

Cerradura magnética con reconocimiento facial de bajo costo

Low-cost facial recognition magnetic lock

Samuel Serrano¹, Nicholas Pitti², Ginnelle Gracia³

¹Facultad de Ingeniería de Sistemas Computacionales, Centro Regional Chiriquí, Universidad Tecnológica de Panamá

Resumen Se presenta un prototipo de cerradura magnética con reconocimiento facial que busca abordar el problema creciente de seguridad y automatización en el control de acceso a espacios físicos. El objetivo es desarrollar un sistema económico basado en la integración de los dispositivos electrónicos Raspberry Pi 4B y ESP32 WROOM. Se analizan soluciones existentes en el mercado y se comparan con el prototipo propuesto. Se describen las características del prototipo, incluyendo el diseño, la implementación y los resultados obtenidos. El uso de componentes de hardware asequibles y soluciones de software de código abierto permite ofrecer una solución más cómoda y adaptable a diferentes entornos y necesidades. El prototipo utiliza el algoritmo Haar Cascade Classifier para el reconocimiento facial. Se resalta la oportunidad de desarrollo del prototipo, así como la posibilidad de escalabilidad y adaptabilidad a futuras necesidades.

Palabras clave Clasificador en cascada Haar, ESP32, Open CV, Raspberry Pi, Sockets

Abstract A prototype magnetic lock with facial recognition is presented that seeks to address the growing problem of security and automation in access control to physical spaces. The objective is to develop an economical system based on the integration of the electronic devices Raspberry Pi 4B and ESP32 WROOM. Existing solutions in the market are analysed and compared with the proposed prototype. The features of the prototype are described, including the design, implementation and results obtained. The use of affordable hardware components and open-source software solutions allows offering a convenient and adaptable solution to different environments and needs. The prototype uses the Haar Cascade Classifier algorithm for facial recognition. The development opportunity of the prototype is highlighted, as well as the possibility of scalability and adaptability to future needs.

Keywords Haar cascade classifier, ESP32, Open CV, Raspberry Pi, Sockets

* Corresponding author: samuel.serrano@utp.ac.pa

1. Introducción

La tecnología de reconocimiento facial ha sido objeto de investigación y desarrollo durante varias décadas. Los primeros sistemas de reconocimiento facial se basaban en técnicas de análisis de características faciales, como la distancia entre los ojos, la forma de la nariz y la boca. Sin embargo, estos sistemas eran propensos a errores y no eran muy precisos. El programa Arduino proporciona una plataforma para prototipado ligero de proyectos con funcionalidades y componentes visuales [1]. Además, con el avance de la tecnología, se han desarrollado nuevos enfoques para el reconocimiento facial, como el uso de redes neuronales y algoritmos de aprendizaje automático. Estos sistemas son capaces de analizar patrones en las características faciales y aprender a identificar rostros con mayor precisión. [2]

El reconocimiento facial se ha utilizado en una variedad de aplicaciones, desde la seguridad y el control de acceso hasta la publicidad y el entretenimiento. En la industria de la seguridad, el reconocimiento facial se ha utilizado para identificar a

personas en aeropuertos, estadios y otros lugares públicos. También se ha utilizado en aplicaciones de control de acceso, como sistemas de cerraduras electrónicas y sistemas de control de acceso a edificios [2] y [3].

En los últimos años, el reconocimiento facial se ha vuelto más accesible gracias a la disponibilidad de componentes electrónicos de bajo costo, como cámaras y microcontroladores. Esto ha permitido a los desarrolladores crear sistemas de reconocimiento facial personalizados y de bajo costo, como el prototipo de cerradura magnética con reconocimiento facial descrito en este proyecto [2] y [3].

Raspberry Pi 4 B tiene un costo alrededor de B./100. Las especificaciones preferibles para desarrollar con Python en Raspberry son de 64bit con un sistema operativo de 64bit. Menos librerías y dependencias disponibles para 32bits además menos capacidad para utilizar ejecutar programas de reconocimiento. Debido a esta limitación se optó a usar "Haar Cascade Classifier", en vez de TensorFlow como método para reconocimiento facial. Raspberry posee la limitación de no poder descargar el software de Anaconda desde su sitio oficial

por lo que se instaló mamba y conda por terminal con el fin de tener las mismas funcionalidades de poder organizar dependencias en entornos virtuales. El objetivo de este proyecto es desarrollar un sistema de control de acceso basado en reconocimiento facial que sea económico.

La estructura del artículo se organiza con la introducción, donde se presenta el contexto del problema y se establecen los criterios y restricciones de la solución. A continuación, se revisarán las soluciones existentes, incluyendo referencias bibliográficas relevantes. Después, se describirá detalladamente el diseño, implementación y resultados obtenidos del prototipo de cerradura magnética con reconocimiento facial. Finalmente, se proporcionará una conclusión que resume los hallazgos claves.

2. Antecedentes

Se ha realizado una revisión de diferentes soluciones o proyectos similares. A continuación, se listan y explican cada uno de ellos.

El artículo titulado "Sistema de cerradura inteligente basado en reconocimiento facial" del Departamento de Ingeniería Eléctrica/Electrónica de la Universidad del Estado de Rivers, Nigeria. El sistema descrito en el artículo utiliza Raspberry Pi 3 B junto con una cámara Raspberry, una cerradura solenoide eléctrica, un relé, un regulador de voltaje y una pantalla de visualización (ver Figura 1). El algoritmo de reconocimiento facial utilizado se basa en Eigenface para la extracción de características y Análisis de Componentes Principales (PCA) como clasificador. La salida del algoritmo de reconocimiento facial está conectada a un circuito de relé que controla una cerradura magnética en la puerta. El artículo informa resultados prometedores con un 90% de precisión en el reconocimiento facial [5].

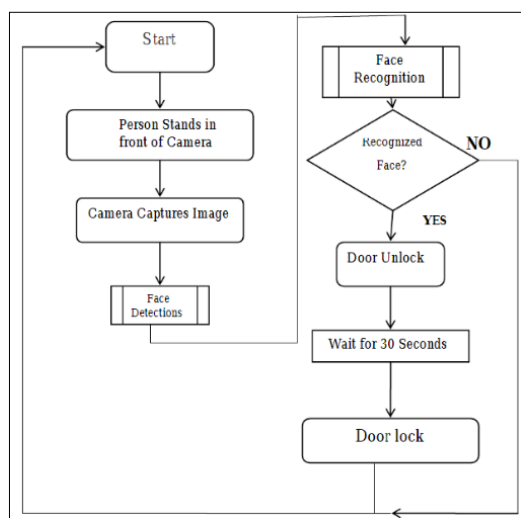


Figura 1. Diagrama de flujo de cerradura de puerta inteligente mediante reconocimiento facial [5].

Fuente: Facial Recognition Based Smart Door Lock System.

El segundo artículo es "Smart Door with Facial Recognition". El objetivo principal del proyecto es proporcionar comodidad y seguridad a través de la tecnología de hogar inteligente. El proyecto se basa en el uso de una cámara HD conectada a un monitor frente a la puerta, que permite detectar y reconocer caras (ver Figura 2), realizado por el funcionamiento sobre Microsoft Visual Studio IDE para permitir el acceso a personas autorizadas [6].

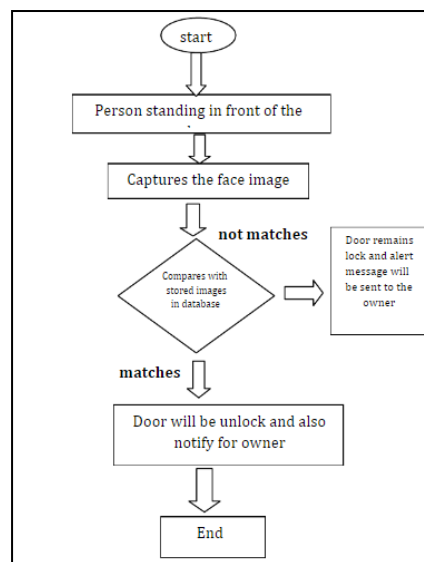


Figura 2. Diagrama de flujo para el proceso de reconocimiento facial [6].

Fuente: Smart Door with Facial Recognition.

En el artículo "Automatic Door Lock System by Face Recognition" propuesto por Sharvani Yedulapuram, Rajeshwarrao Arabelli, Kommabatla Mahender y Chintaju Sidhardha, se presenta un sistema de cerradura automática con reconocimiento facial utilizando Raspberry Pi para propósitos de seguridad. Este propone un sistema de cerradura automática con reconocimiento facial utilizando Raspberry Pi, cámara Pi y algoritmo del método Haar para la detección facial (ver Figura 3). El sistema ofrece una solución de seguridad activa al desbloquear la puerta cuando se detecta un rostro reconocido y enviar notificaciones con imágenes de intrusos a través de una red GSM y LAN cuando se detecta un rostro desconocido [7].

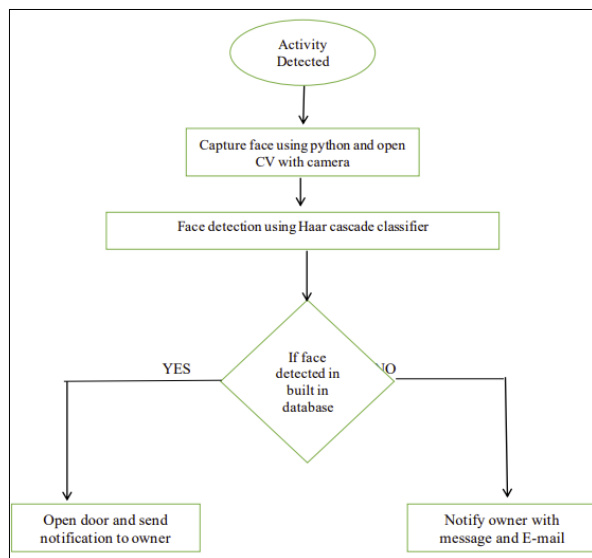


Figura 3. Diagrama de flujo del sistema [7].
Fuente: Automatic Door Lock System by Face Recognition.

3. Diseño y Metodología

Para llevar a cabo este prototipo se requiere de un LED rojo, una resistencia de 470Ω , un Raspberry Pi 4B con fuente de alimentación, un EMEET SmartCam C960 Webcam USB, un ESP-WROOM-32, un 7inch LCD Display-H con cable HDMI y cargador de celular USB mini A. El programa de control en el ESP32 se compiló en el IDE de Arduino en su versión 2.1.0 [1]. Se programó en el Raspberry con el lenguaje Python, debido a que este provee librerías como OpenCV para funciones y algoritmos para el procesamiento de imágenes y videos. Utilizando OpenCV con el método Haar se pueden analizar características visuales, como patrones de textura y cambios de intensidad, para identificar objetos específicos en una imagen.

Las tres funcionalidades de principales que se desarrollaron fueron el reconocimiento utilizando el método Haar, el de sockets para enviar comandos al ESP32 por una red local (ver Figura 4); y el de verificar que se podía enviar notificación con el API “CallMeBot” al número celular proporcionada [8]. Esta requiere de internet para enviar notificaciones, mientras que la conexión entre el Raspberry y el ESP32 en la misma red por sockets no lo requiere.

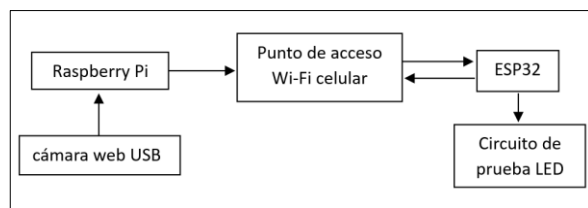


Figura 4. Conexiones del prototipo

La cámara web usada es de enchufar y usar por lo cual se podía usar en cualquier laptop e incluso en el Raspberry sin configuración adicional. Se utilizó código Python para hacer capturas a través de la cámara y guardarlas en un archivo de referencias. Luego para utilizar el método Haar solo se requiere del archivo XML de su repositorio GitHub y una ruta para acceder a las referencias.

Cabe destacar que el Raspberry Pi 4 Modelo B cuenta con una CPU ARM Cortex-A72 de cuatro núcleos [9]. Cada núcleo funciona a una velocidad de reloj de 1,5 GHz, lo que proporciona un rendimiento mejorado en comparación con sus modelos predecesores. El método Haar a diferencia de TensorFlow es principalmente un algoritmo de subproceso único. Procesa imágenes de forma secuencial y no es compatible inherentemente con subprocesos múltiples o procesamiento paralelo [10],[11] y [12].

Sin embargo, existen técnicas que puede emplear para aprovechar múltiples núcleos incluso cuando usa un algoritmo de subproceso único como el método Haar. Un enfoque es dividir los cuadros de imagen o video en regiones más pequeñas y procesar cada región en un núcleo diferente al mismo tiempo.

4. Resultados

4.1 Construcción y Validación del prototipo

El Raspberry Pi se le instaló un sistema operativo Debian de 64 bits utilizando un lector microSD a USB con Raspberry Pi Imager a través del ordenador usado en el proyecto. Luego utilizando teclado y ratón se procedió a la elaboración de código para capturar imágenes de referencia, reconocimiento a través del método Haar y finalmente la conexión por sockets al ESP32. El prototipo se encuentra en una fase funcional, cumpliendo con los requisitos establecidos de permitir un control de un componente conectado al ESP32 una vez se logra un reconocimiento facial (ver Figura 5).

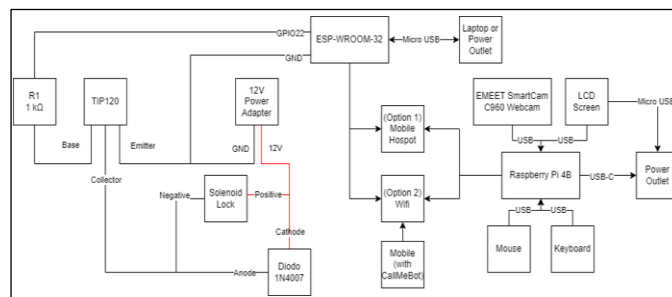


Figura 5. Diagrama de circuito del prototipo

Durante las pruebas, se observó que la conectividad del ESP32 a la red se logró proporcionando las credenciales de manera estática en el código, aunque esto no es una práctica de seguridad recomendada para implementaciones reales. Por otro lado, el Raspberry Pi 4B permitió la conexión a un punto

de acceso agregando las credenciales en el archivo de configuración localizado en "/etc/wpa_supplicant/". La IP del ESP32 se fijó de manera estática en el código de Arduino utilizando la subred del punto de acceso y luego se proporcionó de manera estática en el código de reconocimiento para permitir la conexión por socket [13],[14],[15],[16],[17] y [18]. En el código de reconocimiento, se ajustaron los siguientes parámetros para optimizar el rendimiento:

- **threshold:** Determina el mínimo porcentaje de similitud a una referencia para aceptar. Un valor más alto implica mayor estricto. El valor usado fue de 0.6 para pruebas y se recomienda un valor entre 0.7 a 0.8 para uso final.
- **minNeighbors:** Al aumentar este valor, se reducen los falsos positivos, pero el reconocimiento se vuelve más estricto. El valor usado fue de 3, lo que indica la cantidad mínima de vecinos que debe tener cada rectángulo de cara potencial para que se considere una detección válida.
- **minSize:** Establece el tamaño mínimo válido en píxeles de la cara que puede aparecer en la cámara. El valor usado fue de 50x50 píxeles para poder detectar caras lejanas de la cámara.

Las referencias capturadas funcionaron mejor con imágenes de alta calidad. Usando 40 imágenes de referencia el algoritmo mostró un porcentaje de similitud entre 60-70% con las referencias dadas, se comprobó que el modelo funciona bien si hay suficiente iluminación y un fondo blanco (ver Tabla 1) pero se debe usar ensayo y error para conseguir resultados mejores. Sin embargo, se podría mejorar la rapidez ya que hay menos velocidad de fotogramas una vez se detecta cualquiera cara en el marco de la cámara web. El tiempo para detectar la cara correcta y encender el LED demora entre 5-10 segundos. Se compararon los valores de similitud entre un usuario al que se le tomaron fotos de referencia y otros los cuales eran desconocidos (ver tabla 1). El prototipo en su estado preliminar solo detecta un usuario y rechaza cualquier otra cara que no cumple con 70% de similitud (ver Figura 5,6,7), pero se podría implementar código para desplegar el nombre de distintos usuarios que entran en el marco de la cámara.

Tabla 1. Pruebas de promedio de porcentaje de similitud de cara

#	Escenario de comparación	Similitud de cara sin referencia (%)	Similitud de cara con referencia (%)
1	no familiar de genero igual y edad similar	60	65
2	familiar de genero igual y edad mayor	61	64
3	familiar de genero opuesto y edad mayor	62	67
4	familiar de genero igual y edad menor	64	67
5	familiar de genero opuesto y edad menor	61	67

Según los datos recopilados hay un mayor porcentaje de similitud con el usuario al que se le tomaron imágenes de referencia aun cuando se comparaba con familiares. Cabe destacar que los resultados fueron tomados con pose neutral mirando a la cámara, iluminación buena y fondo blanco sin lentes u otro objeto obstruyendo la cara. El promedio se calculó utilizando salida de terminal con reconocimientos por unos 30 segundos.

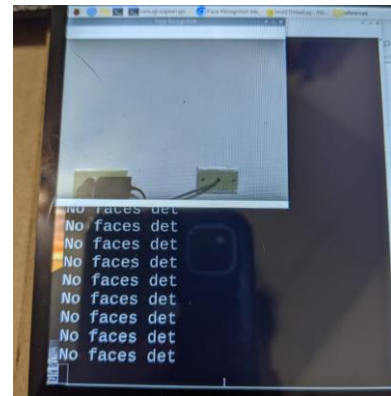


Figura 5. Ninguna detección indicada

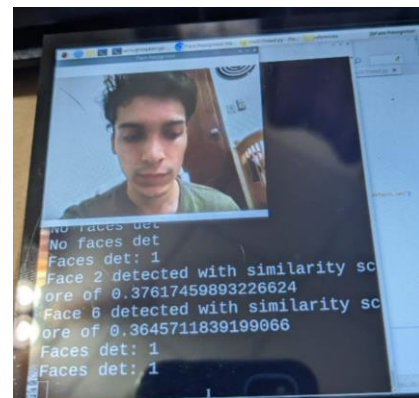


Figura 6. Detección con baja similitud

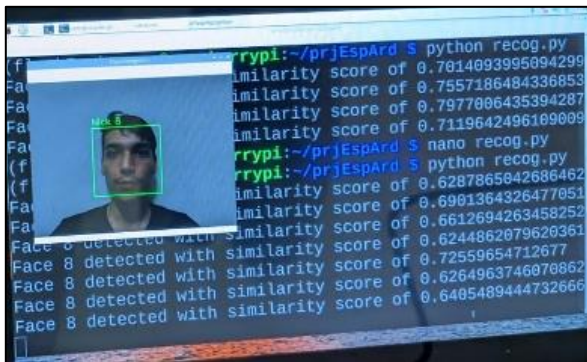


Figura 7. Detección con alta similitud

4.2 Oportunidades de desarrollo del prototipo

- Mercado objetivo: el prototipo está dirigido a empresas y organizaciones que requieren sistemas de reconocimiento facial para control de acceso y seguridad.
- Ventajas: el prototipo ofrece una solución de bajo costo y fácil implementación en comparación con otras soluciones en el mercado.
- Costos de implementación: los costos reales de implementación podrían ser bajos debido al uso de componentes económicos como el ESP32 y Raspberry Pi 4B; y además software y modelos gratuitos.
- Riesgos: la seguridad de la conexión a la red debe mejorarse para evitar vulnerabilidades.
- Análisis de costo-beneficio: el bajo costo de implementación y la eficacia del reconocimiento facial hacen que el prototipo sea una inversión rentable.
- Fuentes de financiamiento: se pueden buscar inversionistas interesados en tecnologías de reconocimiento facial o subvenciones gubernamentales para proyectos de innovación.

5. Conclusiones

Este trabajo aborda el problema de la seguridad y automatización en el control de acceso a espacios físicos mediante la implementación de un prototipo de cerradura magnética con reconocimiento facial. Las contribuciones de este estudio se consideran que podrían ser significativas para ofrecer más soluciones al mercado de seguridad y automatización.

Los resultados obtenidos en este estudio demuestran una oportunidad prometedora para posibles aplicaciones en diversos entornos donde se requiera un control de acceso seguro y automatizado.

La limitación de este proyecto recae en la necesidad de tener acceso a internet, un procesador potente para realizar el reconocimiento facial, hacer ensayo y error para mejorar las referencias, un mecanismo para alimentar los equipos

implementados y seguir políticas de seguridad donde se quiera implementar.

Se recomienda realizar investigaciones futuras para mejorar el rendimiento del sistema, por ejemplo, se podría aplicar las capacidades multihilo del Raspberry Pi 4B y así permitir aumentar las imágenes de referencia.

AGRADECIMIENTOS

Se le agradece al Ingeniero Anthony Castillo, por agilizar la compra del ESP32 proporcionando el suyo a un precio razonable. Además, cabe destacar que el Licenciado Jaime Palacio aportó una inmensa disponibilidad de compartir sus conocimientos técnicos y sugerencias durante varias semanas; y proporcionó el Raspberry de manera prestada hasta culminar el curso. Y finalmente la Licenciada Gloria Gutiérrez del Cid por aportar sus consejos para una implementación física del proyecto.

REFERENCIAS

- [1] Arduino, [En línea]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 6 3 2023].
- [2] L. Gómez y J. Escaño. "Sistema de seguridad con reconocimiento facial aplicado al acceso a un bloque de apartamentos". Repositorio de la Universidad de Zaragoza. <https://zaguan.unizar.es/record/96372/files/TAZ-TFG-2020-3987.pdf> (accedido el 4 de junio de 2023).
- [3] keronglock. "Habla sobre el desarrollo de cerraduras interesantes". kerong. <https://www.kerong.hk/post/habla-sobre-el-desarrollo-de-cerraduras-interesantes> (accedido el 4 de junio de 2023).
- [4] P. Elechi, E. Okowa y U. Ekwueme. "Facial Recognition Based Smart Door Lock System". Researchgate. https://www.researchgate.net/publication/360783137_Facial_Recognition_Based_Smart_Door_Lock_System (accedido el 4 de junio de 2023).
- [5] C. K. Gomathy, K. Keerthi y N. Pavithra. "Smart Door with Facial Recognition". Researchgate. https://www.researchgate.net/publication/357748499_Smart_Door_with_Facial_Recognition (accedido el 4 de junio de 2023).
- [6] S. Yedulapuram, R. Arabelli, K. Mahender y C. Sidhardha. "Automatic Door Lock System by Face Recognition". IOPscience. <https://iopscience.iop.org/article/10.1088/1757-899X/981/3/032036/pdf> (accedido el 4 de junio de 2023).
- [7] "Esp32 sending alert message to whatsapp - techtonions.com," TechTOnions, <https://www.techtonions.com/esp32-sending-alert-message-to-whatsapp/> (accedido el 5 de junio de 2023).
- [8] Raspberry Pi. "Raspberry Pi 4 Tech Specs". Raspberry Pi. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/> (accedido el 6 de junio de 2023).
- [9] Python. "Glossary". Python documentation. <https://docs.python.org/3/glossary.html#term-global-interpretor-lock> (accedido el 6 de junio de 2023).
- [10] OpenCV. "OpenCV and multi-threading: OpenCV, a popular computer vision library, supports multi-threading for certain operations. The OpenCV documentation provides information on how to utilize multi-threading capabilities within the

- library". OpenCV documentation index.
<https://docs.opencv.org/> (accedido el 6 de junio de 2023).
- [11] OpenCV. "Haar cascade classifier and single-threading: The Haar cascade classifier algorithm in OpenCV is a single-threaded approach. This is mentioned in the OpenCV documentation under the section". OpenCV documentation index. <https://docs.opencv.org/> (accedido el 6 de junio de 2023).
- [12] Platformio. "AI Thinker ESP32-CAM — PlatformIO latest documentation". Platformio.
<https://docs.platformio.org/en/latest/boards/espressif32/esp32cam.html#static-ip-address> (accedido el 6 de junio de 2023).
- [13] Arduino. "WiFi - Arduino Reference (WiFi Configuration)". Arduino - Home.
<https://www.arduino.cc/en/Reference/WiFiConfig> (accedido el 6 de junio de 2023).
- [14] RandomNerdTutorials. "ESP32 Static/Fixed IP Address | Random Nerd Tutorials". Random Nerd Tutorials.
<https://randomnerdtutorials.com/esp32-static-fixed-ip-address-arduino-ide/> (accedido el 6 de junio de 2023).
- [15] Real Python. "Socket Programming in Python (Guide) – Real Python". Python Tutorials – Real Python.
<https://realpython.com/python-sockets/> (accedido el 6 de junio de 2023).
- [16] Pymotw. "socket — Network Communication — PyMOTW 3". PyMOTW. <https://pymotw.com/3/socket/> (accedido el 6 de junio de 2023).
- [17] Python. "Socket Programming HOWTO". Python documentation. <https://docs.python.org/3/howto/sockets.html> (accedido el 6 de junio de 2023).