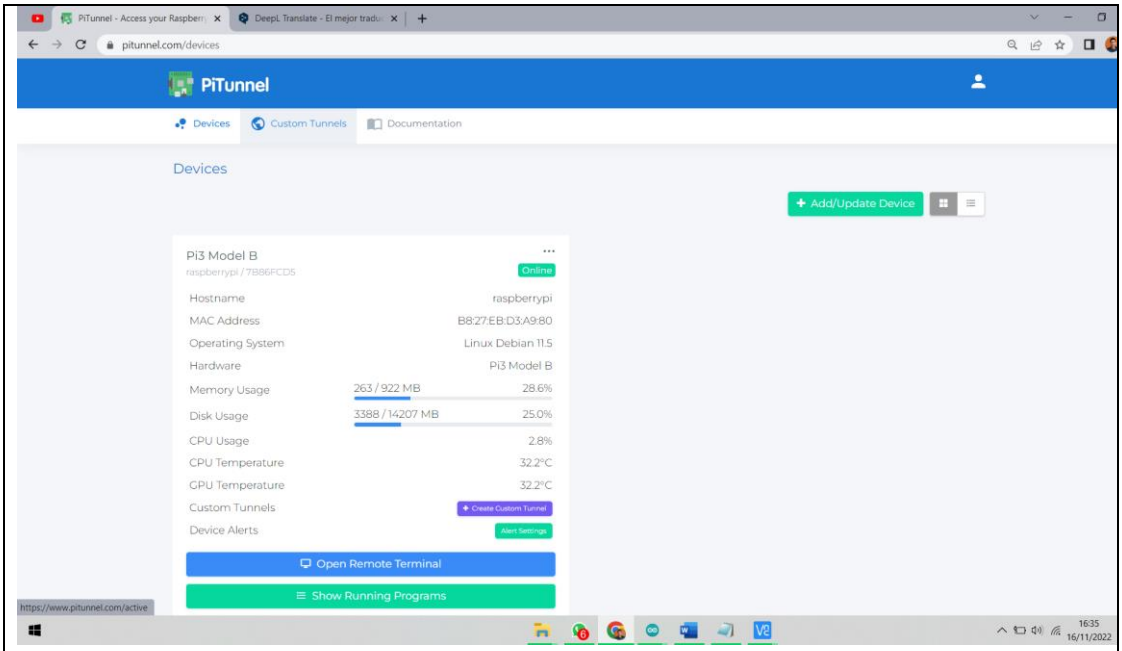

raspberrry de prueba (raspberrypi): VNC Viewer
pi@raspberrypi: ~
21:30
Wednesday 16/11/22
Wastebasket
pi@raspberrypi: ~
File Edit Tabs Help
Registering the device
Initializing PiTunnel
* PiTunnel installation was successful! *
1. Go to "Devices" at pitunnel.com to view and manage your device.
2. Use the `pitunnel` command at this terminal to create any custom
tunnels you need. For example:
pitunnel --port=80 --http --name=myweb
3. Custom tunnels can be viewed and accessed from "Custom Tunnels"
at www.pitunnel.com. A command-line builder/helper is available
through the "Create Custom Tunnel" button.
4. You can find great documentation and guides at www.pitunnel.com/doc.
Happy Tunnelling!
pi@raspberrypi:~ $



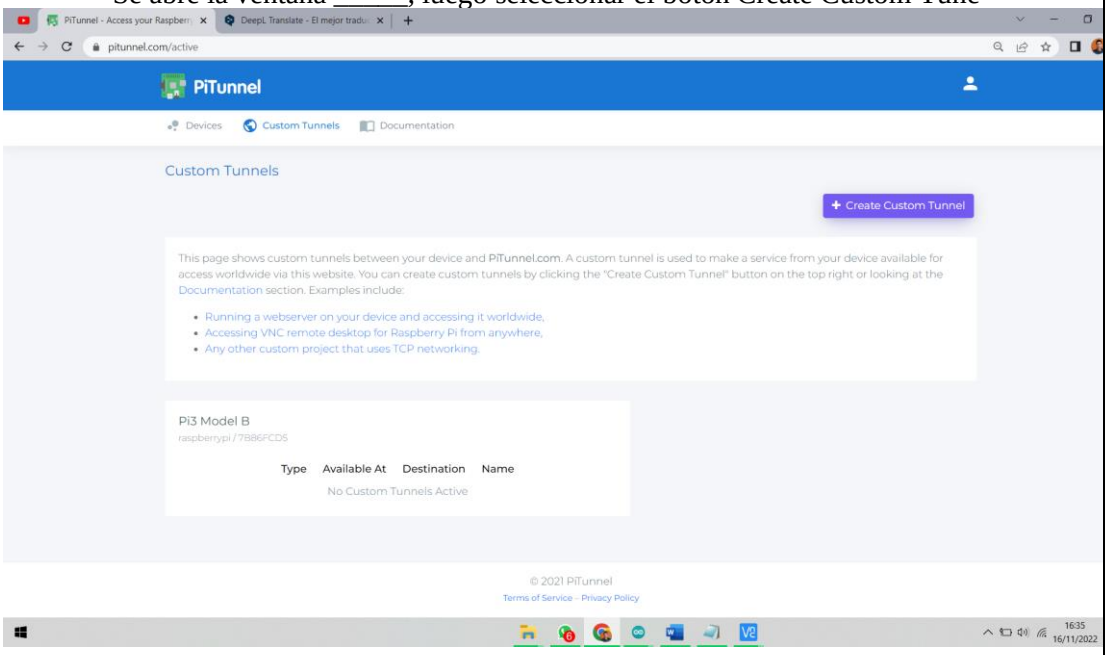
esta



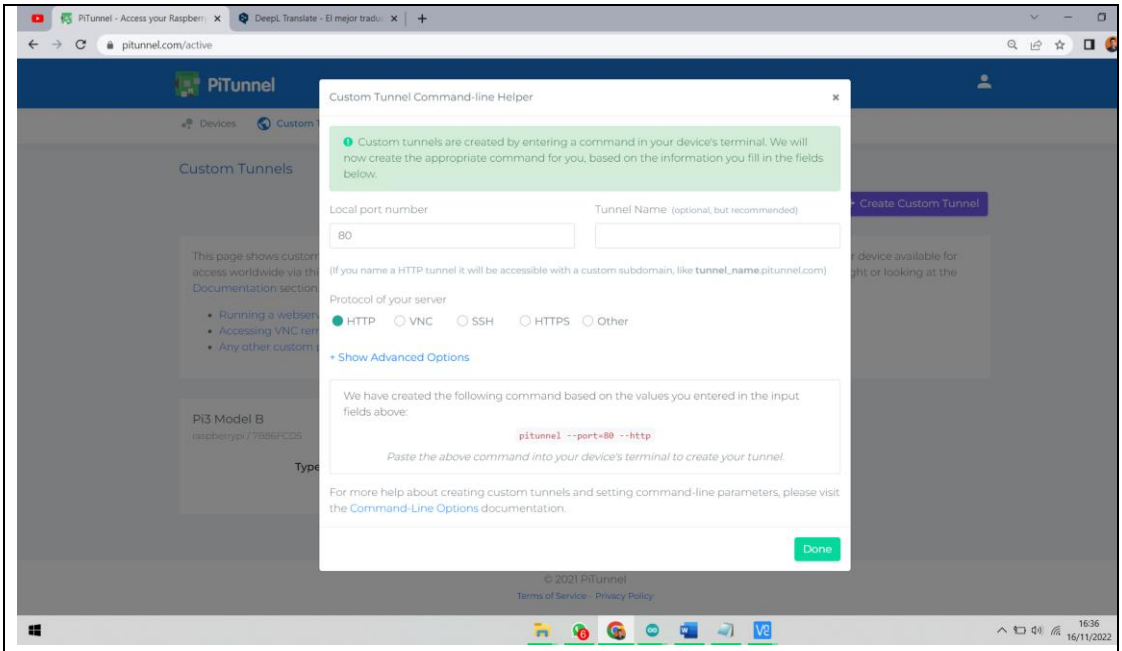
Seleccionamos CUSTOM TUNNEL



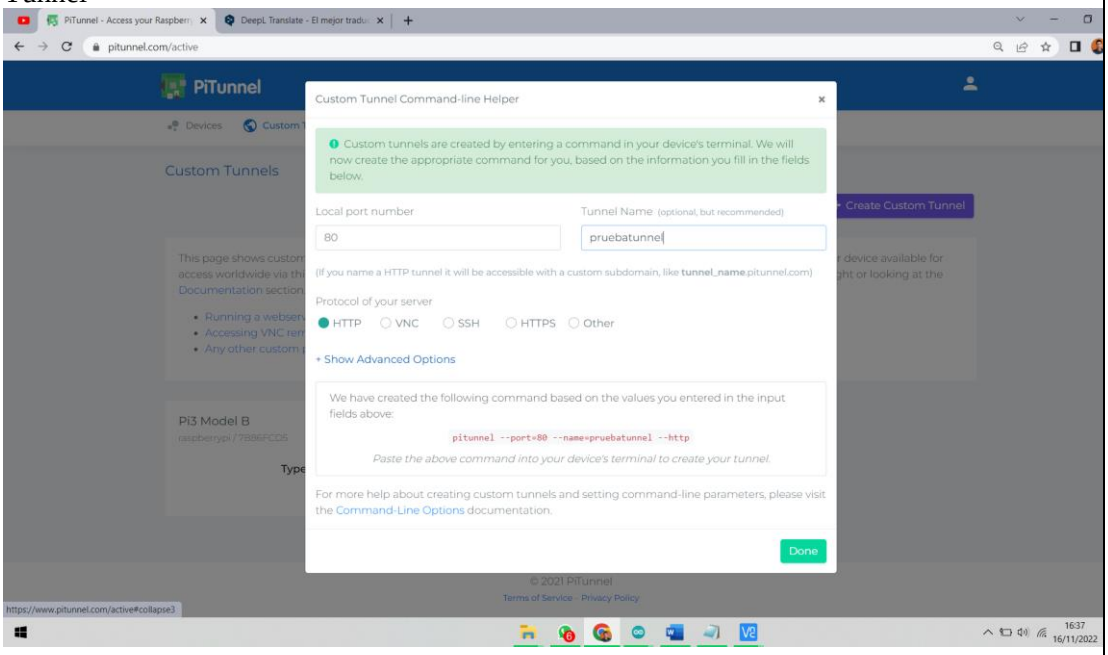
Se abre la ventana _____, luego seleccionar el botón Create Custom Tune



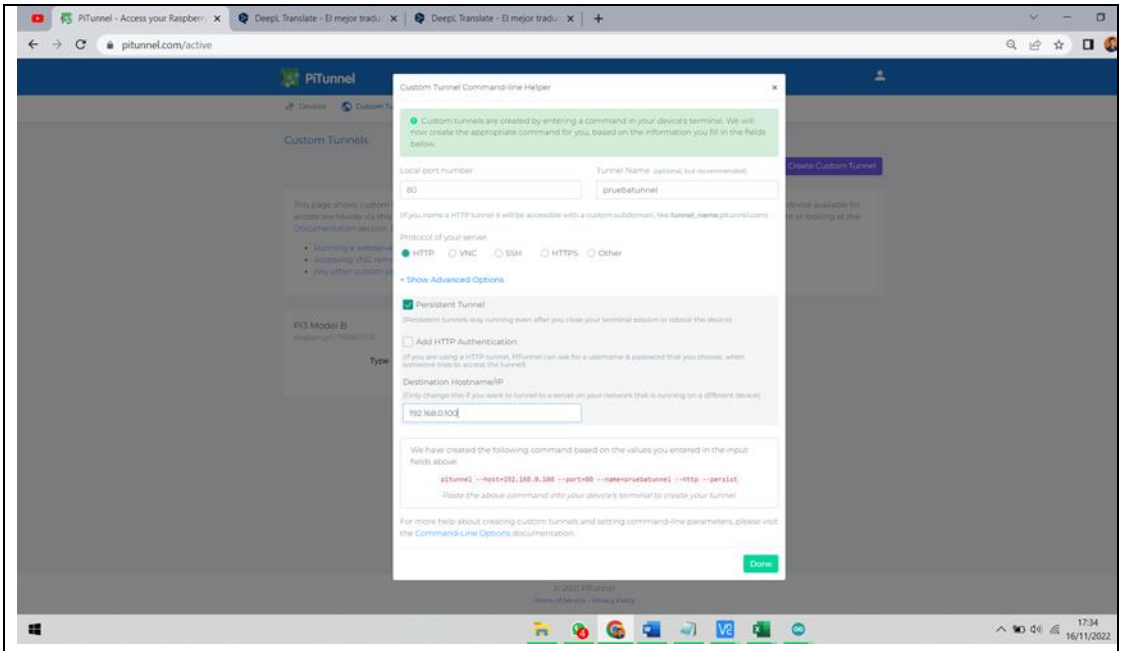
Se abre la ventana Custom Tunnel Command-line Helper en donde:
aparece seleccionado por defecto HTTP
Y en Local port number está previsto con 80.



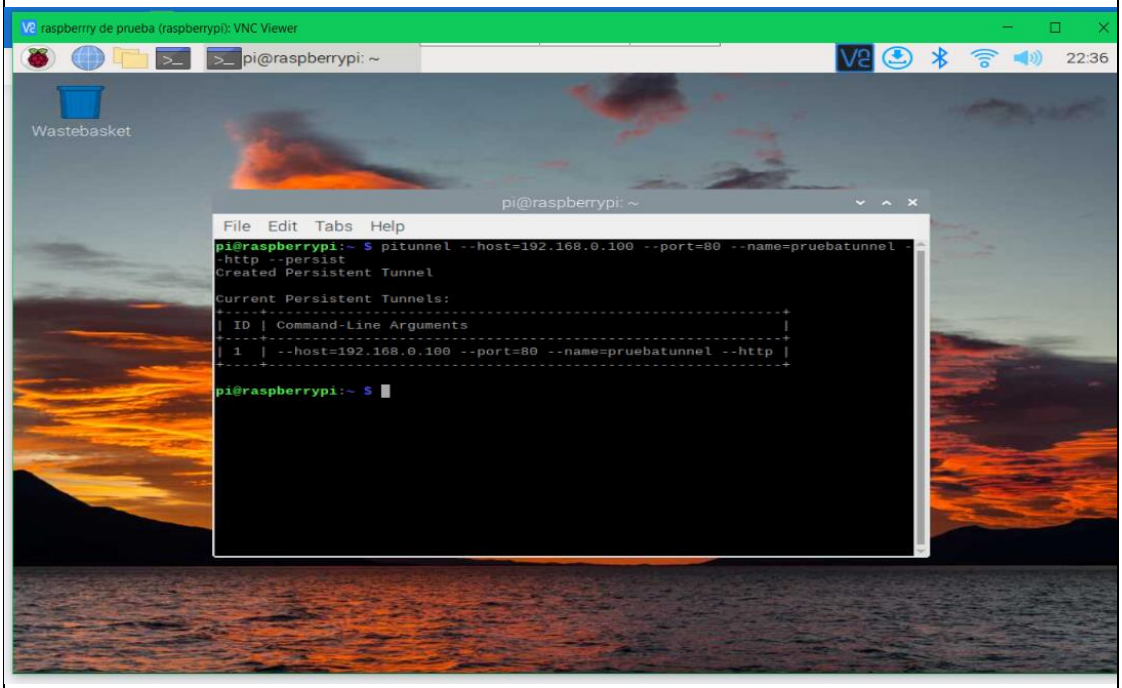
En el apartado de Tunnel Name (optional, bud recommended), agregar el nombre del Tunnel



Escribir la dirección IP en HostName IP

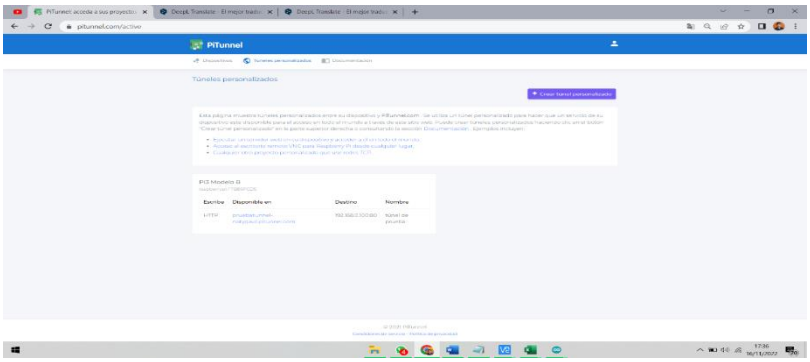


Se pega el comando de salida:

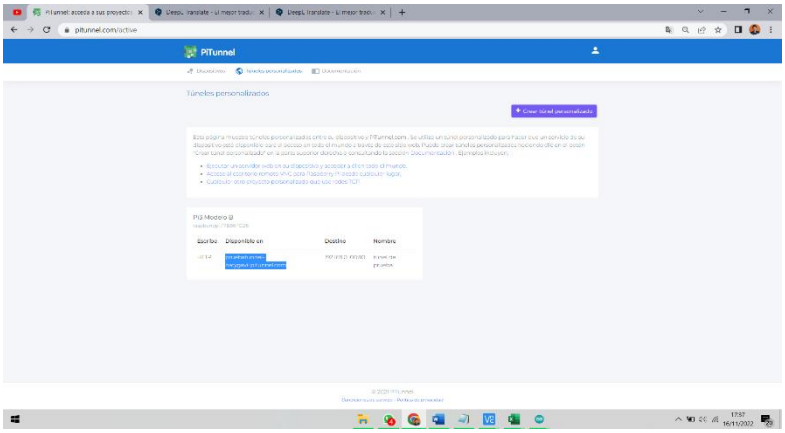


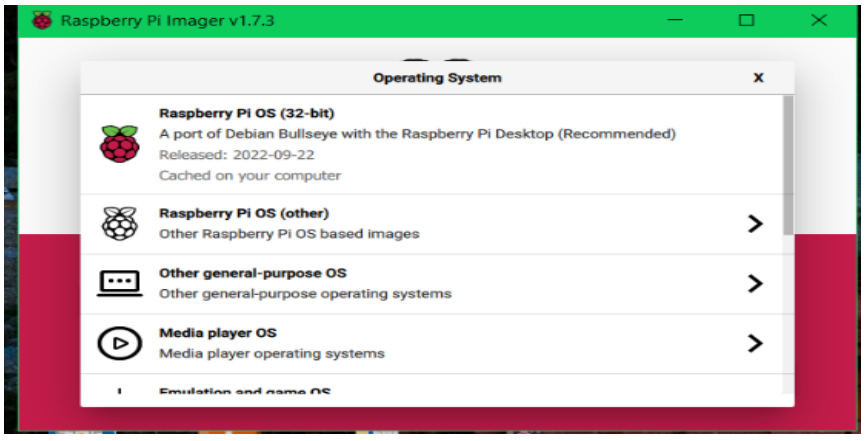
Aparece la pantalla

Seleccionar La URL que permite ver las cámaras desde cualquier ubicación.
El Pi Tunnel da una URL y agregar al final de /mjpeg/1

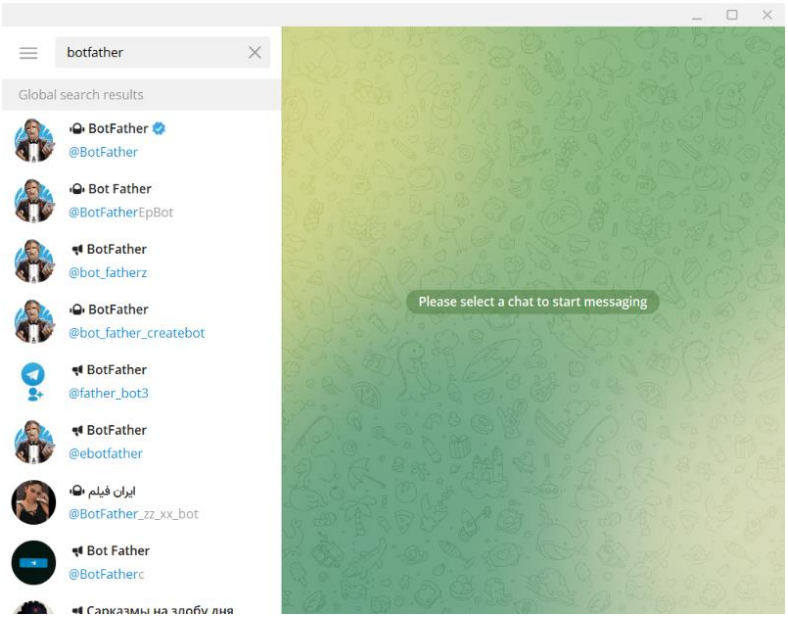


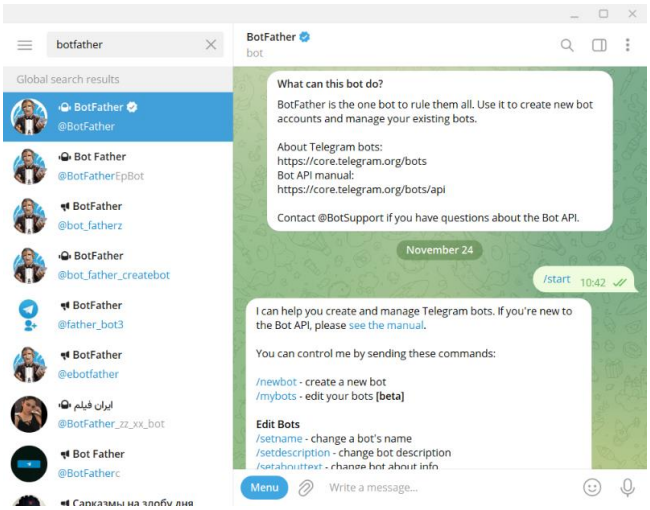
pruebatunnel-natygavil.pitunnel.com/mjpeg/1

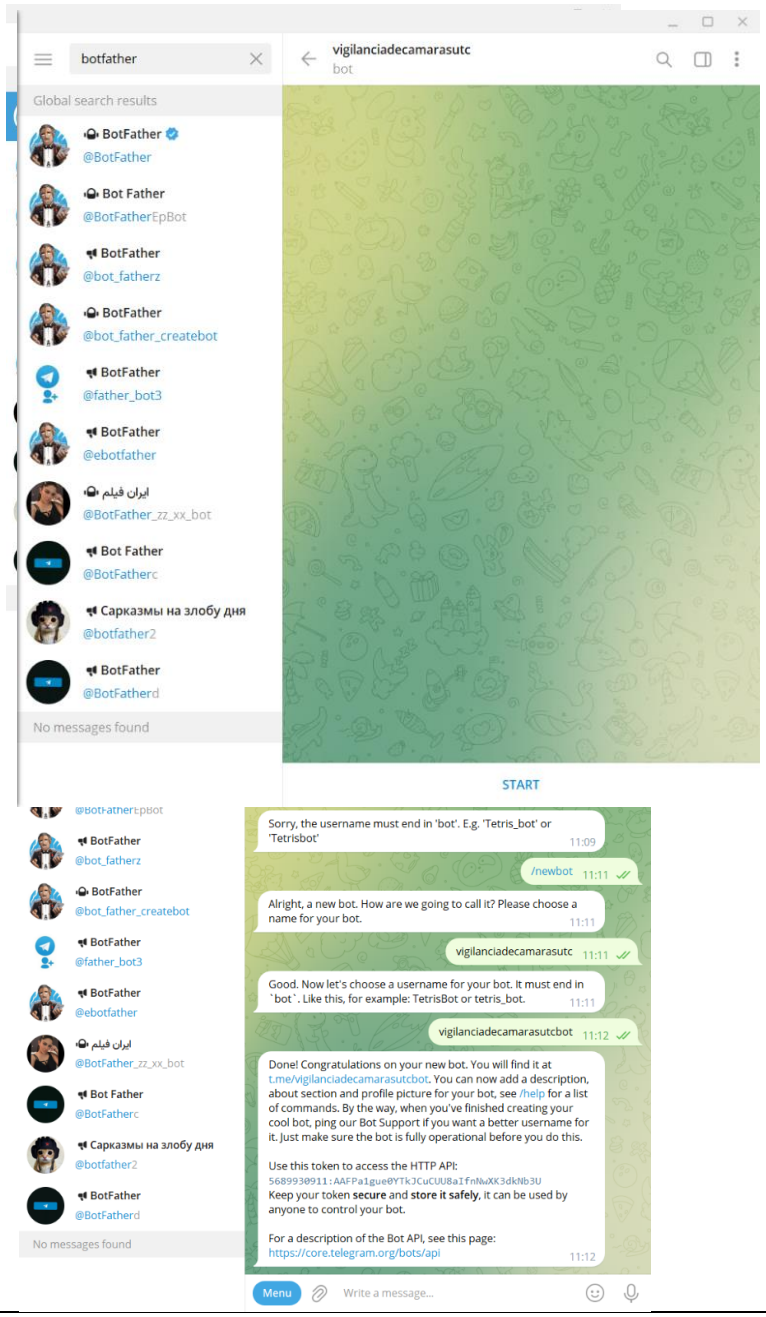


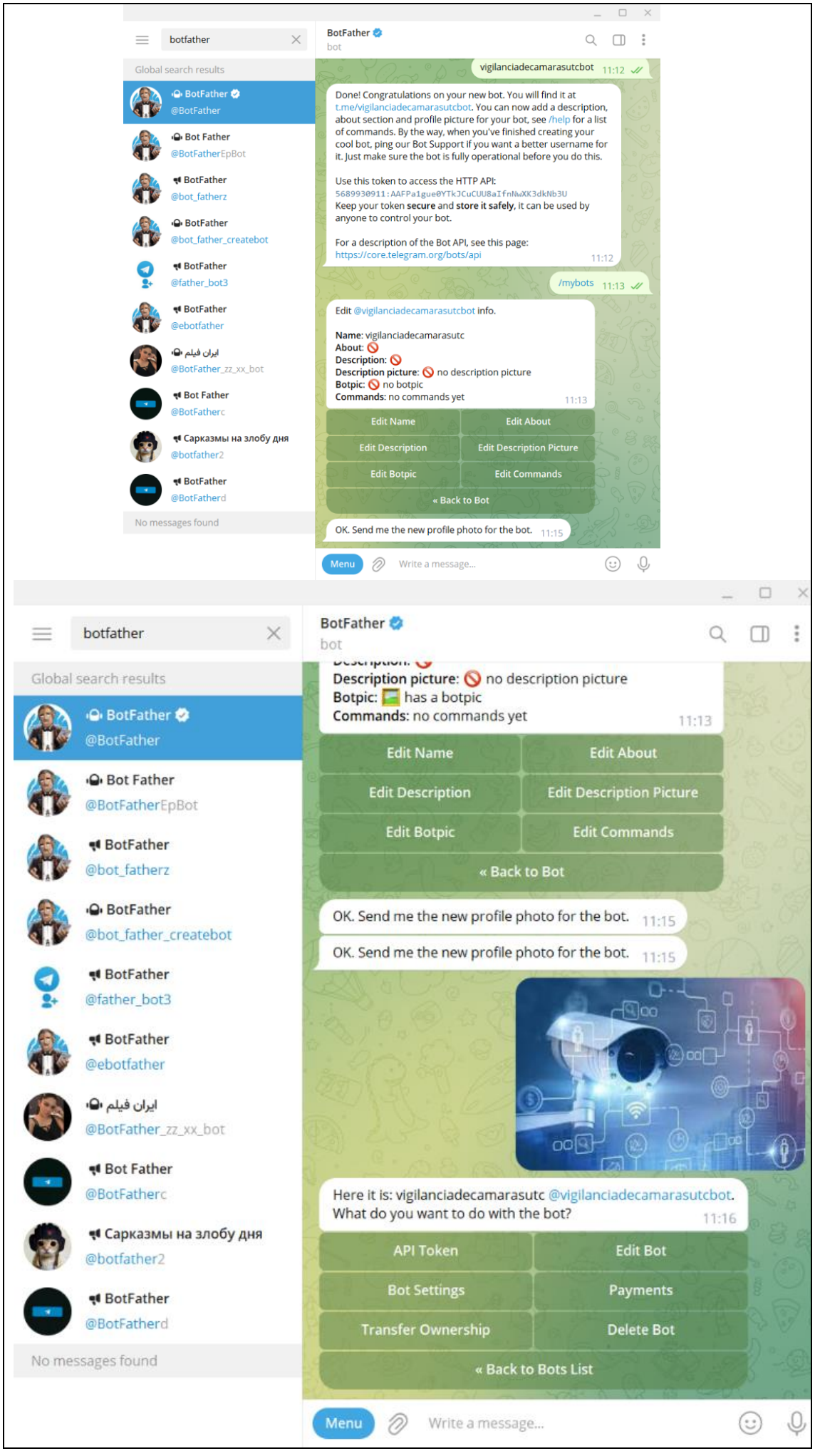


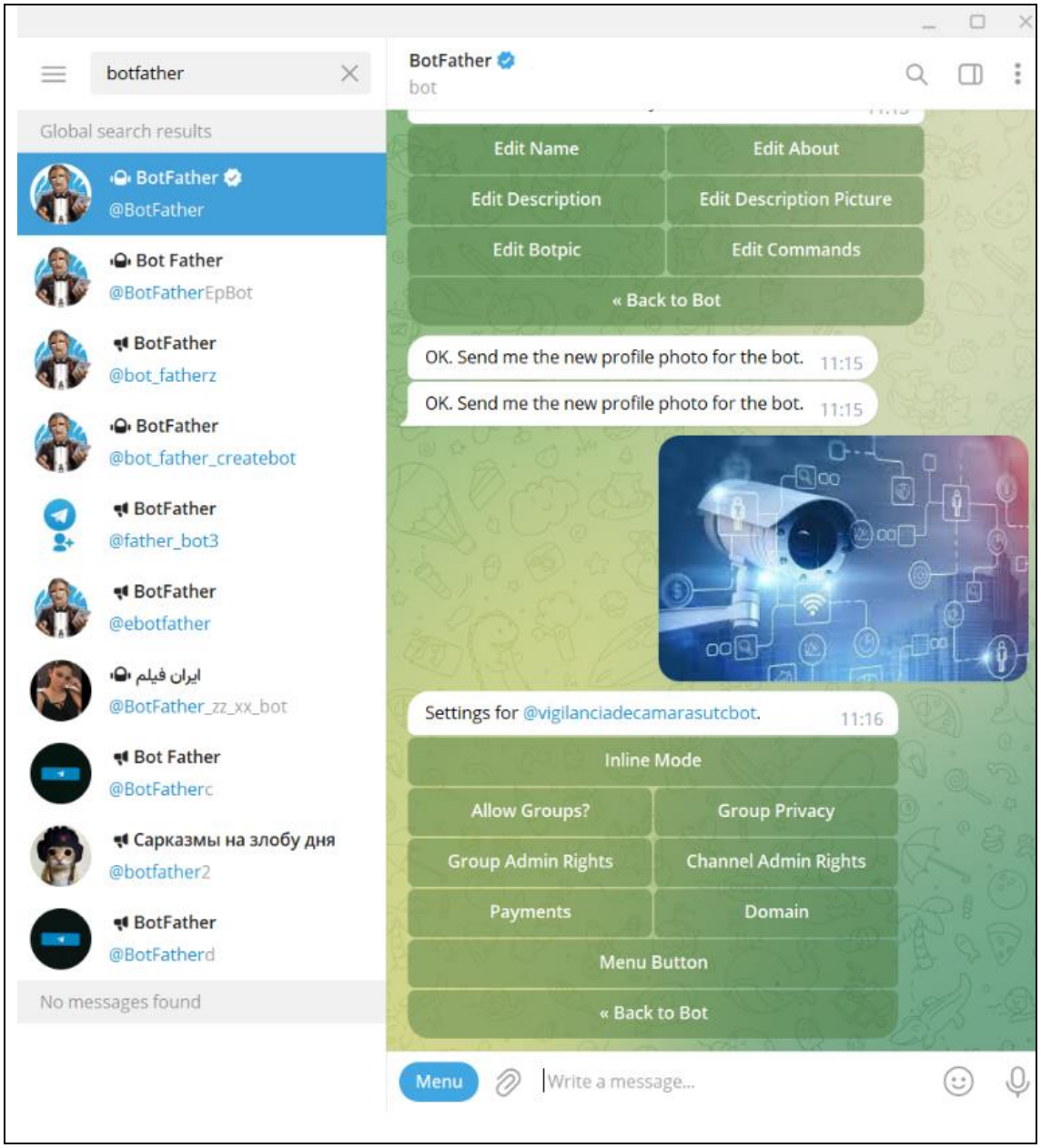
ANEXO II	GUIA PARA CREAR EL BOT MEDIANTE TELEGRAM

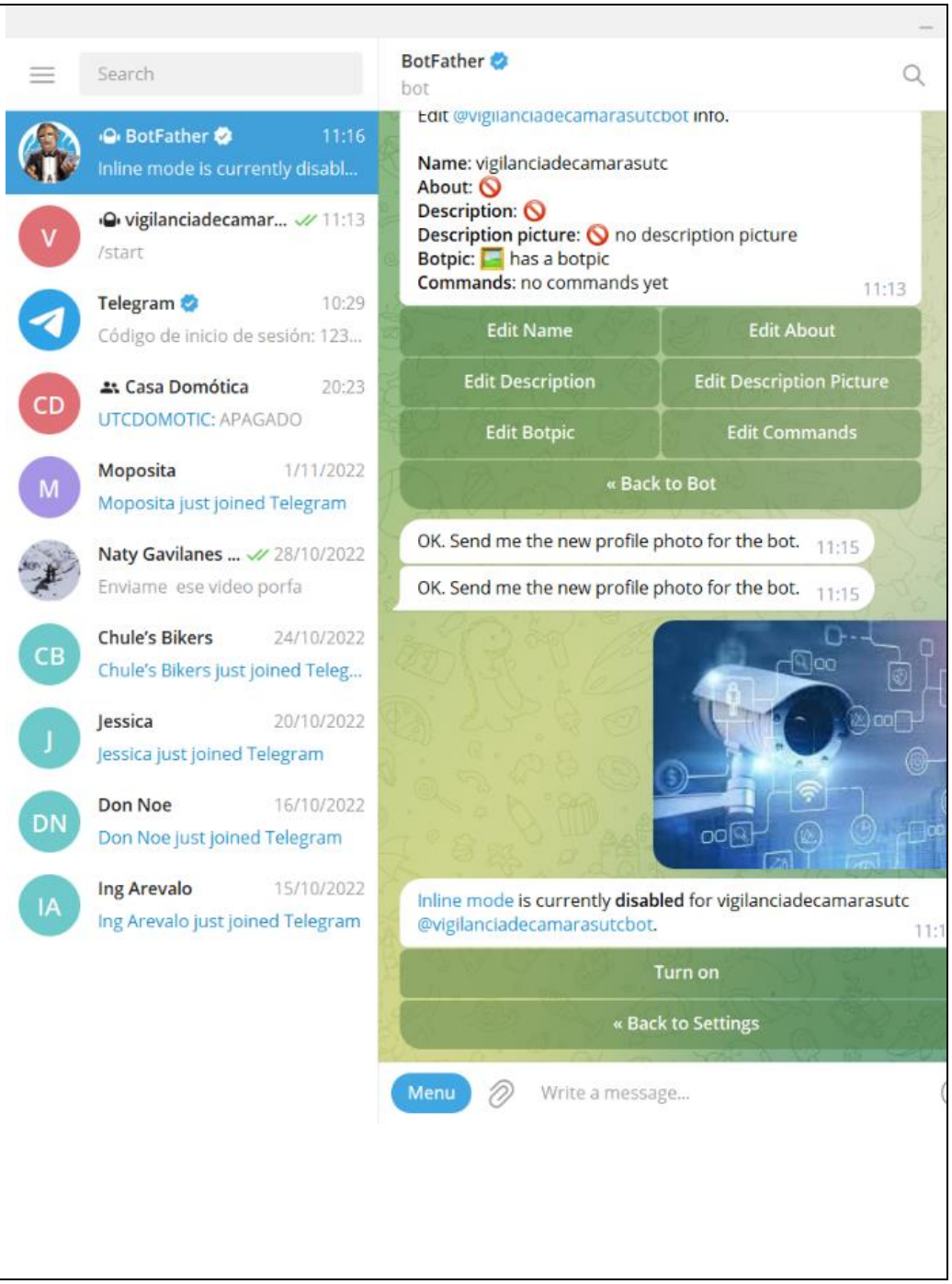


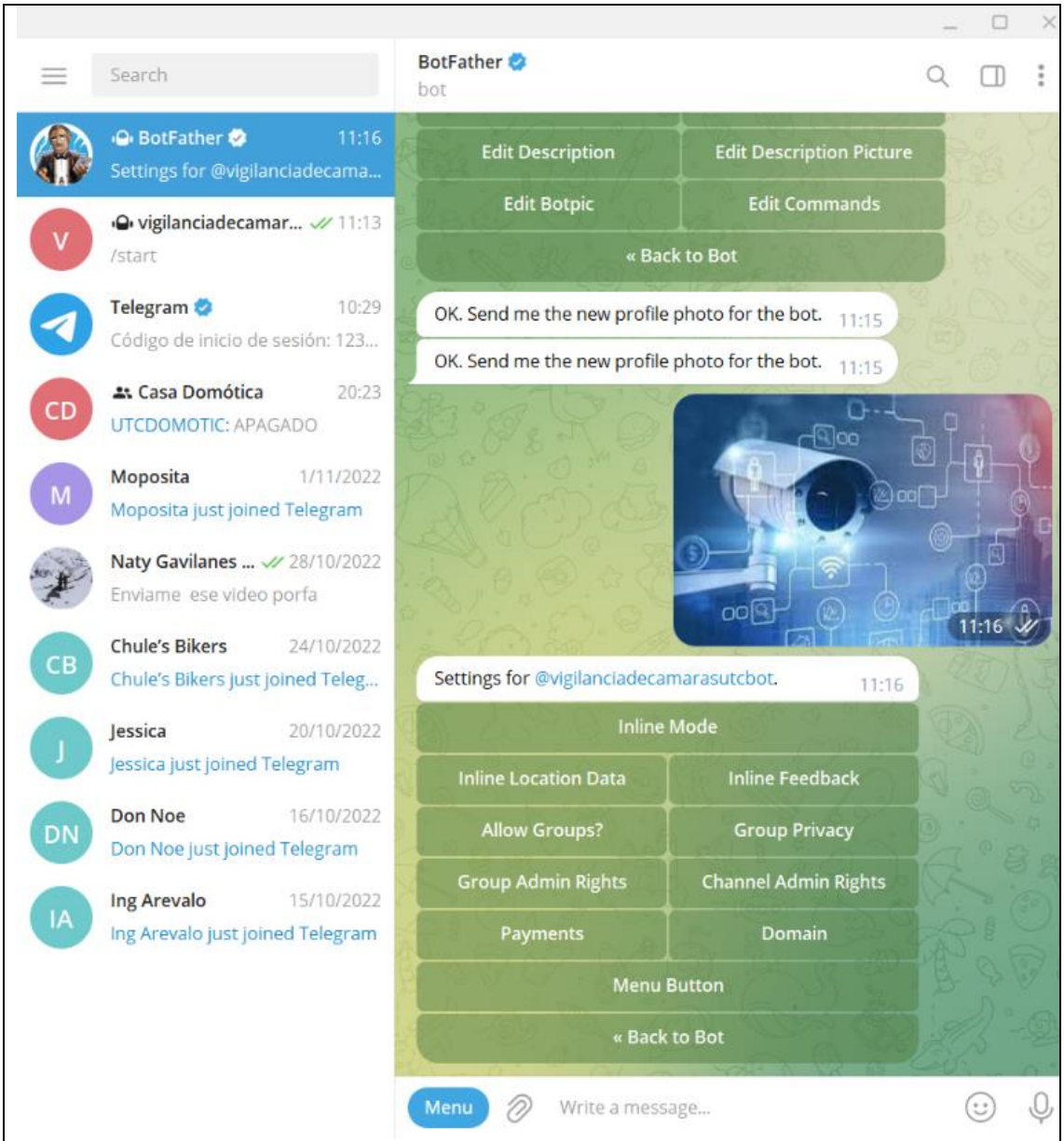



botfather
Global search results
BotFather
@BotFather
Bot Father
@BotFatherEpBot
BotFather
@bot_fatherz
BotFather
@bot_father_createbot
BotFather
@father_bot3
BotFather
@ebotfather
ایران فیلم
@BotFather_zz_xx_bot
Bot Father
@BotFatherc
Сарказмы на злобу дня
@botfather2
BotFather
@BotFatherd
No messages found
vigilanciadecamarasutc
bot
START
Sorry, the username must end in 'bot'. E.g. 'Tetris_bot' or 'Tetrisbot'
11:09
/newbot
11:11
Alright, a new bot. How are we going to call it? Please choose a name for your bot.
11:11
vigilanciadecamarasutc
11:11
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
11:11
vigilanciadecamarasutcbot
11:12
Done! Congratulations on your new bot. You will find it at t.me/vigilanciadecamarasutcbot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
Use this token to access the HTTP API:
5689930911:AAFPa1gue0YTkJCuCUU8aIfNwXK3dkNb3U
Keep your token secure and store it safely, it can be used by anyone to control your bot.
For a description of the Bot API, see this page:
https://core.telegram.org/bots/api
11:12
Menu
Write a message...

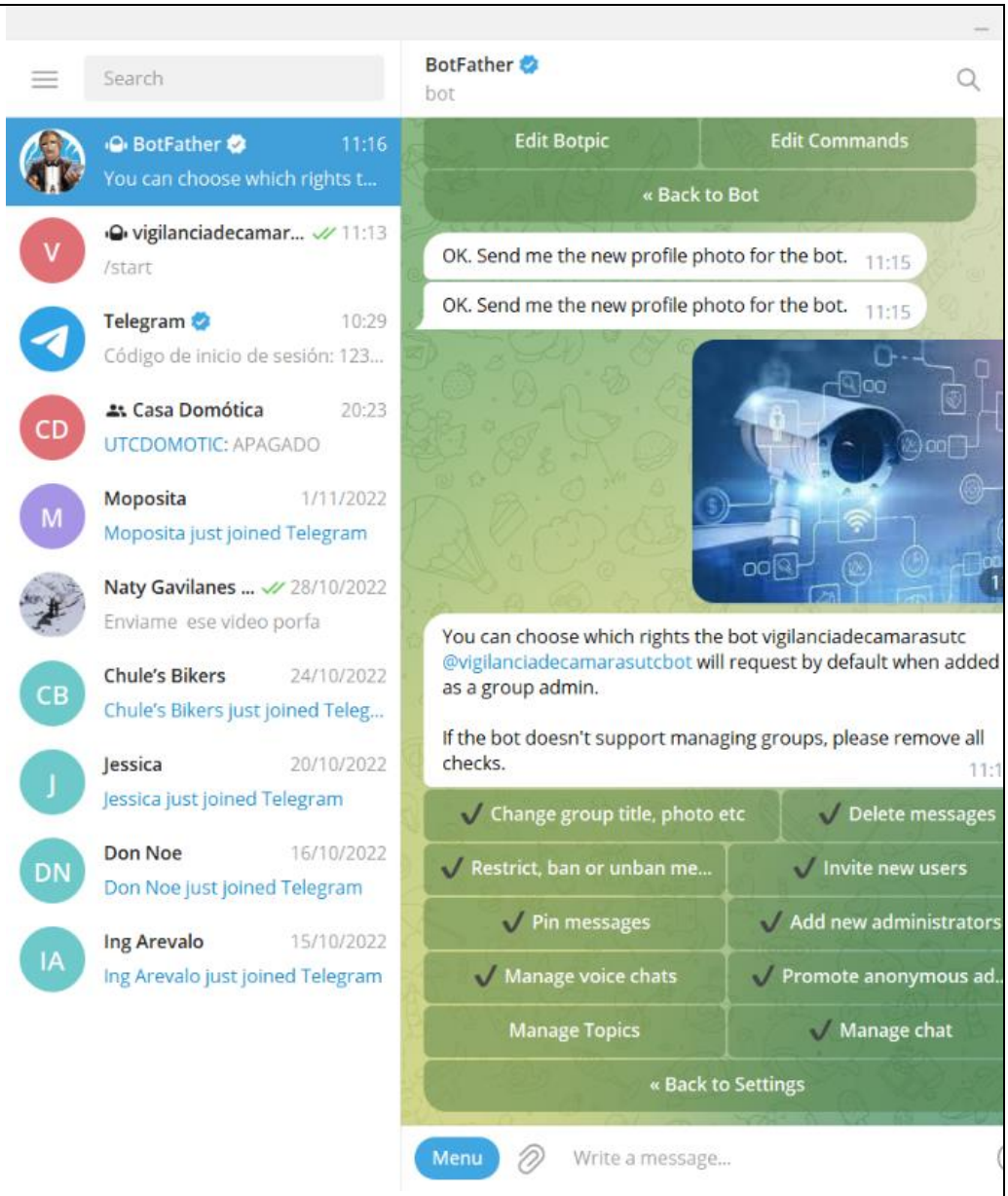

botfather
Global search results
BotFather
@BotFather
Bot Father
@BotFatherEpBot
BotFather
@bot_fatherz
BotFather
@bot_father_createbot
BotFather
@father_bot3
BotFather
@ebotfather
ایران فیلم
@BotFather_zz_xx_bot
Bot Father
@BotFatherc
Сарказмы на злобу дня
@botfather2
BotFather
@BotFatherd
No messages found
BotFather
bot
vigilanciadecamarasutcbot 11:12
Done! Congratulations on your new bot. You will find it at t.me/vigilanciadecamarasutcbot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
Use this token to access the HTTP API:
5689930911:AAFPa1gue0YTkJCuCUU8aIfnNwXK3dkNb3U
Keep your token secure and store it safely, it can be used by anyone to control your bot.
For a description of the Bot API, see this page:
https://core.telegram.org/bots/api 11:12
/mybots 11:13
Edit @vigilanciadecamarasutcbot info.
Name: vigilanciadecamarasutc
About:
Description:
Description picture: no description picture
Botpic: no botpic
Commands: no commands yet 11:13
Edit Name
Edit About
Edit Description
Edit Description Picture
Edit Botpic
Edit Commands
« Back to Bot
OK. Send me the new profile photo for the bot. 11:15
Menu
Write a message...
Description picture: no description picture
Botpic: has a botpic
Commands: no commands yet 11:13
Edit Name
Edit About
Edit Description
Edit Description Picture
Edit Botpic
Edit Commands
« Back to Bot
OK. Send me the new profile photo for the bot. 11:15
OK. Send me the new profile photo for the bot. 11:15
Here it is: vigilanciadecamarasutc @vigilanciadecamarasutcbot.
What do you want to do with the bot? 11:16
API Token
Edit Bot
Bot Settings
Payments
Transfer Ownership
Delete Bot
« Back to Bots List
Menu
Write a message...


botfather
Global search results
BotFather
@BotFather
Bot Father
@BotFatherEpBot
BotFather
@bot_fatherz
BotFather
@bot_father_createbot
BotFather
@father_bot3
BotFather
@ebotfather
ایران فیلم
@BotFather_zz_xx_bot
Bot Father
@BotFatherc
Сарказмы на злобу дня
@botfather2
BotFather
@BotFatherd
No messages found
BotFather
bot
Edit Name
Edit About
Edit Description
Edit Description Picture
Edit Botpic
Edit Commands
« Back to Bot
OK. Send me the new profile photo for the bot. 11:15
OK. Send me the new profile photo for the bot. 11:15
Settings for @vigilanciadecamarasutcbot. 11:16
Inline Mode
Allow Groups?
Group Privacy
Group Admin Rights
Channel Admin Rights
Payments
Domain
Menu Button
« Back to Bot
Menu
Write a message...

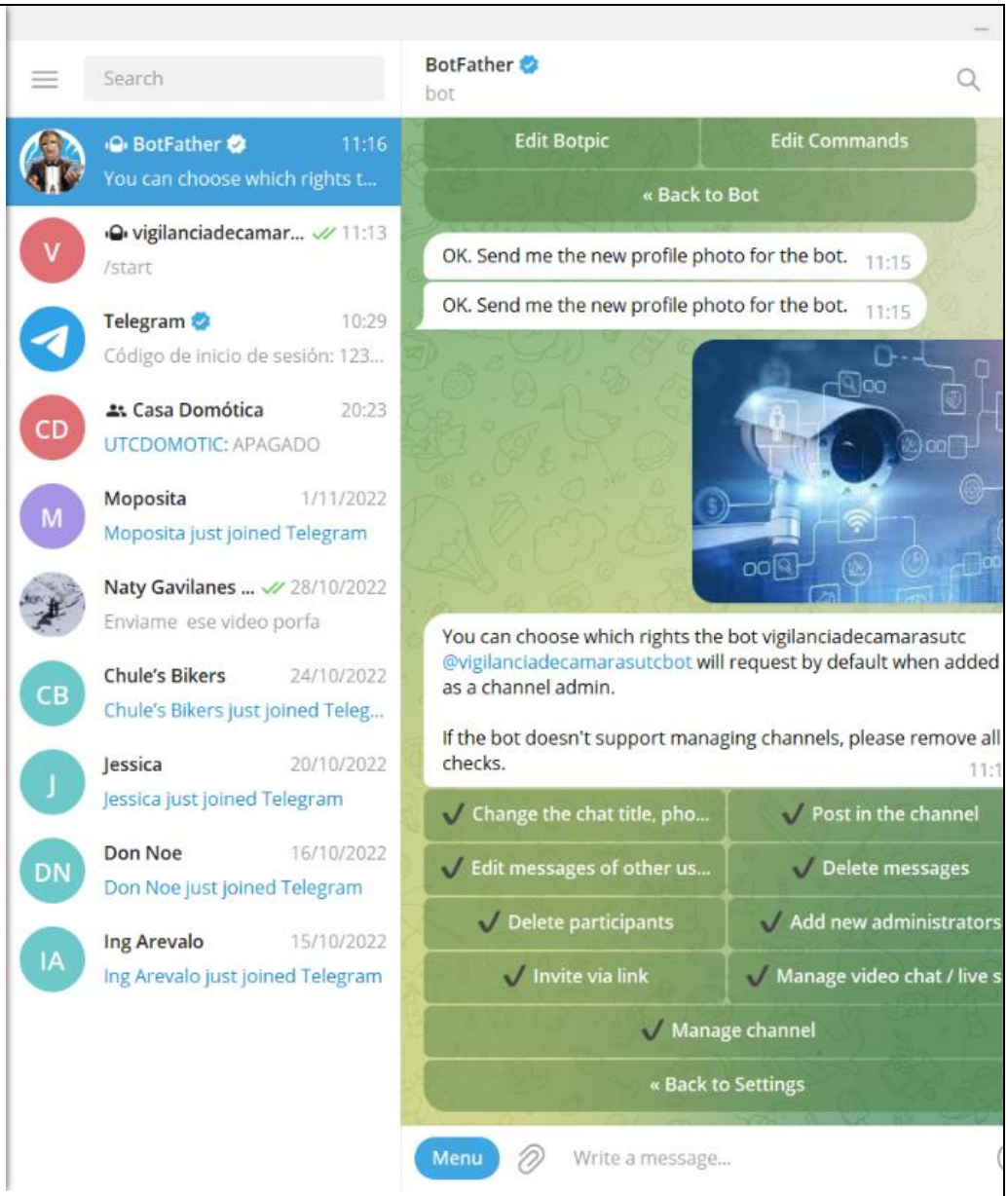

Search
BotFather
bot
BotFather 11:16
Inline mode is currently disabl...
vigilanciadecamar... 11:13
/start
Telegram 10:29
Código de inicio de sesión: 123...
Casa Domótica 20:23
UTCDOMOTIC: APAGADO
Moposita 1/11/2022
Moposita just joined Telegram
Naty Gavilanes ... 28/10/2022
Enviame ese video porfa
Chule's Bikers 24/10/2022
Chule's Bikers just joined Teleg...
Jessica 20/10/2022
Jessica just joined Telegram
Don Noe 16/10/2022
Don Noe just joined Telegram
Ing Arevalo 15/10/2022
Ing Arevalo just joined Telegram
Edit @vigilanciadecamarasutcbot info.
Name: vigilanciadecamarasutc
About:
Description:
Description picture: no description picture
Botpic: has a botpic
Commands: no commands yet
11:13
Edit Name
Edit About
Edit Description
Edit Description Picture
Edit Botpic
Edit Commands
« Back to Bot
OK. Send me the new profile photo for the bot. 11:15
OK. Send me the new profile photo for the bot. 11:15
Inline mode is currently disabled for vigilanciadecamarasutc @vigilanciadecamarasutcbot.
Turn on
« Back to Settings
Menu
Write a message...

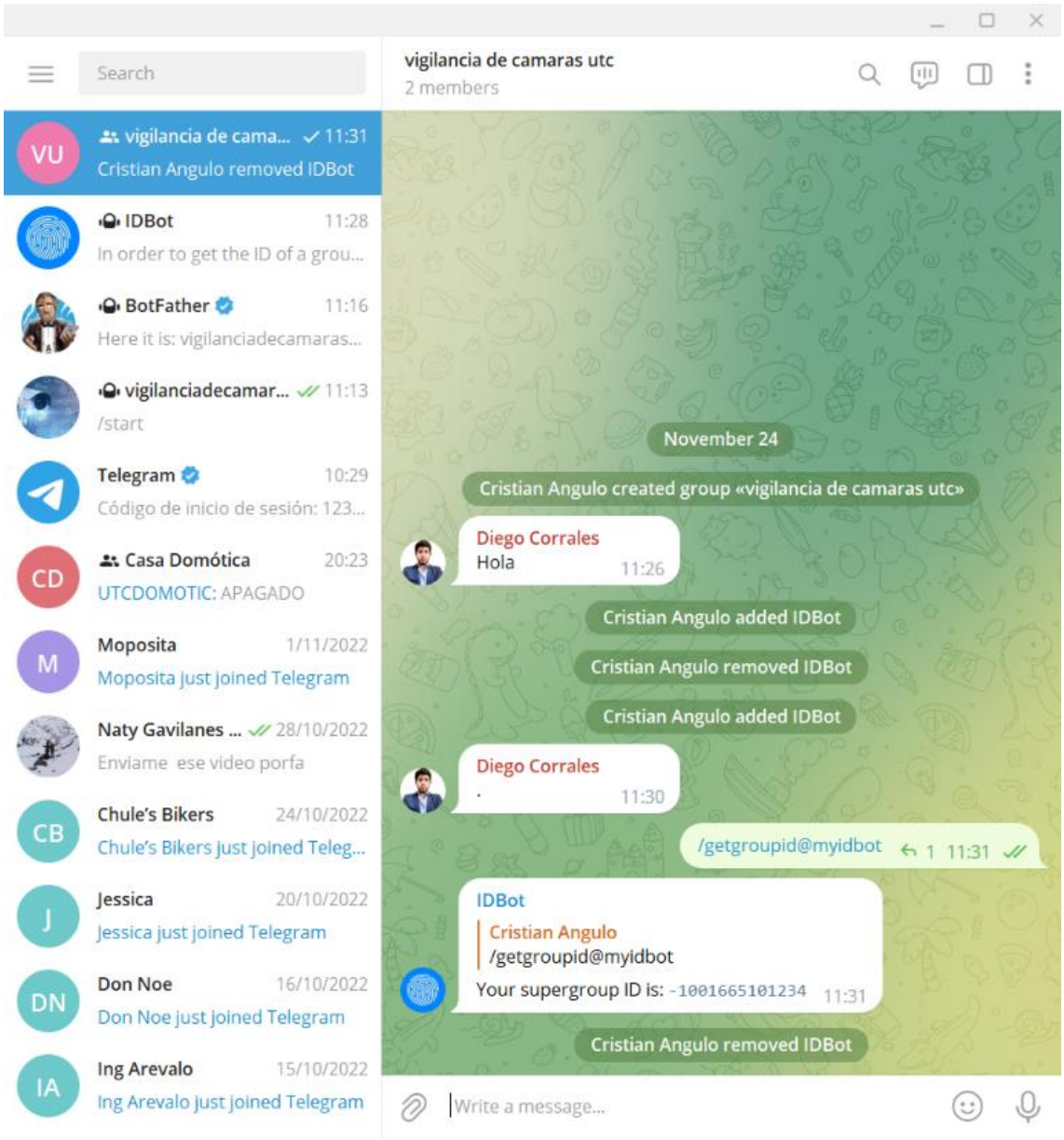


Revisar que allow groups este activado e ingresamos en group admin rights



Señalamos Chanel admin rights todos que esten con los vistos


Search
BotFather
bot
BotFather 11:16
You can choose which rights t...
vigilanciadecamar... 11:13
/start
Telegram 10:29
Código de inicio de sesión: 123...
Casa Domótica 20:23
UTCDOMOTIC: APAGADO
Moposita 1/11/2022
Moposita just joined Telegram
Naty Gavilanes ... 28/10/2022
Enviame ese video porfa
Chule's Bikers 24/10/2022
Chule's Bikers just joined Teleg...
Jessica 20/10/2022
Jessica just joined Telegram
Don Noe 16/10/2022
Don Noe just joined Telegram
Ing Arevalo 15/10/2022
Ing Arevalo just joined Telegram
Edit Botpic
Edit Commands
« Back to Bot
OK. Send me the new profile photo for the bot. 11:15
OK. Send me the new profile photo for the bot. 11:15
You can choose which rights the bot vigilanciadecamarasutc @vigilanciadecamarasutcbot will request by default when added as a channel admin.
If the bot doesn't support managing channels, please remove all checks.
Change the chat title, pho...
Post in the channel
Edit messages of other us...
Delete messages
Delete participants
Add new administrators
Invite via link
Manage video chat / live s
Manage channel
« Back to Settings
Menu
Write a message...


Search
vigilancia de camaras utc
2 members
VU
vigilancia de cama... 11:31
Cristian Angulo removed IDBot
IDBot 11:28
In order to get the ID of a grou...
BotFather 11:16
Here it is: vigilanciadecamaras...
vigilanciadecamar... 11:13
/start
Telegram 10:29
Código de inicio de sesión: 123...
CD
Casa Domótica 20:23
UTCDOMOTIC: APAGADO
M
Moposita 1/11/2022
Moposita just joined Telegram
Naty Gavilanes ... 28/10/2022
Enviame ese video porfa
CB
Chule's Bikers 24/10/2022
Chule's Bikers just joined Teleg...
J
Jessica 20/10/2022
Jessica just joined Telegram
DN
Don Noe 16/10/2022
Don Noe just joined Telegram
IA
Ing Arevalo 15/10/2022
Ing Arevalo just joined Telegram
November 24
Cristian Angulo created group «vigilancia de camaras utc»
Diego Corrales
Hola 11:26
Cristian Angulo added IDBot
Cristian Angulo removed IDBot
Cristian Angulo added IDBot
Diego Corrales
. 11:30
/getgroupid@myidbot 1 11:31
IDBot
Cristian Angulo
/getgroupid@myidbot
Your supergroup ID is: -1001665101234 11:31
Cristian Angulo removed IDBot
Write a message...

ANEXO IV	Código de ESP32CAM para telegram

```
#define PWDN_GPIO_NUM   32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM    0
#define SIOD_GPIO_NUM   26
#define SIOC_GPIO_NUM   27
#define Y9_GPIO_NUM     35
#define Y8_GPIO_NUM     34
#define Y7_GPIO_NUM     39
#define Y6_GPIO_NUM     36
#define Y5_GPIO_NUM     21
#define Y4_GPIO_NUM     19
#define Y3_GPIO_NUM     18
#define Y2_GPIO_NUM      5
#define VSYNC_GPIO_NUM  25
#define HREF_GPIO_NUM   23
#define PCLK_GPIO_NUM   22

#define tímeseme 1 // tu zona horaria
#define flash Pin 4 // pin al que se conecta el flash
#define pantt1Pin 15 // pin al que se conecta el servomotor


static const int servoPin = 14; //printed G14 on the board

Servo_ESP32 servo1;

int angle = 0;
int angleStep = 20;

int angleMin =0;
int angleMax = 180;

#define pirSensorPin 13 // pin al que está conectado el sensor PIR
#define getMessageInterval 1000 // intervalo de recepción de mensajes de un usuario de Telegram


//const char WiFiSSID[] = "WIFI-UTC"; // el nombre de su red WiFi
//const char WiFiPASS[] = ""; // la contraseña de su red WiFi
const String botToken = "5689930911:AAFPa1gue0YTkJCuCUU8aIfnNwXK3dkNb3U"; // el token de tu bot de Telegram
const String yourChatIDs[] = {"-1001665101234"}; //  ID de chat de los usuarios de Telegram que pueden controlar su bot de Telegram
WiFiClientSecure client; // Crear un cliente WiFi compatible con TSL
UniversalTelegramBot bot(botToken, client); //crea un objeto para controlar tu bot de Telegram

String videoFileName; //una variable para almacenar el nombre del archivo de vídeo que la placa ESP32-CAM le enviará a Telegram
String minutes; // variable para almacenar los minutos
String hours; // variable para almacenar el reloj
String days; // una variable para almacenar los días desde el comienzo del mes
String month; //una variable para almacenar el número del mes
```

```
int year; // una variable para almacenar el año

String settings; // variable para almacenar los parámetros de la imagen, los parámetros de la alarma y la aceleración del vídeo
int firstHashIndex; // variable para almacenar el índice del primer hashtag de separación
int secondHashIndex; // una variable para almacenar el índice del segundo hashtag de separación
int thirdHashIndex; // una variable para almacenar el índice del tercer hashtag de separación
int fourthHashIndex; // variable para almacenar el índice del cuarto hashtag de separación
int fifthHashIndex; // una variable para almacenar el índice del quinto hashtag de separación
int sixthHashIndex; // variable para almacenar el índice del sexto hashtag de separación
int seventhHashIndex; // variable para almacenar el índice del séptimo hashtag de separación
int eighthHashIndex; // variable para almacenar el índice del octavo hashtag de separación
int ninthHashIndex; // variable para almacenar el índice del noveno hashtag de separación

String message; // una variable para almacenar el mensaje recibido del usuario de Telegram
String chatID; // una variable para almacenar el identificador de chat del usuario de Telegram que envió el mensaje a tu bot de Telegram
String videoChatID; // una variable para almacenar el identificador de chat del usuario de Telegram que envió la orden de grabar y enviar el vídeo a Telegram
String alarmChatID; // una variable para almacenar el identificador de chat del usuario de Telegram que envió el comando para activar la alarma de movimiento
String keyboardJson; //una variable para almacenar el teclado de Telegram en formato JSON
String settingsMessage; // una variable para almacenar el mensaje de parámetros que le enviaremos en Telegram

int framesizeInt; // una variable para almacenar la resolución de la imagen como int
String framesize = "VGA"; // una variable para almacenar la resolución de la imagen como una cadena
String flipImage = "Нет"; // una variable para almacenar si la imagen será volteada
String brightness = "100%"; // una variable para almacenar el brillo de la imagen
String turnOnFlash = "Нет"; // una variable para almacenar si el flash se disparará para la foto
String specialEffect = "comun"; // una variable para almacenar el efecto especial de la imagen
String whiteBalanceMode = "Balance Automatico"; // una variable para almacenar el modo de balance de blancos de la imagen

int pirSensorDelay = 15; // variable para almacenar el retardo del sensor PIR
bool setImageFlipMode; // una variable para almacenar si se está ejecutando un modo para establecer si la imagen será volteada
bool setPirSensorDelayMode; // una variable para almacenar si el modo de ajuste del retardo del sensor PIR está en marcha
bool isAlarmRunning = false; // una variable para almacenar si la alarma de movimiento se activa como un booleano
String alarmRunning = "Нет"; // una variable para almacenar si se activa una alarma de movimiento como una cadena
String alarmMessageType = "Video"; //una variable para almacenar el tipo de mensaje de alarma de movimiento

bool flashState; // una variable para almacenar si el flash está encendido
bool pantState;


bool pantt1State; // una variable para almacenar si el pant está encendido
```

```
int frameInterval = 125; // una variable para almacenar el intervalo de fotogramas del vídeo en milisegundos
float videoSpeedFloat = 1; // una variable para almacenar la aceleración del vídeo como un flotador
String videoSpeedString = "1x"; // variable para almacenar la aceleración de vídeo como una cadena

// constante para almacenar los comandos de tu bot de Telegram
const String botCommands = F ("["
                        "{\"command\":\"takephoto\", \"description\":\"enviar foto\"},"
                        "{\"command\":\"recordvideo\", \"description\":\"enviar video\"},"
                        "{\"command\":\"alarm\", \"description\":\"gestion de alarmas\"},"
                        "{\"command\":\"flash\", \"description\":\" control de flash\"},"
                        "{\"command\":\"pantt1\", \"description\":\" control de ángulo de camara\"},"

                        "{\"command\":\"alarmsettings\", \"description\":\"parámetros de alarma \"},"
                        "{\"command\":\"imagesettings\", \"description\":\" parámetros de imagen\"}"
                        "]");

unsigned long lastMessageMillis; // una variable para almacenar la hora del último mensaje recibido de Telegram del usuario
unsigned long pirSensorDelayMillis; //una variable para mantener la hora de la última comprobación de que se ha detectado movimiento

sensor_t * camera = NULL; // objeto para ajustar la configuración de la imagen
camera_fb_t * fb = NULL; // facilidad para almacenar la foto que enviaremos a Telegram
camera_fb_t * videoFb = NULL; // una variable para almacenar el fotograma de vídeo
camera_fb_t * nextFb = NULL; // una variable para almacenar el siguiente fotograma de vídeo
camera_fb_t * currentFb = NULL; // variable para almacenar el fotograma de vídeo actual

int videoBufferSize;
int idxBufferSize;

int videoPtr;
uint8_t* videoBuffer;
size_t videoFileLength;

uint8_t * psramVideoBuffer = NULL;
uint8_t * psramIdxBuffer = NULL;
uint8_t * psramVideoPtr = 0;
uint8_t * psramIdxPtr = 0;

bool videoRecorded; // una variable para almacenar si la grabación de vídeo ha terminado
bool motionDetected; // una variable para almacenar si se detecta ahora el movimiento

float fps; // una variable para almacenar la frecuencia de imagen del vídeo
float realFPS; // una variable para almacenar la frecuencia de imagen real del vídeo
```

```
uint8_t attainedFPS; // una variable para almacenar la frecuencia de imagen redondeada del vídeo
uint32_t attainedPerFrame; // una variable para almacenar la longitud redondeada del fotograma en el vídeo en microsegundos
float microsecondsPerFrame; // una variable para almacenar la duración del fotograma en el vídeo en microsegundos

int getGoodPictureFailures; // una variable para almacenar el número de intentos fallidos de obtener un buen fotograma para el vídeo

int fbBlockLength;
int jpegSizeRemnant;
uint8_t* fbBlockStart;

unsigned long videoStartMillis; // variable para almacenar la hora de inicio de la grabación de vídeo
unsigned long videoEndMillis; // una variable para almacenar la hora de finalización de la grabación de vídeo
unsigned long lastFrameMillis; // una variable para almacenar la hora en que se recibió el último fotograma de vídeo
unsigned long frameWriteMillis; // variable para almacenar el tiempo de grabación de fotogramas de vídeo en la PSRAM
unsigned long currentFrameMillis; // una variable para almacenar la hora de recepción del fotograma de vídeo actual
unsigned long startMillis;
unsigned long fbBlockMillis;

uint32_t videoDuration; // una variable para almacenar la duración del vídeo en milisegundos
uint16_t frameCount; // una variable para almacenar el número de fotogramas del vídeo
uint16_t remnant;
uint32_t videoLength;

unsigned long videoSize;
unsigned long idxOffset;
unsigned long jpegSize;

uint8_t dcBuffer [4] = {0x30, 0x30, 0x64, 0x63};
uint8_t zeroBuffer [4] = {0x00, 0x00, 0x00, 0x00};
uint8_t idx1Buffer [4] = {0x69, 0x64, 0x78, 0x31};
uint8_t video1Buffer [4] = {0x41, 0x56, 0x49, 0x31};

bool isMoreDataAvailable();

int currentByte;
uint8_t* fbBuffer;
size_t fbLength;

// para configurar y hacer funcionar la cámara en la placa ESP32-CAM
bool setupCamera() {
  camera_config_t config; // crear un objeto para configurar la cámara en la placa ESP32-CAM
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
```

```
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000; // ajustar la frecuencia de la cámara
  config.pixel_format = PIXFORMAT_JPEG; // Establezca el formato de la imagen
(PIXFORMAT_+YUV422|GRAYSCALE|RGB565|JPEG)

  config.frame_size = FRAMESIZE_VGA; // Ajuste la resolución de la imagen
(FRAMESIZE_+
UXGA|SXGA|XGA|HD|SVGA|VGA|HVGA|CIF|QVGA|240X240|HQVGA|QCIF|QVGA|96
X96)

  if (psramFound()) { // si se detecta un tamaño suficiente de PSRAM
    config.jpeg_quality = 5; // ajustar la calidad de la imagen (1 - 63, a menor número, mayor
calidad de imagen)
    config.fb_count = 4; // establecer el número máximo de fotos que se pueden guardar en la
PSRAM
  } else {
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  static char * memTmp1 = (char*) malloc(32 * 1024);
  static char * memTmp2 = (char*) malloc(32 * 1024);

  esp_err_t err = esp_camera_init(&config); // ejecutar la cámara en la placa ESP32-CAM
  if (err != ESP_OK) { // si la cámara de la placa ESP32-CAM no se inicia
    Serial.printf("se ha producido el error en la cámara: 0x%x ", err);
    return false; // devolver lo que la cámara de la placa ESP32-CAM no pudo iniciar
  }

  // asignar espacio en PSRAM
  free(memTmp2);
  memTmp2 = NULL;
  free(memTmp1);
  memTmp1 = NULL;

  camera = esp_camera_sensor_get(); // obtener el módulo de la cámara en la placa ESP32-
CAM
  camera->set_quality(camera, 10); // establecer la calidad de la imagen
  delay(500);
  return true; // Devuelve que la cámara de la placa ESP32-CAM está funcionando
}
```

```
// función para enviar fotos a Telegram
void sendPhotoToTelegram() {
  if (turnOnFlash == "siii1") { // si el flash encendido para la foto está ajustado
    digitalWrite(flashPin, HIGH); // encender el flash
  } else {
    digitalWrite(flashPin, LOW); // apagar el flash
  }
  delay(50); // esperar 50 milisegundos para que el flash se apague

  fb = NULL; // dejar una foto
  fb = getGoodPicture(); // Recepción y grabación de fotos
  digitalWrite(flashPin, LOW); //apagar el flash

  if (fb) { // si la foto se recibe con éxito
    currentByte = 0; // poner a cero el byte actual
    fbLength = fb->len; // registrar el tamaño de la foto
    fbBuffer = fb->buf; // amortiguar la foto

    if (motionDetected) { // si se detecta movimiento
      bot.sendChatAction(alarmChatID, "upload_photo"); // enviar por Telegram al usuario que
ha activado la alarma, una acción de chat sobre el envío de la foto a Telegram
    } else {
      bot.sendChatAction(chatID, "upload_photo"); //enviar un Telegram al usuario que envió
el comando de la foto, una acción de chat sobre el envío de la foto a Telegram
    }

    if (motionDetected) { // si se detecta movimiento
      // enviar una foto con un pie de foto de Telegram al usuario que activó la alarma
      bot.sendMultipartFormDataToTelegramWithCaption("sendPhoto", "photo", "img.jpg",
"image/jpeg",
          "Movimiento Detectado!", alarmChatID, fbLength, isMoreDataAvailable,
getNextByte, nullptr, nullptr);
    } else {
      // enviar una foto de Telegram al usuario que envió el comando de la foto
      bot.sendPhotoByBinary(chatID, "image/jpeg", fbLength, isMoreDataAvailable,
getNextByte, nullptr, nullptr);
    }
    esp_camera_fb_return(fb); // guardar la foto en la PSRAM
  } else {
    Serial.println("No se pudo obtener la foto:(");
    if (motionDetected) { // si se detecta movimiento
      bot.sendMessage(alarmChatID, "No se pudo obtener la foto:(", "Markdown");
    } else {
      bot.sendMessage(chatID, "No se pudo obtener la foto:(", "Markdown");
    }
    motionDetected = false; // restablecer qué movimiento se detecta
  }
}

// función para grabar y enviar vídeos en Telegram
void sendVideoToTelegram() {
  if (motionDetected) { // si se detecta movimiento
    bot.sendChatAction(alarmChatID, "record_video"); // enviar un Telegram al usuario que
activó la alarma, chateando sobre la tarjeta ESP32-CAM que graba el vídeo
  } else {
```

```
    bot.sendChatAction(videoChatID, "record_video"); // enviar un Telegram al usuario que envió el comando de vídeo, chateando sobre la placa ESP32-CAM que graba vídeo
  }

  // crear una tarea freeRTOS para grabar vídeo
  xTaskCreatePinnedToCore(
    videoRecordingTask,   // función
    "videoRecordingTask", // título
    10000,              // tamaño de la pila
    NULL,               // opciones
    1,                // prioridad
    NULL,               // un objeto para gestionar la tarea
    1);               // número de núcleo en la placa ESP32-CAM
}

// Función de grabación de parámetros de imagen, parámetros de alarma y aceleración de vídeo en la memoria SPIFFS
void saveSettingsToSPIFFS() {
  // escribir todos los parámetros en una sola variable, separándolos con un hashtag
  settings = framesize + "#" + brightness + "#" + specialEffect + "#" + whiteBalanceMode + "#" + flipImage + "#" + turnOnFlash + "#";
  settings = settings + videoSpeedString + "#" + String(pirSensorDelay) + "#" + alarmMessageType + "#" + alarmRunning;
  File settingsFile = SPIFFS.open("/Settings.txt", FILE_WRITE); // abrir el archivo con los parámetros para escribir

  if (settingsFile) { // si el archivo de parámetros se abre con éxito
    Serial.println("Archivo con parámetro abierto con éxito para escribir!");
  } else {
    Serial.println("No se pudo abrir el archivo con opciones para escribir:(");
    return;
  }

  if (settingsFile.print(settings)) { // si los parámetros se han escrito con éxito en el archivo
    Serial.println("El archivo de parámetros se escribió con éxito!");
  } else {
    Serial.println("Error al escribir archivo de parámetros:(");
    return;
  }

  settingsFile.close(); // cerrar el archivo de parámetros
}

// función para ajustar la resolución de la imagen
void setImageFramesize() {
  if (framesize == "XGA") {
    framesizeInt = FRAMESIZE_XGA;
    camera->set_framesize(camera, FRAMESIZE_XGA); // ajustar la resolución a un nivel superior al medio
  } else if (framesize == "SVGA") {
    framesizeInt = FRAMESIZE_SVGA;
    camera->set_framesize(camera, FRAMESIZE_SVGA); // establecer la resolución del medio
  } else if (framesize == "VGA") {
    framesizeInt = FRAMESIZE_VGA;
```

```
    camera->set_framesize(camera, FRAMESIZE_VGA); // ajustar la resolución a la
normalidad
  } else if (framesize == "CIF") {
    framesizeInt = FRAMESIZE_CIF;
    camera->set_framesize(camera, FRAMESIZE_CIF); // establecer una resolución baja
  } else if (framesize == "QVGA") {
    framesizeInt = FRAMESIZE_QVGA;
    camera->set_framesize(camera, FRAMESIZE_QVGA); // establecer la resolución mínima
  }
}

// función para ajustar el brillo de la imagen
void setImageBrightness() {
  if (brightness == "200%") {
    camera->set_brightness(camera, 2); // ajustado a la máxima luminosidad
  } else if (brightness == "150%") {
    camera->set_brightness(camera, 1); // ajustar el brillo más alto que el estándar
  } else if (brightness == "100%") {
    camera->set_brightness(camera, 0); // establecer la luminosidad estándar
  } else if (brightness == "75%") {
    camera->set_brightness(camera, -1); // ajustar el brillo más bajo que el estándar
  } else if (brightness == "50%") {
    camera->set_brightness(camera, -2); // establecer el brillo mínimo
  }
}

// para establecer un efecto de imagen especial
void setImageSpecialEffect() {
  if (specialEffect == "comun") {
    camera->set_special_effect(camera, 0); // establecer el efecto especial "Sin efecto"
  } else if (specialEffect == "Negativo") {
    camera->set_special_effect(camera, 1); // establecer el efecto especial "Negativo"
  } else if (specialEffect == "Gris") {
    camera->set_special_effect(camera, 2); // establecer el efecto especial "Blanco y Negro".
  } else if (specialEffect == "Rojo") {
    camera->set_special_effect(camera, 3); // establecer el efecto especial "Sombras de rojo"
  } else if (specialEffect == " Verde") {
    camera->set_special_effect(camera, 4); // establecer el efecto especial "Shades of Green".
  } else if (specialEffect == " Azul") {
    camera->set_special_effect(camera, 5); // establecer el efecto especial "Shades of Blue"
(tonos de azul)
  } else if (specialEffect == " Sepia") {
    camera->set_special_effect(camera, 6); // establecer el efecto especial sepia
  }
}

// para ajustar el modo de balance de blancos de la imagen
void setImageWhiteBalanceMode() {
  if (whiteBalanceMode == "Balance Automatico") {
    camera->set_wb_mode(camera, 0); // ajustar el modo de balance de blancos a "Balance
Automático"
  } else if (whiteBalanceMode == " Solar") {
    camera->set_wb_mode(camera, 1); // ajuste el modo de balance de blancos a soleado
  } else if (whiteBalanceMode == "Nuboso") {
    camera->set_wb_mode(camera, 2); // ajustar el modo de balance de blancos a "Nublado"
```

```
  } else if (whiteBalanceMode == "Oficina") {
    camera->set_wb_mode(camera, 3); // ajustar el modo de balance de blancos a "Oficina"
  } else if (whiteBalanceMode == "Hogar") {
    camera->set_wb_mode(camera, 4); // ajustar el modo de balance de blancos a "Inicio"
  }
}

// para establecer si la imagen está invertida
void setImageFlip() {
  if (flipImage == "siii1") {
    camera->set_vflip(camera, 1); // establecer que la imagen se invierta
  } else {
    camera->set_vflip(camera, 0); // establecer que la imagen no se invierta
  }
}

// función para establecer la aceleración de vídeo
void setVideoSpeed() {
  if (videoSpeedString == "0.5x") {
    videoSpeedFloat = 0.5; // registrar que la aceleración de vídeo es del 50%.
    frameInterval = 0; // grabar que el vídeo tiene 25fps
  } else if (videoSpeedString == "1x") {
    videoSpeedFloat = 1; // registrar que la aceleración de vídeo es del 100%
    frameInterval = 125; // grabar que el video tiene 8fps
  } else if (videoSpeedString == "2.5x") {
    videoSpeedFloat = 2.5; // registrar que la aceleración de vídeo es del 250%.
    frameInterval = 250; // записываем то, что у видео 4fps
  } else if (videoSpeedString == "5x") {
    videoSpeedFloat = 5; // registrar que la aceleración de vídeo es del 500%.
    frameInterval = 500; // grabar que el video tiene 2fps
  }
}

// para comprobar si un mensaje de un usuario de Telegram es un número
bool isMessageNumber() {
  for (int i = 0; i < message.length(); i ++) { // bucle para comprobar cada carácter del mensaje
recibido
    if (isDigit(message.charAt(i))) { // si el carácter del mensaje recibido es un número
      continue; // seguir comprobando
    } else {
      return false; //devuelve false, el mensaje recibido no es un número
    }
  }
  return true; // devuelve verdadero, el mensaje recibido es un número
}

// para comprobar si has introducido el identificador de chat del usuario de Telegram que ha
enviado el mensaje a tu bot de Telegram
bool isChatIDMatches() {
  for (int i = 0; i <= 10; i ++) { // un bucle para comparar cada id de chat que especifiques con
el id de chat del usuario de Telegram que envió el mensaje a tu bot de Telegram
    if (chatID == yourChatIDs[i]) { // si el identificador de chat del usuario de Telegram que
ha enviado el mensaje a tu bot de Telegram se encuentra en la lista con el identificador de
chat que has especificado
      return true; // devuelve true, el identificador de chat del usuario de Telegram que ha
```

```
enviado el mensaje a tu bot de Telegram coincide con uno de los identificadores de chat que has especificado
    } else {
      continue; // seguir comprobando
    }
  }
  return false; // devuelve false, el identificador de chat del usuario de Telegram que ha enviado el mensaje a tu bot de Telegram no coincide con ninguno de los identificadores de chat que has especificado
}

void setup() {
servo1.attach(servoPin);
  pinMode(pantt1Pin,OUTPUT);

  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // desactivación del detector de corte de energía
  Serial.begin(115200); // Ajuste la velocidad del puerto COM
  pinMode(flashPin, OUTPUT); // establecer el pin al que se conecta el flash como salida
  pinMode(pirSensorPin, INPUT_PULLUP); // establecer el pin al que se conecta el detector de movimiento como entrada

  videoBufferSize = 3000 * 1024;
  idxBufferSize = 2020;
  psramVideoBuffer = (uint8_t*)ps_malloc(videoBufferSize);
  if (psramVideoBuffer == 0) Serial.println("No se puede asignar espacio de video en PSRAM :(");
  psramIdxBuffer = (uint8_t*)ps_malloc(idxBufferSize);
  if (psramIdxBuffer == 0) Serial.println("Error al asignar espacio para idx en PSRAM :(");

  if (setupCamera()) { // si la cámara de la placa ESP32-CAM está configurada y funcionando correctamente
    Serial.println("La cámara se lanzó con éxito!");
  } else {
    Serial.println("No se pudo iniciar la cámara :(");
    ESP.restart(); // reiniciar la placa ESP32-CAM
  }

  if (SPIFFS.begin(true)) { // si la memoria SPIFFS se inicia con éxito
    Serial.println("El montaje de SPIFFS fue exitoso!");
  } else {
    Serial.println("No se puede montar SPIFFS :(");
    return;
  }

  if (SPIFFS.exists("/Settings.txt")) { // si existe un archivo con parámetros de imagen, parámetros de alarma y aceleración de vídeo en la memoria del SPIFFS
    File settingsFile = SPIFFS.open("/Settings.txt"); // abrir el archivo con los parámetros de la imagen, los parámetros de la alarma y la aceleración del vídeo para su lectura
    if (settingsFile) { // si el archivo con los parámetros de imagen, los parámetros de alarma y la aceleración de vídeo se abre con éxito para su lectura
      Serial.println("Archivo de configuración abierto con éxito para la lectura!");
    } else {
      Serial.println("No se puede abrir el archivo de configuración de lectura :(");
      return;
```

```
  }

  while (settingsFile.available()) {
    settings = settingsFile.readString(); // recuperar el contenido del archivo
    Serial.println("Contenido del archivo: " + String(settings));

    firstHashIndex   = settings.indexOf('#');                    // obtener el índice del primer
hashtag separador
    secondHashIndex  = settings.indexOf('#', firstHashIndex + 1);  // obtener el índice del
segundo hashtag de separación
    thirdHashIndex   = settings.indexOf('#', secondHashIndex + 1); // obtener el índice del
tercer hashtag de separación
    fourthHashIndex  = settings.indexOf('#', thirdHashIndex + 1);  // obtener el índice del
cuarto hashtag de separación
    fifthHashIndex   = settings.indexOf('#', fourthHashIndex + 1); // obtener el índice del
quinto hashtag de separación
    sixthHashIndex   = settings.indexOf('#', fifthHashIndex + 1);  // obtener el índice del
sexto hashtag de separación
    seventhHashIndex = settings.indexOf('#', sixthHashIndex + 1);  // obtener el índice del
séptimo hashtag de separación
    eighthHashIndex = settings.indexOf('#', seventhHashIndex + 1); // obtener el índice del
octavo hashtag de separación
    ninthHashIndex = settings.indexOf('#', eighthHashIndex + 1);   // obtener el índice del
noveno hashtag de separación

    framesize = settings.substring(0, firstHashIndex); // seleccionar la resolución de la imagen
entre todos los parámetros
    brightness = settings.substring(firstHashIndex + 1, secondHashIndex); // destacar el brillo
de la imagen a partir de todos los parámetros
    specialEffect = settings.substring(secondHashIndex + 1, thirdHashIndex); // resaltar el
efecto de imagen especial de todos los ajustes
    whiteBalanceMode = settings.substring(thirdHashIndex + 1, fourthHashIndex); //Resalte
el modo de balance de blancos de la imagen entre todos los ajustes
    flipImage = settings.substring(fourthHashIndex + 1, fifthHashIndex); // seleccione entre
todos los parámetros si la imagen debe ser volteada
    turnOnFlash = settings.substring(fifthHashIndex + 1, sixthHashIndex); // seleccionar
entre todas las opciones si se utilizará el flash para la foto
    videoSpeedString = settings.substring(sixthHashIndex + 1, seventhHashIndex); // resaltar
la aceleración de vídeo de todos los ajustes
    alarmMessageType = settings.substring(eighthHashIndex + 1, ninthHashIndex); //
seleccionar el tipo de mensaje de alarma de movimiento entre todos los parámetros
    alarmRunning = settings.substring(ninthHashIndex + 1, settings.length()); // destacar de
todos los parámetros si se activa o no una alarma de movimiento
    pirSensorDelay = (settings.substring(seventhHashIndex + 1, eighthHashIndex)).toInt(); //
asignar el retraso del sensor PIR de todos los parámetros
  }

  settingsFile.close(); // cerrar el archivo con la configuración de la imagen, la configuración
de la alarma y la aceleración del vídeo

  setImageFramesize(); // establecer la resolución de la imagen
  setImageFlip(); // establecer si la imagen debe ser volteada
  setImageBrightness(); // ajustar el brillo de la imagen
  setImageSpecialEffect(); // establecer un efecto de imagen especial
  setImageWhiteBalanceMode(); // establecer el modo de balance de blancos de la imagen
```

```
  setVideoSpeed(); // establecer la aceleración de vídeo

  if (alarmRunning == "siii1") { // si se activa una alarma de movimiento
    isAlarmRunning = true; // registrar que se ha activado una alarma de movimiento
  }
 }

 //Serial.println("Conectándose a " + String(WiFiSSID));
 //WiFi.begin(WiFiSSID, WiFiPASS); // conectarse a su red WiFi

 WiFiManager wm;

 bool res;

 res = wm.autoConnect("CamAP","12345678"); // password protected ap

  if(!res) {
     Serial.println("Failed to connect");
     // ESP.restart();
  }
  else {
     //if you get here you have connected to the WiFi
     Serial.println("connected...yeey :)");
  }


 if (WiFi.waitForConnectResult() == WL_CONNECTED) { // si la tarjeta ESP32-CAM se
conecta con éxito a su red WiFi
  Serial.print("¡Conectado a una red WiFi! Dirección IP local: ");
Serial.println(WiFi.localIP());

  esp_err_t setPowerSave = esp_wifi_set_ps(WIFI_PS_MIN_MODEM); //
WIFI_PS_MIN_MODEM WIFI_PS_NONE desactiva el ahorro de energía para WiFi
  client.setInsecure(); // Desactiva los certificados HTTPS para conectarte a tu bot de
Telegram
  bot.longPoll = 3; // establece cuánto tiempo esperará el bot de Telegram antes de devolver
los "mensajes ahora" en segundos
  bot.setMyCommands(botCommands); // Configuración de los comandos del bot de
Telegram

  configTime(timezone * 3600, 3600, "pool.ntp.org"); // establecer la zona horaria y la
dirección del servidor NTP para recibir la hora
  Serial.println("Obtener la hora de internet ...");

  while (!time(nullptr)) { // esperar a que se reciba la hora del servidor NTP
    Serial.print("*");
    delay(500);
  }
  Serial.println("Comentarios recibidos!");
 } else if (WiFi.waitForConnectResult() == WL_CONNECT_FAILED) { // Si la ESP32-
CAM no se conecta a su red WiFi
  Serial.println("La contraseña de la red WiFi es incorrecta :(");
 }
}
```

```
void loop() {

  if (WiFi.status() == WL_CONNECTED) { // si la tarjeta ESP32-CAM está conectada a su red WiFi
    if (videoRecorded) { // si la grabación de vídeo está completa
      videoRecorded = false; // restablecer que la grabación de vídeo ha finalizado
      videoBuffer = psramVideoBuffer; // recuperar el buffer de vídeo de la PSRAM
      videoPtr = 0;
      videoLength = psramVideoPtr - psramVideoBuffer;

      do { // делаем
        time_t now = time(nullptr); // obtener la hora del servidor NTP
        struct tm* p_tm = localtime(&now); // escribir la hora en la estructura

        hours = String(p_tm->tm_hour + 1); // registrar las horas
        minutes = String(p_tm->tm_min);    // registrar las actas
        days = String(p_tm->tm_mday);      // registrar los días desde el comienzo del mes
        month = String(p_tm->tm_mon + 1);  // registrar el número del mes
        year = p_tm->tm_year + 1900;       // registrar el año
      } while (year == 1900 or year == 1970); // hasta que la hora sea correcta

      if (hours.toInt() < 10) { // si el reloj es más pequeño 10
        hours = "0" + hours; // añadir el cero al reloj
      }
      if (minutes.toInt() < 10) { // si los minutos son inferiores a 10
        minutes = "0" + minutes; // añadir un cero a los minutos
      }
      if (days.toInt() < 10) { // si los días transcurridos desde el inicio del mes son inferiores a 10
        days = "0" + days; // añadir cero a los días desde el principio del mes
      }
      if (month.toInt() < 10) { // si el número del mes es inferior a 10
        month = "0" + month; // añadir un cero al número del mes
      }

      videoFileName = hours + "." + minutes + " " + days + "-" + month + "-" + String(year) + ".avi"; //introduzca el nombre del vídeo que enviaremos a Telegram

      Serial.println("Enviar un video a Telegram ...");
      if (motionDetected) { // si se detecta movimiento
        // enviar un Telegram al usuario que activó la alarma, una acción de chat sobre el vídeo que se envía a Telegram
        bot.sendChatAction(alarmChatID, "upload_video");
        // enviar un archivo de vídeo de Telegram al usuario que ha provocado la alarma
        bot.sendMultipartFormDataToTelegramWithCaption("sendDocument", "document", videoFileName, "image/jpeg",
          "Movimiento Detectado!", alarmChatID, psramVideoPtr - psramVideoBuffer, videoMore, videoNext, nullptr, nullptr);
      } else {
        // enviar un Telegram al usuario que envió el comando para grabar y enviar el vídeo, una acción de chat sobre el vídeo que se envía a Telegram
        bot.sendChatAction(videoChatID, "upload_video");
        //enviar el archivo de vídeo de Telegram al usuario que envió la orden de grabar y enviar el vídeo
        bot.sendMultipartFormDataToTelegram("sendDocument", "document", videoFileName,
```

```
"image/jpeg",
                                videoChatID, psramVideoPtr - psramVideoBuffer, videoMore,
videoNext, nullptr, nullptr);
     }
     motionDetected = false; // restablecer qué movimiento se detecta
   }

   if (isAlarmRunning) { // si se activa una alarma de movimiento
     if (millis() - pirSensorDelayMillis > pirSensorDelay * 1000) { // hacer un control de
intervalos para ver si se detecta movimiento
       pirSensorDelayMillis = millis(); // registrar la hora para comprobar si se detecta el
movimiento

       if (digitalRead(pirSensorPin)) { // si un sensor PIR ha detectado movimiento
         motionDetected = true; // registrar el movimiento detectado
         if (alarmMessageType == "Foto") { // si el tipo de mensaje de alarma es una foto
           sendPhotoToTelegram(); // enviar fotos a Telegram
         } else if (alarmMessageType == "Video") { // si el tipo de mensaje de alarma es vídeo
           sendVideoToTelegram(); // Grabar y enviar vídeo a Telegram
         } else if (alarmMessageType == "Texto") { // si el tipo de mensaje de alarma es de
texto
           bot.sendMessage(alarmChatID, "Movimiento Detectado!", "Markdown");
           motionDetected = false; // restablecer qué movimiento se detecta
         }
       }
     }
   }

   if (millis() > lastMessageMillis + getMessageInterval) { // hacer el intervalo de recepción
de mensajes de un usuario de Telegram
     int newMessageCount = bot.getUpdates(bot.last_message_received + 1); // registrar el
número de mensajes no leídos
     while (newMessageCount) {
       for (int i = 0; i < newMessageCount; i++) {
         Serial.println("Respuesta recibida del usuario de Telegram!");
         chatID = String(bot.messages[i].chat_id); // registra el identificador de chat del usuario
de Telegram que envió el mensaje a tu bot de Telegram
         if (isChatIDMatches()) { // si el identificador de chat del usuario de Telegram que ha
enviado el mensaje a tu bot de Telegram coincide con uno de los identificadores de chat que
has especificado
           message = bot.messages[i].text; // grabar un mensaje enviado por un usuario de
Telegram a tu bot de Telegram

           if (message == "/takephoto") sendPhotoToTelegram(); // si se recibe el comando
"/takephoto", envía la foto a Telegram
           if (message == "/recordvideo") { // si se recibe el comando "/recordvideo"
             videoChatID = chatID; // graba el identificador del chat del usuario de Telegram que
ha enviado la orden de grabar y envía el vídeo a tu bot de Telegram
             sendVideoToTelegram(); // Grabar y enviar vídeo a Telegram
           }

           if (message == "/flash") { // si se recibe el comando "/flash"
             flashState = !flashState; // cambiar el estado actual del flash
             digitalWrite(flashPin, flashState); // encender/apagar el flash
             if (flashState) { // si el flash está activado
```

```
      bot.sendMessage(chatID, "El flash está encendido!", "Markdown");
     } else {
      bot.sendMessage(chatID, "El flash está apagado!", "Markdown");
     }
    }




     if (message == "/pantt1") { // si se recibe el comando "/pant"


      for(int angle = 0; angle <= angleMax; angle +=angleStep) {
       servo1.write(angle);
       Serial.println(angle);
       delay(1000);
       bot.sendMessage(chatID, "cambio", "Markdown");
       }

      for(int angle = 180; angle >= angleMin; angle -=angleStep) {
      servo1.write(angle);
      Serial.println(angle);
      delay(1000);
       }



     pantt1State = !pantt1State; // cambiar el estado actual del flash
     digitalWrite(pantt1Pin, pantt1State); // encender/apagar el flash
     if (pantt1State) { // si el flash está activado
      bot.sendMessage(chatID, "El pantt1 está encendido!", "Markdown");
     } else {
      bot.sendMessage(chatID, "El pantt1 está apagado!", "Markdown");
     }

    }







    if (message == "/alarm") { // si se recibe el comando "/alarma"
     isAlarmRunning = !isAlarmRunning; // cambiar el estado de la alarma de tráfico
     if (isAlarmRunning) { // si se activa una alarma de movimiento
      alarmRunning = "siii1"; // registrar que se ha activado una alarma de movimiento
      alarmChatID = chatID; // registrar el identificador del chat de Telegram del usuario
que envió la orden de iniciar/detener la alarma de movimiento
      saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
      bot.sendMessage(chatID, "La alarma se activa!", "Markdown");
     } else {
      alarmRunning = "Нет"; // registrar que la alarma de tráfico está detenida
      saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
      bot.sendMessage(chatID, "La alarma se detiene!", "Markdown");
     }
```

```
        }

        if (message == "/imagesettings") { // si se recibe el comando "/imagesettings"
          // escribir un mensaje para mostrar y establecer los parámetros de la imagen
          settingsMessage  = "Resolucion: " + framesize + "\n";
          settingsMessage += "Luminosidad: " + brightness + "\n";
          settingsMessage += "Voltear: " + flipImage + "\n";
          settingsMessage += "Efecto fotografico: " + specialEffect + "\n";
          settingsMessage += "Balance de blancos: " + whiteBalanceMode + "\n";
          settingsMessage += "Encender el flash: " + turnOnFlash + "\n";
          settingsMessage += "/changeimagesettings - restablecer la imagen";
          bot.sendMessage(chatID, settingsMessage, "Markdown");
        }
        if (message == "/changeimagesettings") { // si se recibe el comando
"/changeimagesettings"
          // escribir el teclado de Telegram en formato JSON para seleccionar el parámetro de
la imagen
          keyboardJson = "[[\"Resolucion\", \"Luminosidad\", \"Voltear\"],[\"efecto especial\",
\"Balance de blancos\", \"si se enciende el flash\"]]";
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione una opcion de imagen",
"", keyboardJson, true);
        }
        if (message == "Resolucion") { // si se recibe el mensaje "Permiso"
          keyboardJson = "[[\"XGA\", \"SVGA\", \"VGA\"],[\"CIF\", \"QVGA\"]]"; // escribir
un teclado de Telegram en formato JSON para seleccionar la resolución de la imagen
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione la resolucion de la
imagen", "", keyboardJson, true);
        }
        if (message == "Luminosidad") { // si se recibe el mensaje "Brillo"
          keyboardJson = "[[\"200%\", \"150%\", \"100%\"],[\"75%\", \"50%\"]]"; // escribir
un teclado de Telegram en formato JSON para seleccionar el brillo de la imagen
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione el brillo de la imagen", "",
keyboardJson, true);
        }
        if (message == "efecto especial") { // si se recibe el mensaje "Efecto especial"
          // escribir el teclado de Telegram en formato JSON para seleccionar un efecto de
imagen especial
          keyboardJson = "[[\"comun\", \"Negativo\", \"Gris\", \"Rojo\"],[\" Verde\", \" Azul\",
\" Sepia\"]]";
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione un efecto especial de la
imagen", "", keyboardJson, true);
        }
        if (message == "Balance de blancos") { // si se recibe el mensaje "Balance de blancos"
          // escribir el teclado de Telegram en formato JSON para seleccionar el modo de
balance de blancos de la imagen
          keyboardJson = "[[\"Balance Automatico\", \" Solar\", \"Nuboso\"],[\"Oficina\",
\"Hogar\"]]";
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione el balance de blancos de
la imagen", "", keyboardJson, true);
        }
        if (message == "si se enciende el flash") { // si se recibe el mensaje "Debe encenderse
el flash".
          setImageFlipMode = false; // registrar que el usuario de Telegram envió un comando
para establecer si el flash se encenderá para la foto
          keyboardJson = "[[\"siii1\", \"Нет\"]]"; // escribir un teclado de Telegram en formato
```

```
JSON para seleccionar si el flash se encenderá para la foto
          bot.sendMessageWithReplyKeyboard(chatID, "se encedera el flash para la foto", "",
keyboardJson, true);
         }
         if (message == "Voltear") { // si se recibe un mensaje de "Vuelta"
          setImageFlipMode = true; // registrar que el usuario de Telegram envió un comando
para establecer si la imagen se voltea
          keyboardJson = "[[\"siii1\", \"Нет\"]]"; // escribir el teclado de Telegram en formato
JSON para seleccionar si la imagen debe ser volteada
          bot.sendMessageWithReplyKeyboard(chatID, "se invertira la imagen", "",
keyboardJson, true);
         }

         if (message == "XGA" or message == "SVGA" or message == "VGA" or message ==
"CIF" or message == "QVGA") { // si se obtiene la resolución de la imagen
          bot.sendMessage(chatID, "La resolucion de la imagen se ajusta!", "Markdown");
          framesize = message; // registrar la resolución de la imagen
          setImageFramesize(); // establecer la resolución de la imagen
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
         }

         if (message == "200%" or message == "150%" or message == "100%" or message ==
"75%" or message == "50%") { // si se obtiene la luminosidad de la imagen
          bot.sendMessage(chatID, "El brillo de la imagen se ajusta!", "Markdown");
          brightness = message; // registrar el brillo de la imagen
          setImageBrightness(); // ajustar el brillo de la imagen
          saveSettingsToSPIFFS(); //  escribir parámetros en la memoria SPIFFS
         }

         if (message == "comun" or message == "Negativo" or message == "Gris" or message
== "Rojo" or message == " Azul" or message == " Sepia") { // si se obtiene un efecto de
imagen
          bot.sendMessage(chatID, "Efecto de imagen especial instalado!", "Markdown");
          specialEffect = message; // grabar un efecto de imagen especial
          setImageSpecialEffect(); // establecer un efecto de imagen especial
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
         }

         if (message == "Balance Automatico" or message == " Solar" or message ==
"Nuboso" or message == "Oficina" or message == "Hogar") { // si se obtiene el modo de
balance de blancos de la imagen
          bot.sendMessage(chatID, "El balance de blancos de la imagen se ajusta!",
"Markdown");
          whiteBalanceMode = message; // registrar el modo de balance de blancos de la
imagen
          setImageWhiteBalanceMode(); // establecer el modo de balance de blancos de la
imagen
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
         }

         if ((message == "siii1" or message == "Нет") and !setImageFlipMode) { // si se
recibe, se encenderá el flash de la foto
          turnOnFlash = message; // registrar si el flash se encenderá o no para la foto
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
          bot.sendMessage(chatID, "El encendido de flash para la foto se ajusta!",
```

```
"Markdown");
        }

        if ((message == "siii1" or message == "Нет") and setImageFlipMode) { // si se recibe,
se invertirá la imagen
          flipImage = message; // registrar si la imagen será volteada
          setImageFlip(); // establecer si la imagen debe ser volteada
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
          bot.sendMessage(chatID, "Juego de inversión de imagenes!", "Markdown");
        }

        if (message == "/videospeed") { // si se recibe el comando "/videospeed"
          // grabar un mensaje para mostrar y ajustar la aceleración del vídeo
          settingsMessage  = "aceleración de vídeo : " + videoSpeedString + "\n";
          settingsMessage += "/changevideospeed - cambiar la aceleracion de video";
          bot.sendMessage(chatID, settingsMessage, "Markdown");
        }
        if (message == "/changevideospeed") { // si se recibe el comando
"/changevideospeed"
          keyboardJson = "[[\"0.5x - 25fps\", \"1x - 8fps\"],[\"2.5x - 4fps\", \"5x - 2fps\"]]"; //
grabar el teclado de Telegram en formato JSON para configurar la aceleración de vídeo
          bot.sendMessageWithReplyKeyboard(chatID, "seleccione la aceleracion del video",
"", keyboardJson, true);
        }

        if (message == "0.5x - 25fps" or message == "1x - 8fps" or message == "2.5x - 4fps"
or message == "5x - 2fps") { // si se obtiene la aceleración de vídeo
          videoSpeedString = message.substring(0, message.indexOf(" ")); // grabación de la
aceleración del vídeo
          setVideoSpeed(); // establecer la aceleración de vídeo
          saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
          bot.sendMessage(chatID, "La aceleracion del video se ajusta!", "Markdown");
        }

        if (message == "/alarmsettings") { // si se recibe el comando "/alarmsettings"
          // escribir un mensaje para mostrar y establecer los parámetros de la alarma de
movimiento
          settingsMessage  = "Retraso del sensor PIR: " + String(pirSensorDelay) + " сек\n";
          settingsMessage += "Tipo de mensaje: " + String(alarmMessageType) + "\n";
          settingsMessage += "La alarma se activa: " + alarmRunning + "\n";
          settingsMessage += "/changealarmsettings - Cambiar la configuracion de la alarma";
          bot.sendMessage(chatID, settingsMessage, "Markdown");
        }
        if (message == "/changealarmsettings") { // si se recibe el comando
"/changealarmsettings"
          keyboardJson = "[[\"Retraso del sensor\", \"Tipo de mensaje\"]]"; // escribir un
teclado de Telegram en formato JSON para seleccionar el parámetro de alarma de
movimiento
          bot.sendMessageWithReplyKeyboard(chatID, "Seleccione una opcion de alarma",
"", keyboardJson, true);
        }

        if (setPirSensorDelayMode) { // si se recibe un retraso del sensor PIR
          if (isMessageNumber()) { // si el mensaje recibido es un número
            if (message.toInt() >= 3 and message.toInt() <= 120) { // si el retardo del sensor
```

```
PIR recibido es superior a 3 pero inferior a 120 segundos
              setPirSensorDelayMode = false; // Salir del modo de ajuste del retardo del sensor
PIR
              pirSensorDelay = message.toInt(); // registrar el retardo del sensor PIR
              saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
              bot.sendMessage(chatID, "Retraso del sensor PIR se ajusta!", "Markdown");
             } else {
              bot.sendMessage(chatID, "El retardo del sensor PIR debe estar entre 3 y 120
segundos!", "Markdown");
             }
            } else {
             bot.sendMessage(chatID, "El retraso del sensor PIR debe ser un número!",
"Markdown");
            }
           }
           if (message == "/cancel") { // si se recibe el comando "/cancel".
            setPirSensorDelayMode = false; // Salir del modo de ajuste del retardo del sensor
PIR
            bot.sendMessage(chatID, "Ha salido del modo de ajuste del retardo del sensor PIR!",
"Markdown");
           }

           if (message == "Retraso del sensor") { // si se recibe el mensaje "Retraso del sensor".
            setPirSensorDelayMode = true; // entrar en el modo de ajuste del retardo del sensor
PIR
            settingsMessage  = "Enviar retraso del sensor PIR en segundos (3 - 120)\n";
            settingsMessage += "/cancel - salir del modo de ajuste del retardo del sensor PIR";
            bot.sendMessage(chatID, settingsMessage, "Markdown");
           }
           if (message == "Tipo de mensaje") { // si se recibe el mensaje "Tipo de mensaje
            keyboardJson = "[[\"Foto\", \"Video\", \"Texto\"]]"; // escribir un teclado de
Telegram en formato JSON para establecer el tipo de mensaje de alarma de movimiento
            bot.sendMessageWithReplyKeyboard(chatID, "Seleccione el tipo de mensaje", "",
keyboardJson, true);
           }

           if (message == "Foto" or message == "Video" or message == "Texto") { // si se recibe
un tipo de mensaje de alarma de tráfico
            alarmMessageType = message; // registrar el tipo de mensaje de alarma de
movimiento
            saveSettingsToSPIFFS(); // escribir los parámetros en la memoria SPIFFS
            bot.sendMessage(chatID, "Tipo de mensaje de alarma establecido!", "Markdown");
           }
          } else {
           bot.sendMessage(chatID, "Su identificador de chat no coincide!", "Markdown");
          }
         }
         newMessageCount = bot.getUpdates(bot.last_message_received + 1); // actualizar el
número de mensajes no leídos
       }
       lastMessageMillis = millis(); // registrar la hora del último mensaje recibido de Telegram
del usuario
     }
   } else {
     WiFi.reconnect(); // volver a conectarse a su red WiFi
```

```
    client.setInsecure(); // Desactiva los certificados HTTPS para conectarte a tu bot de
Telegram
    bot.longPoll = 3; // establece cuánto tiempo esperará el bot de Telegram antes de devolver
los "mensajes ahora" en segundos
    bot.setMyCommands(botCommands); // Configuración de los comandos del bot de
Telegram

    configTime(timezone * 3600, 3600, "pool.ntp.org"); // establecer la zona horaria y la
dirección del servidor NTP para recibir la hora
    Serial.println("Obtener la hora de internet ...");

    while (!time(nullptr)) { // esperar a que se reciba la hora del servidor NTP
      Serial.print("*");
      delay(500);
    }
    Serial.println("Comentarios recibidos!");
  }
}

bool isMoreDataAvailable() {
  return (fbLength - currentByte);
}

uint8_t getNextByte() {
  currentByte++;
  return (fbBuffer[currentByte - 1]);
}

bool videoMore() {
  return (videoLength - videoPtr);
}

uint8_t videoNext() {
  videoPtr++;
  return (videoBuffer[videoPtr - 1]);
}

struct frameSizeStruct { // una estructura para almacenar los tamaños de resolución de las
imágenes
  uint8_t frameWidth[2];  // la anchura de una imagen con una resolución determinada
  uint8_t frameHeight[2]; // la altura de una imagen con una resolución determinada
};

static const frameSizeStruct frameSizeData[] = { // estructura para almacenar el buffer de
resolución de imagen
  {{0x40, 0x01}, {0xF0, 0x00}}, // permiso QVGA, 320x240
  {{0x90, 0x01}, {0x28, 0x01}}, // permiso  CIF,  400x296
  {{0x80, 0x02}, {0xE0, 0x01}}, // permiso  VGA,  640x480
  {{0x20, 0x03}, {0x58, 0x02}}, // permiso SVGA, 800x600
  {{0x00, 0x04}, {0x00, 0x03}}  // permiso  XGA,  1024x768
};

uint8_t buf[240] = {
  0x52, 0x49, 0x46, 0x46, 0xD8, 0x01, 0x0E, 0x00, 0x41, 0x56, 0x49, 0x20, 0x4C, 0x49,
```

```
0x53, 0x54,
  0xD0, 0x00, 0x00, 0x00, 0x68, 0x64, 0x72, 0x6C, 0x61, 0x76, 0x69, 0x68, 0x38, 0x00,
0x00, 0x00,
  0xA0, 0x86, 0x01, 0x00, 0x80, 0x66, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00,
0x00, 0x00,
  0x64, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
  0x80, 0x02, 0x00, 0x00, 0xe0, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4C, 0x49, 0x53, 0x54, 0x84, 0x00,
0x00, 0x00,
  0x73, 0x74, 0x72, 0x6C, 0x73, 0x74, 0x72, 0x68, 0x30, 0x00, 0x00, 0x00, 0x76, 0x69,
0x64, 0x73,
  0x4D, 0x4A, 0x50, 0x47, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0A, 0x00,
0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x73, 0x74,
0x72, 0x66,
  0x28, 0x00, 0x00, 0x00, 0x28, 0x00, 0x00, 0x00, 0x80, 0x02, 0x00, 0x00, 0xe0, 0x01,
0x00, 0x00,
  0x01, 0x00, 0x18, 0x00, 0x4D, 0x4A, 0x50, 0x47, 0x00, 0x84, 0x03, 0x00, 0x00, 0x00,
0x00, 0x00,
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x49, 0x4E,
0x46, 0x4F,
  0x10, 0x00, 0x00, 0x00, 0x6A, 0x61, 0x6D, 0x65, 0x73, 0x7A, 0x61, 0x68, 0x61, 0x72,
0x79, 0x20,
  0x76, 0x38, 0x38, 0x20, 0x4C, 0x49, 0x53, 0x54, 0x00, 0x01, 0x0E, 0x00, 0x6D, 0x6F,
0x76, 0x69,
};

static void inline printQuartet(unsigned long i, uint8_t * quartetFb) {
  uint8_t y[4];
  y[0] = i % 0x100;
  y[1] = (i >> 8) % 0x100;
  y[2] = (i >> 16) % 0x100;
  y[3] = (i >> 24) % 0x100;
  memcpy(quartetFb, y, 4);
}

static void inline printOctet(unsigned long i, unsigned long j, uint8_t * octetFb) {
  uint8_t y[8];
  y[0] = i % 0x100;
  y[1] = (i >> 8) % 0x100;
  y[2] = (i >> 16) % 0x100;
  y[3] = (i >> 24) % 0x100;
  y[4] = j % 0x100;
  y[5] = (j >> 8) % 0x100;
  y[6] = (j >> 16) % 0x100;
  y[7] = (j >> 24) % 0x100;
  memcpy(octetFb, y, 8);
}

// Función de tarea de freeRTOS para la grabación de vídeo
void videoRecordingTask(void* taskParameters) {
```

```
  videoFb = getGoodPicture(); // obtener y grabar un fotograma para el vídeo
  if (!videoFb) { // si no fuera posible conseguir un fotograma para el vídeo
    Serial.println("¡No pude conseguir un fotograma para el vídeo!");
    return;
  } else {
    frameCount = 0; // restablecer el número de fotogramas del vídeo
    videoStartMillis = millis(); // registrar la hora de inicio del vídeo
    Serial.println("Empezar a grabar un vídeo!");

    nextFb = getGoodPicture(); // recuperar y grabar el siguiente fotograma del vídeo
    lastFrameMillis = millis(); // registrar la hora del último fotograma del vídeo
    startVideoRecording(); // empezar a grabar vídeo

    for (int j = 0; j <= 148 ; j++) { // grabar vídeo siempre que el número de fotogramas sea
inferior a 149
      currentFrameMillis = millis(); // registrar la hora del fotograma de vídeo actual
      if (currentFrameMillis - lastFrameMillis < frameInterval) { // si el intervalo entre el
fotograma actual y el anterior es menor que el intervalo de fotogramas que ha establecido
        // retardo igual a la diferencia entre el intervalo entre la trama actual y la anterior y el
intervalo de tramas que hayas establecido
        delay(frameInterval - (currentFrameMillis - lastFrameMillis));
      }

      lastFrameMillis = millis(); // registrar la hora del último fotograma del vídeo
      frameCount++; // actualizar el número de fotogramas del vídeo

      if (frameCount != 1) esp_camera_fb_return(currentFb); // si se recibe al menos un
fotograma, guarda el fotograma de vídeo actual en la PSRAM
      currentFb = nextFb; // guardar el siguiente fotograma del vídeo en el fotograma actual
      writeFrameToVideo(currentFb); // grabar el fotograma actual del vídeo
      nextFb = getGoodPicture(); // recuperar y grabar el siguiente fotograma del vídeo

      if (videoSize > videoBufferSize * 0.95) break;
    }

    Serial.println("Finalización de la grabación de vídeo!");

    esp_camera_fb_return(currentFb); // guardar el fotograma de vídeo actual en la PSRAM
    frameCount++; // actualizar el número de fotogramas del vídeo

    currentFb = nextFb; // guardar el siguiente fotograma del vídeo en el fotograma actual
    nextFb = NULL; // restablecer el siguiente fotograma del vídeo
    writeFrameToVideo(currentFb); // grabar el fotograma actual del vídeo
    esp_camera_fb_return(currentFb); // guardar el fotograma de vídeo actual en la PSRAM

    currentFb = NULL; // restablecer el fotograma actual del vídeo
    endVideoRecording(); // Terminar de grabar el vídeo
    videoEndMillis = millis(); // registrar la hora de finalización del vídeo

    fps = 1.0 * frameCount / ((videoEndMillis - videoStartMillis) / 1000); // grabación de
vídeo en fps
    frameCount = 0; // restablecer el número de fotogramas del vídeo
    videoRecorded = true; // registrar que la grabación de vídeo ha finalizado
    delay(500);
    vTaskDelete(NULL); // borrado de la tarea de grabación de vídeo de freeRTOS
```

```
  }
}

// función para iniciar la grabación de vídeo
static esp_err_t startVideoRecording() {
  psramVideoPtr = 0;
  psramIdxPtr = 0;

  memcpy(buf + 0x40, frameSizeData[framesizeInt].frameWidth,  2);
  memcpy(buf + 0xA8, frameSizeData[framesizeInt].frameWidth,  2);
  memcpy(buf + 0x44, frameSizeData[framesizeInt].frameHeight, 2);
  memcpy(buf + 0xAC, frameSizeData[framesizeInt].frameHeight, 2);

  psramVideoPtr = psramVideoBuffer;
  psramIdxPtr = psramIdxBuffer;

  memcpy(psramVideoPtr, buf, 240);
  psramVideoPtr += 240;
  startMillis = millis(); // registrar la hora de inicio de la grabación de vídeo

  jpegSize = 0;
  videoSize = 0;
  videoLength = 0;
  idxOffset = 4;
}

// función para finalizar la grabación de vídeo
static esp_err_t endVideoRecording() {
  if (frameCount < 10) { // si el número de fotogramas del vídeo es inferior a 5
    Serial.println("No se ha podido grabar el vídeo porque se han recibido menos de 5
fotogramas :(");
  } else {
    videoDuration = millis() - startMillis; // registrar la duración del vídeo en milisegundos

    realFPS = (1000.0f * (float)frameCount) / ((float)videoDuration) * videoSpeedFloat; //
grabar video fps real
    microsecondsPerFrame = 1000000.0f / realFPS; // registrar la duración de los fotogramas
en vídeo en microsegundos
    attainedFPS = round(realFPS); // redondear y registrar los fps de vídeo
    attainedPerFrame = round(microsecondsPerFrame); // redondear y registrar la duración del
cuadro en el vídeo en microsegundos

    printQuartet(videoSize + 240 + frameCount * 16 + 8 * frameCount, psramVideoBuffer +
4);
    printQuartet(attainedPerFrame, psramVideoBuffer + 0x20);

    unsigned long maxBytesPerSecond = (1.0f * videoSize * attainedFPS) / frameCount;
    printQuartet(maxBytesPerSecond, psramVideoBuffer + 0x24);
    printQuartet(frameCount, psramVideoBuffer + 0x30);
    printQuartet(frameCount, psramVideoBuffer + 0x8c);
    printQuartet((int)attainedFPS, psramVideoBuffer + 0x84);
    printQuartet(videoSize + frameCount * 8 + 4, psramVideoBuffer + 0xe8);

    memcpy (psramVideoPtr, idx1Buffer, 4);
    psramVideoPtr += 4;
```

```
    printQuartet(frameCount * 16, psramVideoPtr);
    psramVideoPtr += 4;
    psramIdxPtr = psramIdxBuffer;

    for (int i = 0; i < frameCount; i++) {
      memcpy (psramVideoPtr, dcBuffer, 4);
      psramVideoPtr += 4;
      memcpy (psramVideoPtr, zeroBuffer, 4);
      psramVideoPtr += 4;

      memcpy (psramVideoPtr, psramIdxPtr, 8);
      psramVideoPtr += 8;
      psramIdxPtr += 8;
    }
  }
}

// función para grabar un fotograma en el vídeo
static esp_err_t writeFrameToVideo(camera_fb_t * fb) {
  int fbLength; // una variable para almacenar el tamaño de los fotogramas en el vídeo
  fbLength = fb->len; // registrar el tamaño de los fotogramas en el vídeo
  jpegSize = fbLength; // grabar el tamaño del jpeg
  remnant = (4 - (jpegSize & 0x00000003)) & 0x00000003;

  fbBlockMillis = millis(); // registrar el tiempo de grabación del buffer de fotogramas en el
vídeo
  frameWriteMillis = millis(); // registrar el tiempo de grabación del fotograma en el vídeo
  memcpy(psramVideoPtr, dcBuffer, 4);
  jpegSizeRemnant = jpegSize + remnant;

  printQuartet(jpegSizeRemnant, psramVideoPtr + 4);
  fbBlockStart = fb->buf; // escribir un buffer de fotogramas en el vídeo
  if (fbLength > 8 * 1024 - 8 ) {
    fbBlockLength = 8 * 1024;
    fbLength = fbLength - (8 * 1024 - 8);
    memcpy(psramVideoPtr + 8, fbBlockStart, fbBlockLength - 8);
    fbBlockStart = fbBlockStart + fbBlockLength - 8;
  } else {
    fbBlockLength = fbLength + 8 + remnant;
    memcpy(psramVideoPtr + 8, fbBlockStart, fbBlockLength - 8);
    fbLength = 0; // poner a cero el tamaño de los fotogramas del vídeo
  }

  psramVideoPtr += fbBlockLength;
  while (fbLength > 0) {
    if (fbLength > 8 * 1024) {
      fbBlockLength = 8 * 1024;
      fbLength = fbLength - fbBlockLength;
    } else {
      fbBlockLength = fbLength  + remnant;
      fbLength = 0;
    }

    memcpy(psramVideoPtr, fbBlockStart, fbBlockLength);
```

```
    psramVideoPtr += fbBlockLength;
    fbBlockStart = fbBlockStart + fbBlockLength;
  }

  videoSize += jpegSize;
  videoLength += jpegSize;

  printOctet(idxOffset, jpegSize, psramIdxPtr);
  psramIdxPtr += 8;
  idxOffset = idxOffset + jpegSize + remnant + 8;
  videoSize += remnant;
}

// función para obtener una buena foto
camera_fb_t * getGoodPicture() {
  camera_fb_t * fb; // crear un objeto de almacenamiento de fotos
  do { // haciendo
    int fbLength; // una variable para almacenar el tamaño de la foto
    int foundFfd9;

    fb = esp_camera_fb_get(); // Recepción y grabación de fotos
    if (!fb) { // si no se ha podido obtener una foto
      Serial.println("No se ha podido conseguir un fotograma para el vídeo :(");
      getGoodPictureFailures++; // aumentar el número de intentos fallidos para conseguir una
buena foto
    } else {
      fbLength = fb->len; // registrar el tamaño de la foto

      for (int j = 1; j <= 1025; j++) {
        if (fb->buf[fbLength - j] != 0xD9) {
        } else {
          if (fb->buf[fbLength - j - 1] == 0xFF ) {
            foundFfd9 = 1;
            break;
          }
        }
      }

      if (!foundFfd9) {
        Serial.printf("Trama mala recibida, Número de trama %d, Tamaño %d bytes \n",
frameCount, fbLength);
        esp_camera_fb_return(fb);
        getGoodPictureFailures++;
      } else {
        break;
      }
    }
  } while (getGoodPictureFailures < 5); // mientras que el número de intentos fallidos para
conseguir una buena foto es inferior al 5
  if (getGoodPictureFailures == 5) { // si el número de intentos fallidos para conseguir una
buena foto es de 5
    Serial.printf("El número de intentos fallidos para conseguir una buena foto ha llegado al
5!");
    camera->set_quality(camera, camera->status.quality + 4); // reducir la calidad de la imagen
    delay(500);
```

```
}
return fb; // Traer una buena foto
```