

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Universidad
Politécnica
de Cartagena

MIEMBRO DE



EUROPEAN
UNIVERSITY OF
TECHNOLOGY



Trabajo Fin de Grado

GRADO EN INGENIERÍA TELEMÁTICA

Sistema de monitorización y estimación de la ocupación de una biblioteca.

Programa Encuentro TF 2022-2023 empresa HOPU S.L.



AUTORA: Miriam Riouch Mhine

DIRECTOR: Alejandro Santos Martínez Sala

CO-DIRECTOR: Daniel Ros López (Hopu S.L.)

Septiembre / 2023



Universidad
Politécnica
de Cartagena

MIEMBRO DE



EUROPEAN
UNIVERSITY OF
TECHNOLOGY

Autor	Miriam Riouch Mhine
E-mail del autor	Miriamriouchmhine@gmail.com
Director / mail	Alejandro S. Martínez Sala / alejandros.martinez@upct.es
Co-director / mail	Daniel Ros López (Hopu S.L.) / d.ros@libelium.com
Título del TFG	Sistema de monitorización y estimación de la ocupación de una biblioteca.
Resumen	<p>La Biblioteca de la UPCT requiere un sistema de monitoreo del aforo en tiempo real para generar informes sobre la ocupación, gestionar el espacio y ofrecer un servicio de información de plazas disponibles para los estudiantes. La solución propuesta en este proyecto consiste en implementar un sistema de visión artificial en la entrada principal para llevar a cabo la detección y conteo de personas.</p> <p>Para llevar a cabo el conteo de personas mediante visión artificial, se ha llevado a cabo un estudio, evaluación y adaptación de soluciones de software libre basadas en Python, OpenCV y modelos de detección de objetos basados en redes neuronales de aprendizaje profundo pre-entrenadas.</p> <p>Además, se ha desarrollado un servidor con un backend y una base de datos dedicados a la recopilación y procesamiento de datos generados por el sistema de conteo. Para el servicio de información a los alumnos se integra en el servidor la gestión de un bot de Telegram que publica en un canal el dato de la ocupación en tiempo real. El servidor tiene un frontend de visualización de datos de la ocupación históricos y en tiempo real basado en Grafana.</p> <p>Se ha realizado un prototipo demostrativo y pruebas reales en la Biblioteca de Antigones de la ETSIT y se ha logrado un sistema de bajo coste, sencillo y con un reducido error que se propone al servicio CRAI de la UPCT para que lo lleven a producción. El servidor no almacena imágenes de personas ni registra ni guarda datos de carácter personal para garantizar la privacidad y el cumplimiento de la Ley Orgánica de Protección de Datos (LOPD).</p>
Titulación	Grado en Ingeniería Telemática.
Departamento	Tecnología de la Información y las Comunicaciones.
Fecha de presentación	Septiembre - 2023

Agradecimientos

Agradecer la ayuda del Programa Formación universitaria desde La empresa, «ENCUENTRO TF 2022/2023» financiado por la Dirección General de Universidades de la CARM.

A mi director de proyecto, Alejandro Santos Martínez Sala por darme la oportunidad de hacer este proyecto, por la ayuda en todo este proceso, por sus consejos como mentor. Y por enseñarme que con paciencia todo se puede sacar adelante.

A mi codirector, Daniel Ros, por su ayuda, su cercanía, su gran apoyo y por todo lo que me ha enseñado

A mis padres por educarme de la mejor manera, inculcándome valores fundamentales como el esfuerzo, la dedicación y la constancia, gracias por apoyarme de manera incondicional en todos mis sueños. A mi madre por siempre sentirse orgullosa de mí, por su paciencia y por la confianza que depositan en mí. A mi padre por enseñarme que no hay que rendirse, y que siempre hay que ir detrás de lo que uno quiere. Sin ellos no sería la persona que soy hoy en día.

A mis hermanas, por su apoyo, confianza y siempre estar ahí para mí. Sois todo para mí.

A Jose, por ser mi mayor apoyo y por sus palabras de ánimo en los momentos más frustrantes. Gracias por confiar siempre en mí, contigo a mi lado soy mejor.

A mi mejor amiga, Merche, gracias por acompañarme en todo este camino, estos 5 años no habrían sido igual sin ti.

A mis chicas, Mercedes, Eva, Jessica, Judit y Anabel, por esas tardes de estudio, esas cenas y comidas llenas de risas y momentos de desconexión. Gracias por siempre estar ahí.

Finalmente, gracias a mí, a mi esfuerzo, y por siempre seguir adelante a pesar de las adversidades.

Contenido

Capítulo 1. Introducción y objetivos	8
1.1 Introducción y objetivos	8
1.2 Herramientas software y elementos usados	9
1.3 Metodologías de trabajo	12
1.4 Estructura y organización del proyecto	14
Capítulo 2. Estado del arte de sistemas de visión para conteo de personas	15
2.1 Introducción	15
2.2 Fundamentos de sistemas de visión para conteo de personas	15
2.3 Métodos tradicionales de conteo de personas	16
2.4 Técnicas modernas de conteo de personas	17
2.5 Desafíos y problemáticas	18
2.6 Aplicaciones prácticas	19
2.7 Herramientas y plataformas disponibles	20
Capítulo 3. Cámara Tapo de TP-Link	22
3.1 Características	22
3.2 Instalación y configuración de la cámara Tapo	22
3.3 Pruebas de medición de fps y latencia del streaming	27
Capítulo 4. Sistema de visión para conteo de personas	29
4.1 COCO base de datos de imágenes para entrenamiento	29
4.2 Modelo YOLO3 de detección de clases de objetos	30
4.3 Modelo MobileNet SSD de detección de objetos	31
4.4 Sistema de tracking de objetos en una imagen	33
4.5 MobilNetSSD vs YOLO3	35
4.6 Pruebas y ajustes de parámetros del módulo de conteo y de la ubicación y orientación de la cámara	36
4.6.1 Ajuste de parámetros en los módulos de detección y tracking	37
4.6.2 Estudio de la posición y ángulo de inclinación de la cámara respecto al área de detección y paso	38
4.6.2 Conclusiones de las pruebas	44
Capítulo 5. Diseño e implementación del servidor del sistema de monitorización de la ocupación de la biblioteca	46
5.1 Arquitectura global del sistema	47
5.2 Arquitectura del backend y BBDD MySQL	49
5.3 Frontend de visualización de datos de conteo con Grafana	52
5.4 Servicio de monitorización mediante bot de Telegram	56
5.5 Recursos computacionales requeridos	59
5.6 Herramientas de gestión remota del servidor Zerotier y AnyDesk	60
5.7 Pruebas en el servidor	61
Capítulo 6. Conclusiones y trabajos futuros	63
6.1 Conclusiones	63
6.2 Grado de consecución de los objetivos y formación adquirida	64
6.3 Trabajos futuros	64
Bibliografía y referencias	66
Anexos	69
Anexo 1. Estructura y organización del código	69
Anexo 2. Librerías utilizadas	69

Índice de figuras

Figura 1. Plano de la biblioteca de Antigones	8
Figura 2. Entrada de la biblioteca.....	9
Figura 3. Logo de VLC	9
Figura 4. Logo Ubuntu	10
Figura 5. Logo de ZeroTier.....	10
Figura 6. Logo de GitHub.....	10
Figura 7. Logo de Visual Studio Code	10
Figura 8. Logo de Python.....	10
Figura 9. Logo de Docker	11
Figura 10. Logo de Grafana	11
Figura 11. Logo de MySQL.....	11
Figura 12. Logo de NodeJS y PM2	11
Figura 13. Logo de HTML.....	12
Figura 14. Logo de CSS	12
Figura 15. Logo de Telegram	12
Figura 16. Metodología de trabajo.....	13
Figura 17. Cámara Tapo TC70	22
Figura 18. App Tapo. Añadir cámara	23
Figura 19. App Tapo. Elección de cámara	23
Figura 20. App Tapo. Comprobación del LED de la cámara	23
Figura 21. App Tapo. Conectarse a la red Tapo	24
Figura 22. App Tapo. Vista en vivo	24
Figura 23. App Tapo. Ajustes de la cámara 1	25
Figura 24. App Tapo. Dirección IP de cámara	25
Figura 25. App Tapo. Ajustes de la cámara 2	26
Figura 26. App Tapo. Configuración Avanzada	26
Figura 27. App Tapo. Creación de cuenta para la cámara	27
Figura 28. Captura pruebas con modelo MobileNet SSD.....	28
Figura 29. COCO dataset	30
Figura 30. Arquitectura YOLO. Fuente [26].....	31
Figura 31. Filtros convolucionales estándar de (a) se sustituye por dos capas: convolución en profundidad (b) y la convolución puntual en (c) para construir un separable en profundidad. Fuente [29].....	32
Figura 32. Algoritmo SSD, funcionamiento mediante Redes Convolucionales, Fuente [31]	33
Figura 33. Algoritmo MobileNet SSD, Fuente [33].....	33
Figura 34. Posibles etiquetas de las clases.....	34
Figura 35. Posibles planos de fotografía	38
Figura 36. Esquema de posición de la cámara	39
Figura 37. Plano cenital, imágenes de la biblioteca	39
Figura 38. Plano normal, imágenes de la biblioteca	40
Figura 39. Plano picado, imágenes de la biblioteca 1	41
Figura 40. Plano picado, imágenes de la biblioteca 2	42
Figura 41. Plano picado, imágenes de la biblioteca 3	43
Figura 42. Esquema conexiones	46
Figura 43. Esquema general cámara y servidor	47
Figura 44. Esquema del sistema.....	48

Figura 45. Lista de programas gestionadas por PM2.....	49
Figura 46. Imagen de Dashboard PM2.....	49
Figura 47. Esquema del funcionamiento del programa de conteo Run.py.....	50
Figura 48. Columnas de la BBDD.....	51
Figura 49. Imagen de la cámara de la biblioteca.....	51
Figura 50. Diagrama de flujo programa envío de alertas por bots.....	52
Figura 51. Página de inicio de Grafana.....	52
Figura 52. Configuración de la conexión BBDD parte 1.....	53
Figura 53. Configuración de la conexión BBDD parte 2.....	53
Figura 54. Configuración de la conexión BBDD parte 3.....	53
Figura 55. Configuración de la BBDD parte 4.....	54
Figura 56. Configuración gráfica parte 1.....	54
Figura 57. Configuración gráfica parte 2.....	55
Figura 58. Configuración gráfica parte 3.....	55
Figura 59. Ejemplo grafica Grafana 25-07-2023.....	56
Figura 60. Bots de alumnos y administradores.....	56
Figura 61. Función para enviar mensajes por el bot de Administradores.....	57
Figura 62 ejemplo canal alumnos envío de N.º ocupación.....	57
Figura 63. Formulario para notificar cambios de horario.....	57
Figura 64. Chat ejemplo envío de cambio de horario.....	58
Figura 65. Diagrama de flujo envío de alertas cámara On/Off.....	58
Figura 66. Chat ejemplo envío alertas del estado de la cámara.....	59
Figura 67. Resultados del comando top.....	59
Figura 68. Resultados del comando atop.....	60
Figura 69. Servidor.....	60
Figura 70. Lista de programas gestionados por PM2 primer día.....	61
Figura 71. Lista de programas gestionados por PM2 cuarto día.....	61
Figura 72. Ejemplo Grafana 28-07-2023.....	62
Figura 73. Notificaciones de ocupación 28-07-2023.....	62
Figura 74. Notificación cámara no conectada 28-07-2023.....	62
Figura 75. Proyecto en GitHub.....	69

Índice de tablas

Tabla 1. Tabla MobileNet SSD vs YOLO3	35
Tabla 2. Resultados de las pruebas en plano cenital	40
Tabla 3. Resultados de las pruebas en plano normal.....	41
Tabla 4. Resultados de las pruebas en plano picado posición 1	42
Tabla 5. Resultados de las pruebas en plano picado posición 2	42
Tabla 6. Resultados de las pruebas en plano picado posición 2	43

Capítulo 1. Introducción y objetivos

1.1 Introducción y objetivos

El control y la monitorización de afluencia de personas en edificios es un servicio de gran relevancia para la gestión de espacios, el control climático, la seguridad, entre otros aspectos. En particular, las Bibliotecas de la UPCT realizan un seguimiento de aforo en sus instalaciones, tanto con fines estadísticos como para gestionar eficazmente el espacio como salas de estudio. Hasta la fecha, el servicio CRAI de la UPCT, que administra las tres bibliotecas físicas existentes, no cuenta con un sistema automatizado que proporcione datos en tiempo real sobre la ocupación. Actualmente, se realizan estimaciones basadas en registros manuales efectuados por el personal de la biblioteca en ciertos momentos del día, y estos datos se trasladan desde el papel a hojas de cálculo para la elaboración de informes. Además, durante los períodos de exámenes, la demanda de espacio de estudio es considerable, y no existe un método rentable que informe a los estudiantes sobre la disponibilidad de plazas en las bibliotecas.

En este contexto, la Biblioteca de la UPCT requiere un sistema de seguimiento en tiempo real de la ocupación de estudiantes, tanto para su propia gestión y generación de informes como para brindar información a los alumnos sobre la disponibilidad de plazas. Por tanto, el objetivo de este Trabajo de Fin de Grado (TFG) consiste en analizar esta problemática y desarrollar una prueba de concepto de bajo coste que permita evaluar la viabilidad de implementar un sistema de conteo de personas en la Biblioteca Antigones de la UPCT.

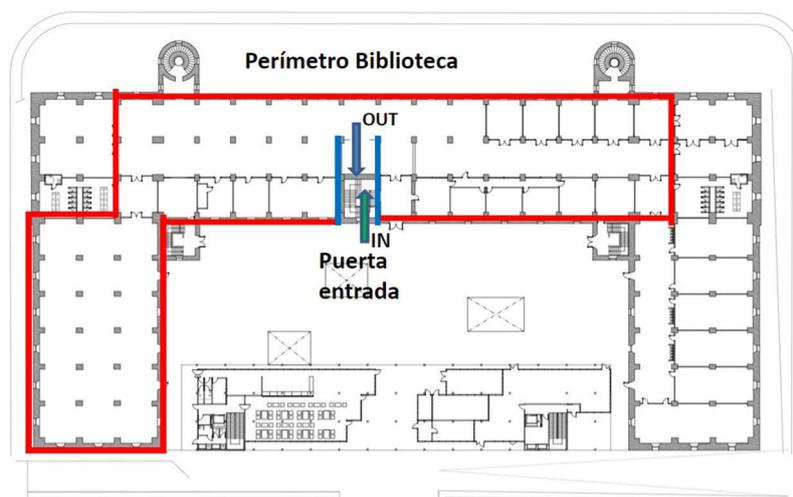


Figura 1. Plano de la biblioteca de Antigones

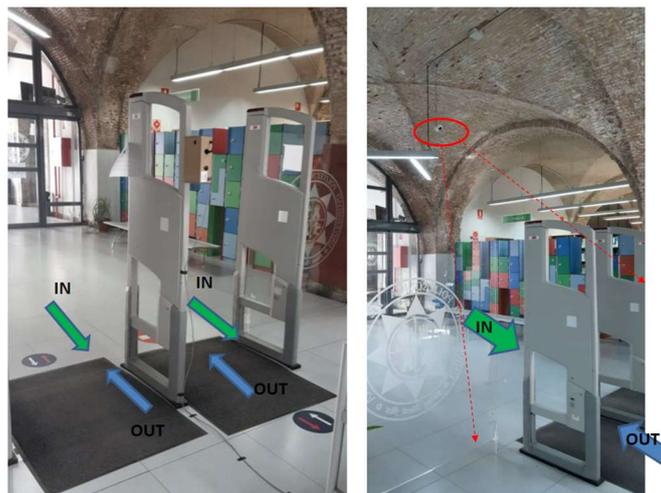


Figura 2. Entrada de la biblioteca

En la Figura 2, se muestra en la primera imagen la puerta de acceso a la biblioteca, y en la segunda imagen se puede observar la ubicación de la cámara, que está situada en la parte superior enfocando hacia la puerta. Estos arcos de entrada implementan un sistema antihurto de libros y es una zona de paso obligado para la entrada o salida de la Biblioteca de Antigonos. Por tanto, se ha escogido esa zona para el prototipo del sistema de detección y conteo de personas mediante visión artificial.

El objetivo fundamental de este proyecto es implementar y evaluar un sistema de conteo de personas basado en visión artificial que sea preciso y eficiente, permitiendo a la biblioteca llevar un registro detallado de la ocupación en tiempo real. Para alcanzar este objetivo, se plantean las siguientes metas:

1. Estudiar y adaptar una solución de conteo de personas mediante software libre de visión artificial basado en Python, librería de visión artificial OpenCV, redes neuronales convolucionales (CNN) pre-entrenadas para la detección de personas y tracking de objetos.
2. Diseñar y desarrollar un servidor con un backend, base de datos y frontend para exponer la información a los gestores de la biblioteca y a los alumnos.
3. Llevar a cabo pruebas en un entorno real para evaluar el sistema en términos de su rendimiento y su funcionamiento en un servidor con recursos limitados.

1.2 Herramientas software y elementos usados

Los diferentes entornos de desarrollo, programas y tecnologías que se han usado son las siguientes:

Cámara Tapo

La cámara que se utiliza para la transmisión de imágenes en tiempo real.

VLC

Se ha utilizado para comprobar la generación de streaming de video mediante RTSP.



Figura 3. Logo de VLC

Ubuntu 22.04 LTS

Se ha utilizado como sistema operativo en el servidor del proyecto.



Figura 4. Logo Ubuntu

ZeroTier

Permite crear y gestionar SDN (Software Defined Networks) para intercomunicar diferentes ordenadores en una red privada virtual (VPN) de manera fácil y rápida. [1]



Figura 5. Logo de ZeroTier

GitHub

Es una plataforma de desarrollo colaborativo que permite alojar, compartir y gestionar proyectos de software usando el sistema de control de versiones Git. [2]



Figura 6. Logo de GitHub

Visual Studio Code

Para el desarrollo e implementación del servidor, se ha utilizado Visual Studio Code como entorno de desarrollo integrado (IDE). VSC es un editor de código gratuito y multiplataforma y permite trabajar con diferentes lenguajes de programación, herramientas y frameworks. [3], [4]



Figura 7. Logo de Visual Studio Code

Lenguaje Python

Es el lenguaje que se usa para el desarrollo del proyecto, es un lenguaje de alto nivel. Se caracteriza por su sintaxis clara y simple, su facilidad de aprendizaje y su amplia biblioteca y de módulos externos. [5], [6]



Figura 8. Logo de Python

Librerías de Vision basada en OpenCV y módulos de tracking de objetos. En Python.
Redes neuronales entrenadas (YOLO, MobileNet) para detección de personas.

Docker

Software para la gestión, descarga y control de contenedores. [7] Se utiliza para contenedor de Grafana y MySQL.



Figura 9. Logo de Docker

Grafana

Se utiliza para la visualización y análisis de datos, esta herramienta permite crear y explorar paneles interactivos con diferentes tipos de gráficos, tablas, mapas, alertas, etc. Además, se conecta con diversas fuentes de datos, como bases de datos. [8], [9]



Figura 10. Logo de Grafana

MySQL

Es el sistema de gestión de bases de datos relacionales de código abierto que se utiliza, que permite almacenar, organizar y manipular datos en tablas y columnas.[10]



Figura 11. Logo de MySQL

NodeJs

Es un entorno de ejecución de JavaScript que permite ejecutar código JavaScript fuera del navegador.[11]

PM2

Es un gestor de procesos de aplicaciones en NodeJS que te permite mantenerlas vivas, recargarlas sin tiempo de inactividad y facilitar tareas comunes de administración del sistema. [12]



Figura 12. Logo de NodeJS y PM2

HTML

Es un lenguaje de marcado de hipertexto que define el significado y la estructura del contenido web. Usa etiquetas para crear elementos que representan el texto, las imágenes, los enlaces, los formularios y otros componentes de una página web. [13]



Figura 13. Logo de HTML

CSS

Es un lenguaje de hojas de estilo en cascada que se usa para dar estilo a los elementos HTML de una página web. [14]



Figura 14. Logo de CSS

Telegram

Es una aplicación de chat para Android y iOS, que te permite enviar y recibir mensajes de forma segura y rápida. La característica que interesa para este proyecto son los bots que pueden ser usados para comunicarse con el servidor y recibir notificaciones. [15]



Figura 15. Logo de Telegram

1.3 Metodologías de trabajo

Se ha seguido una metodología iterativa y flexible que permitió adaptar el enfoque a medida que se avanzaba y se obtenían resultados. Además, se hizo uso fundamental de la herramienta GitHub para gestionar y controlar las versiones del proyecto.

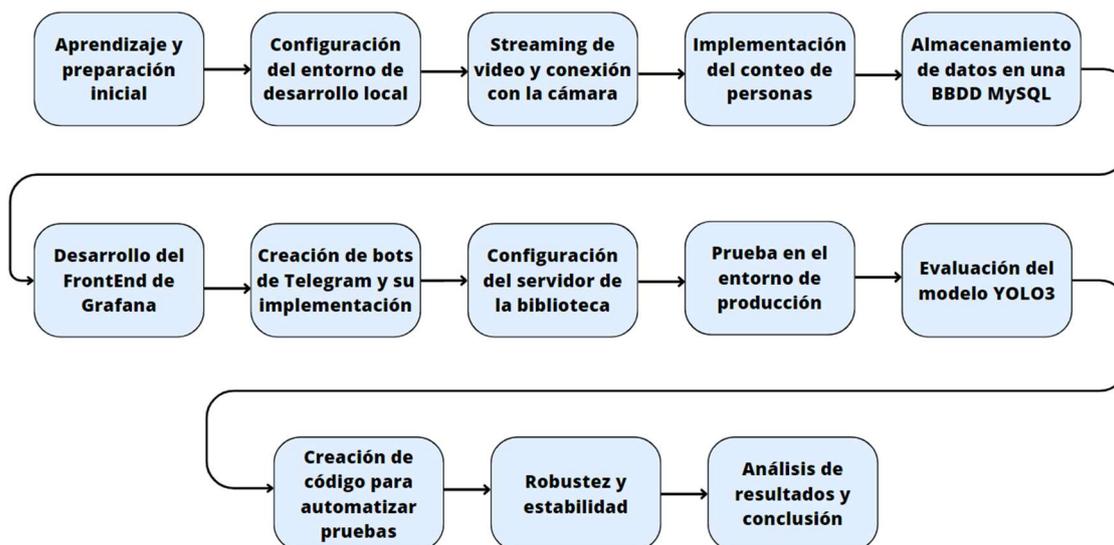


Figura 16. Metodología de trabajo

El objetivo principal era desarrollar un sistema de conteo de personas en una biblioteca, a continuación, se explicarán los pasos seguidos:

1. **Aprendizaje y preparación inicial:** Se comienza con el aprendizaje de Python, un lenguaje de programación esencial para el proyecto, y también se investiga sobre los diferentes modelos de detección y sistemas de conteo de personas.
2. **Configuración del entorno de desarrollo local:** Se crea un entorno de desarrollo local similar al entorno de producción en la biblioteca. Esto permite desarrollar y probar cada parte de manera aislada antes de implementarlo en el entorno real.
3. **Streaming de video y conexión con la cámara:** Se configura la cámara y se establece la transmisión de video mediante RTSP, se asegura así de que la comunicación entre la cámara y el programa de visualización es efectiva.
4. **Implementación del conteo de personas:** Se analiza y se adapta el programa de conteo de personas que se utiliza como punto de partida, en el cual el modelo utilizado es MobileNet SSD. Se conecta este programa a la transmisión de video.
5. **Almacenamiento de datos en una BBDD MySQL:** Se implementa la funcionalidad de almacenamiento de datos de ocupación, así como la fecha y hora en la que se registran, permitiendo de esta manera la gestión de la información recopilada.
6. **Desarrollo del frontend de Grafana:** Se trabaja en el frontend para mostrar gráficas de ocupación en Grafana.
7. **Creación de bots de Telegram y su implementación:** Se desarrollan dos bots encargados de proporcionar información relevante tanto a los alumnos como a los administradores. Estos bots se integran en el backend para facilitar el envío de información a través de ellos.
8. **Configuración del servidor de la biblioteca:** En el servidor de la biblioteca, se configura ZeroTier para el acceso remoto y PM2 para la gestión de programas, aunque se tuvo que recurrir a AnyDesk debido a dificultades con el acceso por SSH y ZeroTier.
9. **Prueba en el entorno de producción:** Se implementan todas las actualizaciones y cambios en el servidor de la biblioteca y se realizan pruebas en el entorno real para asegurarse del sistema funcionara de manera efectiva.

10. **Evaluación del modelo YOLO3:** Se evaluó la posibilidad de utilizar el modelo YOLO3 para la detección de personas, sin embargo, tras las pruebas con los videos de prueba, se determinó que no era eficiente en términos de procesamiento.
11. **Creación de código para automatizar pruebas:** Se desarrolló un código para automatizar las pruebas, lo que permitió ahorrar tiempo. Estas pruebas se llevaron a cabo utilizando videos del entorno real. Tras las diferentes pruebas se tomaba la decisión de mantener ángulo o modificarlo hasta encontrar la mejor opción.
12. **Robustez y estabilidad:** Se mantuvo el sistema en funcionamiento durante más de una semana, supervisando la gestión de programas a través de PM2 para garantizar su robustez y estabilidad.
13. **Análisis de resultados y conclusión:** Finalmente, se evalúan todos los resultados de las pruebas y se llega a una conclusión sobre la efectividad del sistema de conteo de personas.

1.4 Estructura y organización del proyecto

En este trabajo se ha desarrollado un sistema de conteo de personas basado en visión artificial. A continuación, se explica la estructura de esta memoria:

1. **Estado del arte de sistemas de visión para conteo de personas:** Se proporciona una breve contextualización histórica y se exploran las técnicas más avanzadas, desafíos y aplicaciones relacionados con los sistemas de visión para el conteo de personas.
2. **Cámara Tapo:** Se detallan las características de la cámara y se aborda su configuración.
3. **Sistema de detección y tracking:** Se presenta una explicación detallada de los modelos MobileNet SSD y YOLO3, así como del sistema de tracking. Se analiza el proceso de selección entre estos modelos y se describen las pruebas realizadas para tomar la decisión final.
4. **Diseño e implementación del servidor del sistema de monitorización de la ocupación:** Se aborda la estructura integral del sistema, incluyendo el servidor. Se explora el frontend utilizado para visualizar los datos, y se proporciona una explicación detallada sobre el servicio de monitorización implementado con la integración de un bot de Telegram.
5. **Conclusiones y trabajos futuros:** Se presentan las conclusiones derivadas de la finalización del proyecto y se evaluarán los objetivos alcanzados. También se discutirán las perspectivas de trabajos futuros en función de los logros obtenidos en el servicio desarrollado.

Capítulo 2. Estado del arte de sistemas de visión para conteo de personas

2.1 Introducción

La tecnología de visión por computadora ha experimentado un crecimiento exponencial, transformando la manera en que interactuamos con nuestro entorno y resolvemos desafíos complejos en diversas áreas. Uno de los campos impactados por estas innovaciones es el conteo de personas, una tarea muy relevante en múltiples contextos, que abarca desde el análisis de tráfico y la gestión de multitudes hasta la seguridad en espacios públicos y la investigación científica.

La capacidad de rastrear y contar personas en tiempo real ha abierto un abanico de oportunidades para optimizar operaciones, mejorar la experiencia del usuario y aumentar la seguridad. Desde tiendas minoritarias que buscan comprender los patrones de tráfico de clientes hasta eventos masivos que requiere una gestión fluida de multitudes, la precisión en el conteo de personas se ha convertido en una herramienta fundamental.

En este capítulo, se verán tanto los métodos tradicionales que sentaron las bases para el desarrollo de sistemas de conteo de personas como las técnicas de vanguardia que han revolucionado este campo en los últimos años. Además de explorar las herramientas y plataformas disponibles para los desarrolladores, también abordamos los desafíos persistentes que enfrenta la tecnología de conteo de personas, como la variabilidad en la apariencia y el movimiento humano, la oclusión en entornos densos y las preocupaciones de privacidad.

2.2 Fundamentos de sistemas de visión para conteo de personas

Para comprender en profundidad los sistemas de visión utilizados en el conteo de personas, es esencial familiarizarse con los fundamentos subyacentes de la visión por computadora y la detección de objetos. Estos conceptos forman la base de las técnicas más avanzadas y modernas utilizadas en los sistemas de conteo de personas. A continuación, se presentan algunos de los fundamentos clave [16]:

1. **Visión por computadora:**

La visión por computadora es un área interdisciplinaria que combina la informática y la percepción visual para permitir a las máquinas interpretar y comprender el contenido visual de imágenes y videos. Se basa en la extracción de características visuales, el procesamiento de imágenes y el análisis estadístico para lograr tareas como la detección de objetos y el seguimiento de movimiento [17] Para ello se utilizan diferentes algoritmos que permiten procesar, analizar y comprender las imágenes del mundo real.

2. **Detección de objetos:**

La detección de objetos implica identificar y localizar objetos específicos dentro de una imagen o vídeo. Se aborda mediante el análisis de características visuales, como bordes, texturas y patrones, y la correspondencia de estas características con modelos predefinidos.

3. **Segmentación y segmentación de instancias:**

La segmentación implica dividir una imagen en regiones semánticamente significativas, como objetos y fondos. La segmentación de instancias se refiere a la asignación de una etiqueta única a cada instancia individual de un objeto en una escena.

4. **Aprendizaje automático y aprendizaje profundo:**

El aprendizaje automático y el aprendizaje profundo son enfoques que permiten a las máquinas aprender patrones y representaciones a partir de datos.

Las redes neuronales convolucionales (**CNN**) y las redes neuronales recurrentes (**RNN**) son arquitecturas clave en el aprendizaje profundo utilizadas para tareas de visión por computadora.

5. **Supervisión y anotación de datos:**

La supervisión implica el entrenamiento de algoritmos utilizando conjuntos de datos etiquetados, donde las imágenes se anotan con etiquetas que indican la presencia y posición de objetos de interés.

La anotación de datos es un proceso laborioso pero esencial para enseñar a los algoritmos cómo identificar personas en diferentes contextos.

6. **Clasificación y detección:**

La clasificación se refiere a asignar una etiqueta a una imagen completa para identificar qué objeto o clase está presente.

La detección es una extensión de la clasificación que implica localizar la posición exacta de objetos en una imagen.

Al comprender estos fundamentos, es posible apreciar cómo los sistemas de visión para el conteo de personas utilizan estas técnicas y conceptos para identificar y rastrear individuos en entornos diversos.

2.3 Métodos tradicionales de conteo de personas

Históricamente, el conteo de personas se ha elaborado mediante enfoques tradicionales que se basan en técnicas de procesamiento de imágenes y análisis de video. Estos métodos, aunque fundamentales para el desarrollo de sistemas de conteo de personas, a menudo enfrentan limitaciones en términos de precisión y adaptabilidad a entornos diversos. A continuación, se presentan algunos de los métodos tradicionales más utilizados:

1. **Detección de bordes y umbralización [18]:**

Una de las aproximaciones más simples involucra la detección de bordes en las imágenes y la aplicación de umbrales para identificar regiones que podrían corresponder a personas. Aunque este método puede funcionar en escenarios controlados, tiende a ser sensible al ruido y a la variación en la iluminación.

2. **Segmentación de fondos [19]:**

La segmentación de fondos implica la separación de objetos en movimiento del fondo estático en una secuencia de video. Este método puede utilizarse para detectar personas como objetos en movimiento en relación con su entorno. Sin embargo, sufre de problemas en escenarios con cambios bruscos en la iluminación y la presencia de objetos en movimiento que no son personas.

3. **Seguimiento de objetos:**

Los métodos de seguimiento de objetos rastrean el movimiento de un objeto a lo largo del tiempo en un video. Puede usarse para contar personas al rastrear sus trayectorias, pero puede ser desafiante en situaciones con interacciones cercanas y oclusiones.

4. **Sensores de proximidad:**

En entornos controlados, se han utilizado sensores de proximidad (como sensores infrarrojos) para detectar la presencia de personas en una ubicación específica. Estos sensores pueden ser útiles para contar personas en lugares como puertas de entrada, pero no proporcionan información sobre la ubicación exacta.

5. **Análisis de flujo óptico:**

El análisis de flujo óptico implica el seguimiento del movimiento en función de los cambios en la intensidad de píxeles entre frames consecutivos. Puede ayudar a identificar áreas de movimiento en una escena, que podrían corresponder a personas en movimiento.

A pesar de sus contribuciones iniciales, estos métodos tradicionales enfrentan desafíos en términos de precisión en entornos complejos y cambiantes. La evolución de la tecnología de visión por computadora ha impulsado el desarrollo de enfoques más sofisticados y precisos basados en el aprendizaje profundo y las redes neuronales convolucionales, lo que ha llevado a una nueva era de sistemas de conteo de personas altamente efectivos.

2.4 Técnicas modernas de conteo de personas

Con el avance de las tecnologías de aprendizaje profundo y la disponibilidad de conjuntos de datos masivos, las técnicas modernas de conteo de personas han experimentado una revolución significativa en términos de precisión y adaptabilidad. Estas técnicas han superado muchas de las limitaciones de los métodos tradicionales y han permitido la detección y el seguimiento de personas en escenarios complejos y cambiantes. A continuación, se presentan algunas de las técnicas modernas más influyentes [20], [21]:

1. **Redes neuronales convolucionales (CNN):**

Las **CNN** han transformado la visión por computadora al permitir la detección y clasificación precisa de objetos en imágenes y videos. Se utilizan arquitecturas específicas, como Faster R-CNN y YOLO (You Only Look Once), para identificar personas en tiempo real y asignarles coordenadas precisas.

2. **Detección y seguimiento de múltiples objetos:**

Las técnicas modernas no solo detectan personas, sino que también pueden seguir su movimiento a lo largo del tiempo. Los algoritmos de seguimiento utilizan predicciones basadas en el estado anterior de los objetos y las relaciones de proximidad para mantener el seguimiento preciso incluso en situaciones de oclusión.

3. **Detección de personas en multitudes:**

Uno de los desafíos más significativos es la detección de personas en multitudes densas. Las técnicas modernas se basan en el aprendizaje profundo para diferenciar personas y ruido de fondo en situaciones caóticas.

4. **Segmentación semántica y de instancias:**

La segmentación semántica y de instancias permite no solo identificar personas, sino también segmentar cada instancia individual. Esto es útil para contar personas de manera más precisa en situaciones con interacciones cercanas y oclusiones.

5. **Uso de cámaras 3D y sensores inteligentes:**

La integración de cámaras 3D y sensores inteligentes ha mejorado la percepción espacial y la comprensión de la tridimensionalidad. Esto es especialmente valioso para el conteo preciso en entornos complejos como interiores de edificios o áreas exteriores con obstáculos.

6. **Transferencia de aprendizaje (*Transfer learning*) y modelos preentrenados:**

La transferencia de aprendizaje permite aprovechar modelos preentrenados en grandes conjuntos de datos para tareas específicas, incluido el conteo de personas. Esto reduce el tiempo de entrenamiento y mejora la precisión en dominios con datos limitados.

7. Enfoques basados en flujo óptico mejorado:

Las técnicas modernas de conteo también han mejorado el análisis de flujo óptico para lograr seguimiento de personas más robusto y preciso.

Estas técnicas modernas han transformado el conteo de personas en una ciencia más precisa y versátil. Al aprovechar el poder del aprendizaje profundo y la inteligencia artificial, estos enfoques han abierto la puerta a aplicaciones más amplias y sofisticadas, desde la gestión de multitudes en eventos hasta la optimización del flujo de personas en espacios públicos.

2.5 Desafíos y problemáticas

Pese a los notables avances en los sistemas de conteo de personas, persisten desafíos y problemáticas que requieren atención y soluciones innovadoras. Estos desafíos abarcan desde aspectos técnicos hasta consideraciones éticas y prácticas en el despliegue de estos sistemas. A continuación, se presentan algunos de los desafíos más destacados en el campo del conteo de personas:

1. Variabilidad en la apariencia humana:

Las diferencias en la edad, género, vestimenta y postura de las personas pueden dificultar la detección precisa. Los sistemas deben adaptarse a la variabilidad y generalizar a diferentes tipos de personas.

2. Cambios en la iluminación y condiciones ambientales:

Las fluctuaciones en la iluminación y las condiciones climáticas pueden afectar la calidad de las imágenes y la precisión de los algoritmos de detección. La adaptación a condiciones cambiantes es un reto crítico en aplicaciones al aire libre y en espacios con iluminación variable.

3. Oclusiones e interacciones cercanas:

En entornos con multitudes densas, las personas a menudo se ocultan entre sí, lo que dificulta la detección y el seguimiento precisos. Los métodos modernos han abordado este desafío mediante técnicas de seguimiento robustas y la segmentación de instancias.

4. Privacidad y ética:

El despliegue de sistemas de conteo de personas plantea cuestiones de privacidad y ética. La recopilación y el procesamiento de datos de personas deben cumplir con regulaciones de privacidad y considerar la retención y el uso responsable de la información.

5. Escalabilidad y eficiencia computacional:

A medida que aumenta la escala de las aplicaciones, se requiere una mayor capacidad de procesamiento para realizar conteos precisos en tiempo real. La eficiencia computacional es esencial para sistemas que operan en tiempo real y en grandes áreas.

6. Adaptabilidad a diferentes escenarios:

Los sistemas de conteo de personas deben ser adaptables a una variedad de escenarios, desde espacios cerrados como tiendas minoristas hasta áreas públicas al aire libre. La falta de adaptabilidad puede limitar la utilidad y la eficacia de estos sistemas en contextos diversos.

7. Precisión en escenarios difíciles:

A pesar de los avances, todavía existen desafíos en escenarios complejos como condiciones climáticas extremas, iluminación deficiente o multitudes muy densas. Mejorar la precisión en estos contextos sigue siendo un objetivo clave.

8. **Evaluación de rendimiento:**

Medir y evaluar la precisión y el rendimiento de los sistemas de conteo de personas puede ser complicado debido a la falta de conjuntos de datos estandarizados y métricas uniformes. El desarrollo de métricas fiables y conjuntos de datos representativos es esencial para una evaluación precisa.

La superación de estos desafíos no solo requiere innovación técnica, sino también consideraciones éticas y prácticas sólidas. A medida que los sistemas de conteo de personas siguen evolucionando, es esencial abordar estos desafíos para lograr soluciones confiables y eficaces en varias aplicaciones.

2.6 Aplicaciones prácticas

La tecnología de conteo de personas tiene un impacto significativo en una amplia gama de campos y aplicaciones, ofreciendo soluciones innovadoras para desafíos complejos. A continuación, se explicarán algunas de las aplicaciones prácticas en las que los sistemas de conteo de personas están demostrando su utilidad y versatilidad:

1. **Gestión de multitudes en eventos y lugares públicos:**

Los sistemas de conteo de personas son esenciales para eventos masivos, como conciertos, festivales y manifestaciones, donde la gestión de multitudes de manera eficiente es crucial para la seguridad y la logística.

2. **Análisis de tráfico y transporte:**

En entornos urbanos y de transporte, el conteo preciso de personas en estaciones de autobuses, trenes y calles peatonales es fundamental para la planificación del transporte público y la optimización del flujo de tráfico.

3. **Retail y comercio electrónico:**

En tiendas minoristas y centros comerciales, el conteo de personas ayuda a comprender los patrones de tráfico de clientes y a optimizar la disposición de los productos. En comercio electrónico, la medición del tráfico en sitios web es valiosa para la toma de decisiones de marketing y diseño de experiencia de usuario.

4. **Seguridad y vigilancia:**

Los sistemas de conteo de personas mejoran la vigilancia al detectar el comportamiento sospechoso o la entrada no autorizada en áreas restringidas. En espacios públicos como aeropuertos y estaciones de tren, el conteo de personas contribuye a la seguridad y la identificación de posibles amenazas.

5. **Análisis de espacios públicos:**

En parques, plazas y áreas urbanas, el conteo de personas proporciona información valiosa sobre la afluencia de visitantes y permite una mejor planificación de eventos y recursos.

6. **Monitoreo de salud pública:**

En situaciones de emergencia, como brotes de enfermedades o desastres naturales, los sistemas de conteo de personas pueden ayudar a estimar la propagación y el impacto, facilitando la respuesta y mitigación. Por ejemplo, después del COVID se han incorporado muchos sistemas de conteo para el control de aforo.

7. **Optimización de espacios y recursos:**

El conteo de personas permite optimizar el uso de espacios y recursos en entornos como bibliotecas, museos, gimnasios y centros de conferencias. Teniendo así control sobre el número de personas que entran y salen.

Estas aplicaciones ilustran la amplitud de beneficios que los sistemas de conteo de personas pueden aportar en diversos ámbitos. Desde mejorar la seguridad y la eficiencia hasta facilitar la toma de decisiones informadas, estas tecnologías están transformando la forma en que interactuamos con nuestros entornos y abordamos desafíos complejos.

2.7 Herramientas y plataformas disponibles

El desarrollo de sistemas de conteo de personas se ha visto facilitado por una variedad de herramientas, bibliotecas y plataformas de software que ofrecen soluciones predefinidas y recursos para investigadores y desarrolladores. Estas herramientas aceleran la implementación y experimentación, permitiendo una rápida iteración y ajuste de los algoritmos. A continuación, se presentan algunas de las herramientas y plataformas disponibles en este campo:

1. **OpenCV:**
OpenCV es una biblioteca de código abierto ampliamente utilizada para visión por computadora y procesamiento de imágenes. Ofrece funciones para detección de objetos, seguimiento de objetos, análisis de movimiento y más, lo que es útil para implementar sistemas de conteo de personas.
2. **TensorFlow y PyTorch:**
TensorFlow y PyTorch son dos de las bibliotecas más populares para el desarrollo de modelos de aprendizaje profundo. Proporcionan implementaciones de arquitecturas de redes neuronales convolucionales y recurrentes, lo que permite a los desarrolladores crear sistemas de conteo de personas basados en aprendizaje profundo.
3. **YOLO (You Only Look Once):**
YOLO es un enfoque de detección de objetos en tiempo real adoptado para detectar personas. Ofrece alta velocidad y precisión en la detección de objetos, lo que lo hace adecuado para aplicaciones en tiempo real como el conteo de personas en videos de tráfico.
4. **MobilNet SSD:**
MobileNet SSD es una arquitectura de red neuronal especialmente diseñada para aplicaciones en dispositivos móviles y con recursos computacionales limitados. Ofrece detección de objetos en tiempo real, incluidas personas, con una eficiencia adecuada para sistemas embebidos y aplicaciones en tiempo real. Como se diseñó para aplicaciones en dispositivos móviles es más ligera y no necesita muchos recursos para funcionar correctamente.
5. **Mask R-CNN:**
Mask R-CNN es una extensión de Faster R-CNN que también permite la segmentación de instancias en imágenes. Es útil para contar personas individualmente y rastrear sus movimientos en escenarios complejos.
6. **PaddlePaddle:**
PaddlePaddle es una plataforma de aprendizaje profundo de código abierto desarrollada por Baidu. Proporciona herramientas y recursos específicos para aplicaciones de visión por computadora y detección de objetos, incluido el conteo de personas.
7. **Herramientas de Anotación de Datos:**
Plataformas como **Labelbox**, **VGG Image Annotator (VIA)** y **RectLabel** permiten la anotación eficiente de datos para entrenar modelos de conteo de personas.

8. Plataformas Comerciales Especializadas:

Algunas empresas ofrecen soluciones comerciales especializadas en conteo de personas con hardware y software integrados para varias aplicaciones.

Estas herramientas y plataformas brindan a los investigadores y desarrolladores una base sólida para crear sistemas de conteo de personas efectivos y precisos. Además, facilitan la experimentación y la colaboración en la comunidad de visión por computadora, contribuyendo al avance continuo en este emocionante campo.

Capítulo 3. Cámara Tapo de TP-Link

3.1 Características

La cámara **TC70** [22] es una cámara de seguridad Wi-Fi inteligente de bajo coste que ofrece varias características que son interesantes:

- **Visión nocturna.** Incluso de noche alcanza una distancia visual hasta 9 metros.
- **Detección de movimiento.**
- **Visión de alta definición.** Captura todos los detalles con una resolución hasta de 1080p, esto permite capturar detalles más finos y precisos y esto ayudara al programa a detectar las personas con mayor precisión.
- **Rotación de 360°** . Esto facilita el poner la cámara en el ángulo que mejor muestre el área que se quiere enfocar.

En especial estas dos últimas características hacen que la cámara TC70 sea una buena opción para el sistema de visión para conteo de personas. Además, que se puede conectar fácilmente a una red Wi-Fi y se puede controlar desde una aplicación móvil, lo que facilita en sí instalación y configuración.



Figura 17. Cámara Tapo TC70

3.2 Instalación y configuración de la cámara Tapo

La cámara se ha colocado en el techo en una posición donde se enfoca la entrada de la biblioteca, Figura 2.

Para configurar la cámara con la aplicación se han seguido los siguientes pasos:

- Se crea una cuenta de cámara en la aplicación Tapo.
- Se hace click en el icono + para añadir la cámara y se elige la cámara que se quiera añadir.

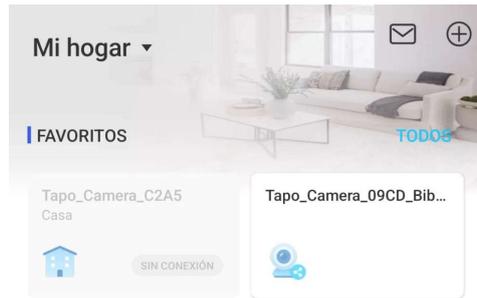


Figura 18. App Tapo. Añadir cámara



Figura 19. App Tapo. Elección de cámara

- Al hacer la selección, se enciende la cámara y se espera 30 segundos hasta que el LED parpadee en rojo y verde, en ese momento se pasa al siguiente paso y se conecta a la red de la cámara, Tapo_Cam_XXXX.



Figura 20. App Tapo. Comprobación del LED de la cámara



Conectar al equipo Tapo

Ir a la configuración de Wi-Fi del teléfono tablet y conectar a la red del dispositivo Tapo: Tapo_Cam_XXXX. Luego volver a esta aplicación.

XXXX son los últimos cuatro dígitos de la dirección MAC única del dispositivo Tapo. Puede encontrarlo en su dispositivo Tapo.



Figura 21. App Tapo. Conectarse a la red Tapo

- Una vez la app ha encontrado la cámara, se realiza la conexión a la red Wi-Fi
- Finalmente se emparejará la cámara, se le asignará un nombre y una ubicación.

La cámara Tapo admite el protocolo RTSP [23], por lo que en caso de que el usuario quiera visualizar la transmisión en tiempo real de la cámara utilizando una aplicación de terceros, como el reproductor VLC, solo necesitará introducir la URL correspondiente a la transmisión RTSP dentro de la aplicación VLC Player. Obteniendo dicha URL, se podrá conectar el programa de conteo a las imágenes de la cámara en tiempo real. Para ello se tiene que configurar la cámara desde la aplicación, siguiendo estos pasos:

- Se selecciona la cámara que queremos configurar, para ir a la página 'Vista en vivo'
- Al estar en la página vista en vivo, hay que pinchar en ajustes, en la parte superior derecha, esto lleva a la página de configuración

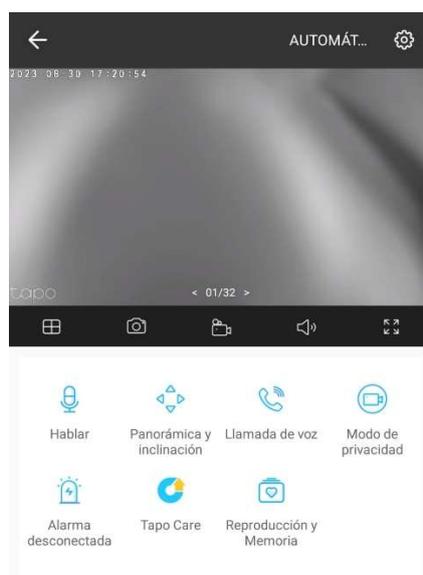


Figura 22. App Tapo. Vista en vivo

- Para saber la dirección IP de la cámara, hay que acceder a la primera opción que aparece al acceder a la página de configuración.



Figura 23. App Tapo. Ajustes de la cámara 1



Figura 24. App Tapo. Dirección IP de cámara

- Para crear una cuenta de la cámara hay que acceder a 'Configuración Avanzada>Cuenta del dispositivo', y ahí se crea la cuenta, introduciendo el usuario y contraseña que se desee. Así ya se tendría la cuenta de cámara.



Figura 25. App Tapo. Ajustes de la cámara 2

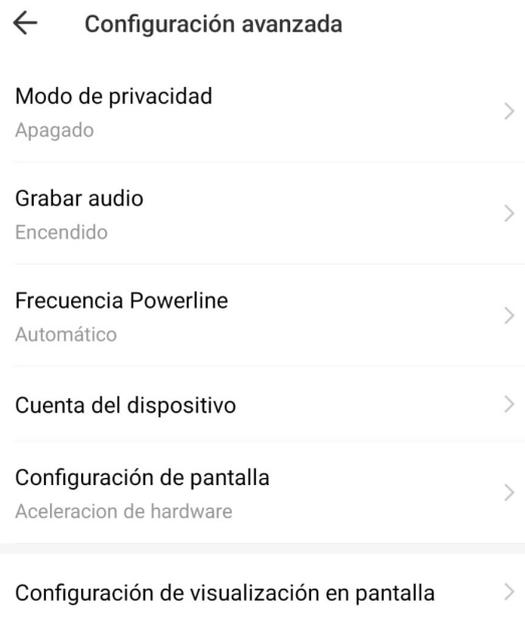
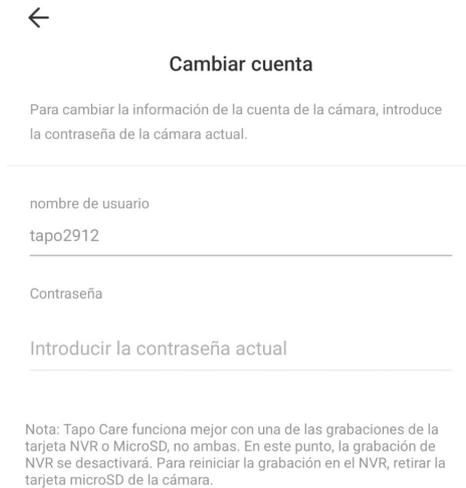


Figura 26. App Tapo. Configuración Avanzada



←

Cambiar cuenta

Para cambiar la información de la cuenta de la cámara, introduce la contraseña de la cámara actual.

nombre de usuario
tapo2912

Contraseña

Introducir la contraseña actual

Nota: Tapo Care funciona mejor con una de las grabaciones de la tarjeta NVR o MicroSD, no ambas. En este punto, la grabación de NVR se desactivará. Para reiniciar la grabación en el NVR, retirar la tarjeta microSD de la cámara.

Figura 27. App Tapo. Creación de cuenta para la cámara

La URL debe seguir esta estructura:

`rtsp://user:password@IPAddress:554/h264/ch1/main/av_stream`

Donde user y password, son el usuario y contraseña con los que se ha creado la cuenta de la cámara, IPAddress es la IP de la cámara, eso se puede saber accediendo a la aplicación y 554 es el puerto que utiliza RTSP.

Configuración final de la cámara:

- **Resolución:** 720p
- **Modo de imagen:** Color

3.3 Pruebas de medición de fps y latencia del streaming

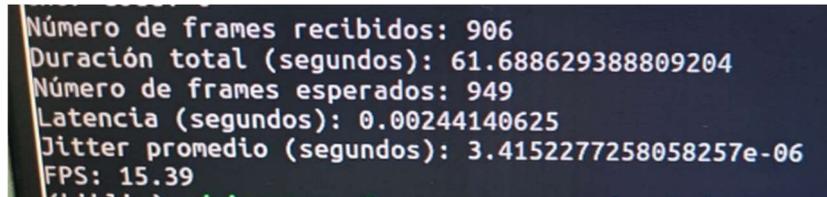
El programa de conteo puede operar tanto con un video mp4 pregrabado como con un streaming de video en tiempo real. En el caso de un video mp4 pregrabado, se pueden obtener condiciones ideales para la recepción de imágenes. Sin embargo, es importante destacar que el funcionamiento del programa de conteo, incluyendo sus módulos de detección de personas y de seguimiento, podría verse afectado si la cantidad de frames por segundo no es adecuada, lo que podría aumentar la latencia y el retardo en la llegada de frames.

La cámara Tapo instalada en la puerta de entrada de la Biblioteca de Antigones usa la red Wifi IoTUT gestionada por el Servicio de Informática de la UPCT. La red está dentro de una VLAN privada securizada por el Servicio de Informática, por lo que el servidor debe estar en la misma red Wifi IoTUT. En ese espacio hay muchas redes Wifi operando, puede haber interferencias y esto puede afectar al rendimiento de la cámara y a la recepción de suficientes frames por segundo en el servidor.

Por lo tanto, las pruebas de medición se llevaron a cabo en condiciones de funcionamiento real de la cámara y se centraron exclusivamente en la cuantificación de imágenes recibidas y en la evaluación del retardo. Es importante destacar que estas pruebas no abordaron el funcionamiento del servidor ni del módulo de conteo en sí.

Durante las pruebas realizadas, se observaron los siguientes resultados con la configuración de la cámara mencionada (streaming RSTP, color, resolución 720p):

- Velocidad de frame por segundo (**FPS**): 15,39 FPS
- **Latencia**: 0.0024 egundos.
- **Jitter**: 3.42×10^{-6}



```
Número de frames recibidos: 906
Duración total (segundos): 61.688629388809204
Número de frames esperados: 949
Latencia (segundos): 0.00244140625
Jitter promedio (segundos): 3.4152277258058257e-06
FPS: 15.39
```

Figura 28. Captura pruebas con modelo MobileNet SSD

Los valores de latencia y jitter obtenidos, indican un rendimiento bueno. Una latencia baja significa que la respuesta entre la cámara y el sistema es prácticamente instantánea, lo que es altamente deseable en aplicaciones de monitoreo y seguimiento. El jitter extremadamente bajo también sugiere una estabilidad en la transmisión de datos, lo que garantiza una captura de video constante. En conjunto, estos resultados garantizan un número medio de fps adecuados para el correcto funcionamiento del módulo de conteo.

Capítulo 4. Sistema de visión para conteo de personas.

El conteo de personas mediante visión artificial que pasan por una zona requiere de un sistema de visión formado por los siguientes elementos básicos:

1. Cámara de visión que adquiere imágenes en tiempo real y las transmite a un equipo de procesamiento de imágenes.
2. Equipo de procesamiento de imágenes, por ejemplo, un PC con suficiente capacidad de procesamiento, que implementa en software conceptualmente los siguientes módulos:
 - **Módulo de recepción de streaming de imágenes.** Alternativamente puede ser módulo reproductor de un video mp4.
 - **Módulo de detección de personas.** Cada frame de la imagen se procesa mediante un programa de detección de objetos; el programa de detección identifica y clasifica una clase de objeto con cierta probabilidad. En la detección de personas, la clase persona se entrena y especializa. Cuando se detecta una persona se etiqueta con un ID y en sucesivos frames se busca si se sigue detectando a la misma persona.
 - **Módulo de tracking,** en sucesivos frames se aplica un algoritmo de tracking de un ID de objeto detectado.
 - o **Procesamiento de una línea *boundary*:** en la imagen se fija una línea *boundary* divisoria de lo que es entrada o salida del área. El módulo de tracking genera un evento si el ID de objeto atraviesa dicha línea divisoria configurable.

La detección de objetos en visión artificial es una tecnología informática relacionada con el procesamiento de imágenes y la visión por computadora. Su objetivo es identificar y predecir objetos, así como localizar su área dentro de una imagen o video. Para lograrlo, se basa en la clasificación de imágenes, que puede realizarse con redes neuronales o clasificadores primitivos. Hoy en día, existen soluciones de detección de objetos, y en particular de personas, pre-entrenadas y basadas en redes neuronales de aprendizaje profundo. Un primer enfoque ha sido estudiar y entender las arquitecturas de aprendizaje profundo disponibles en programas de software libre e intentar adaptarlo y evaluarlo para el propósito del proyecto.

Construyendo sobre esta base, la detección de objetos nos permite localizar todos los objetos posibles en un marco determinado. Es decir, podemos detectar instancias de objetos semánticos de ciertas clases, como humanos, edificios o automóviles, en una sola imagen o video. Algunos de los dominios más investigados en esta área incluyen la detección de rostros y la detección de peatones.

4.1 COCO base de datos de imágenes para entrenamiento

Common Objects in Context (COCO) es un conjunto de datos fundamental en el campo de la visión por computadora y el aprendizaje profundo. COCO se destaca por su diversidad y riqueza, ya que contiene una amplia variedad de imágenes etiquetadas que abarcan 80 categorías diferentes de objetos. Este conjunto de datos se ha convertido en un recurso esencial para el entrenamiento y evaluación de modelos de detección y segmentación de objetos en imágenes.

[24]

Las características clave de YOLO incluyen:

- **Detección en una sola pasada:** YOLO divide la imagen en una cuadrícula y predice las cajas delimitadoras y las probabilidades de clase para cada celda de la cuadrícula en una sola pasada, lo que lo hace extremadamente rápido.
- **Precisión:** A pesar de su velocidad, YOLO mantiene una precisión competitiva en comparación con los enfoques más lentos y complejos.
- **Flexibilidad:** YOLO es altamente configurable y puede adaptarse a diferentes tamaños de cajas, clases de objetos y resoluciones de imagen.
- **Uso general:** Además de la detección de objetos, YOLO se ha aplicado con éxito en tareas como la detección de personas, vehículos, animales, entre otros

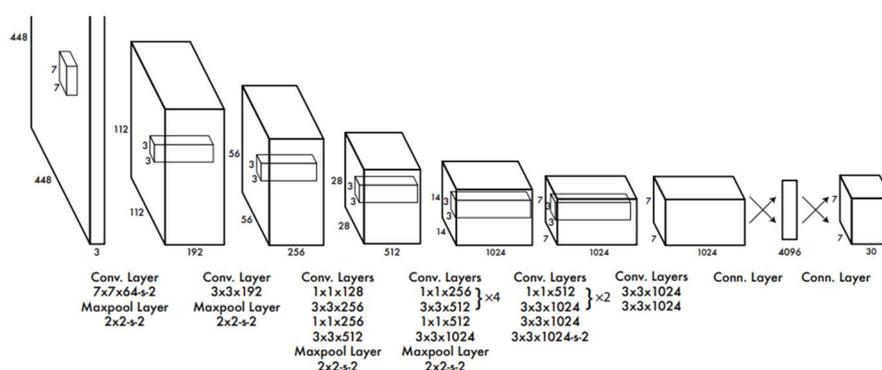


Figura 30. Arquitectura YOLO. Fuente [26]

YOLO3

YOLO3 es la tercera iteración de la familia YOLO y trae varias mejoras con respecto a las versiones anteriores. Estas mejoras incluyen [27], [28]:

- **Mejor precisión:** YOLO3 presenta una mayor precisión en la detección de objetos, lo que lo hace más confiable para aplicaciones críticas.
- **Detección de objetos múltiples:** YOLO3 puede detectar un mayor número de objetos en una sola imagen, incluso cuando se superponen.
- **Detección de objetos pequeños:** Se ha mejorado la capacidad de YOLO3 para detectar objetos pequeños, lo que lo hace más adecuado para una variedad de aplicaciones.
- **Soporte para diferentes arquitecturas:** YOLO3 se ha adaptado para funcionar en diferentes arquitecturas de red, lo que permite su implementación en una variedad de plataformas y dispositivos.

4.3 Modelo MobileNet SSD de detección de objetos

MobileNet SSD (MobileNet Single Shot MultiBox Detector) es una red neuronal convolucional que se utiliza para la detección de objetos en tiempo real. Es una versión optimizada de la arquitectura MobileNet que utiliza una técnica llamada **SSD (Single Shot Detector)** para detectar objetos en una imagen.

MobileNet

MobileNet es una arquitectura de red neuronal convolucional (CNN) desarrollada por Google Research. Fue diseñada específicamente para ejecutarse de manera eficiente en dispositivos con recursos limitados, como teléfonos móviles y sistemas integrados. La motivación detrás de

MobileNet era lograr una red neuronal lo suficiente mente ligera y rápida para tareas de visión por computadora en dispositivos móviles sin sacrificar la precisión.[29]

MobileNet utiliza una serie de técnicas y conceptos como:

- **Convoluciones profundas separables:** En lugar de aplicar convoluciones estándar en todos los canales de entrada, MobileNet utiliza convoluciones profundas separables, que dividen las convoluciones en dos etapas: una convolución en cada canal y una convolución punto a punto. Esto reduce drásticamente el costo computacional. [30]
- **Factor de ancho y resolución:** MobileNet permite ajustar el factor de ancho y la resolución de la red equilibrar la velocidad y la precisión, lo que permite una personalización según las necesidades del dispositivo y la aplicación.

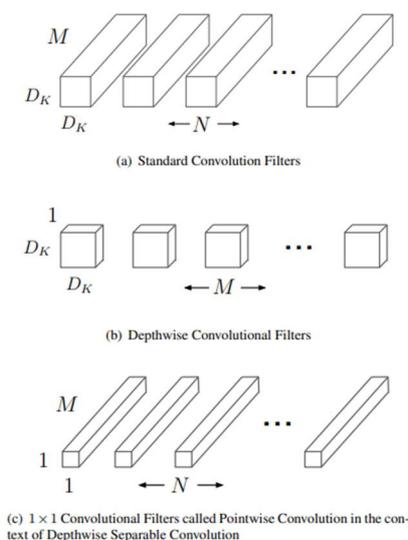


Figura 31. Filtros convolucionales estándar de (a) se sustituye por dos capas: convolución en profundidad (b) y la convolución puntual en (c) para construir un separable en profundidad. Fuente [29]

SSD

SSD es un algoritmo de detección de objetos que se destaca por su capacidad para detectar múltiples objetos en una sola pasada de una red convolucional. **Wei Liu** lo propuso por primera vez en 2016. A diferencia de los enfoques anteriores que requerían múltiples pasadas de la red para detectar objetos a diferentes escalas y relaciones de aspecto, SSD realiza esta tarea en un solo paso.[31]

SSD logra esto al incluir múltiples cajas delimitadoras (bounding boxes) y clasificadores en diferentes capas de la red, lo que permite detectar objetos de diferentes tamaños y formas en paralelo. Luego, utiliza un algoritmo de selección para filtrar las detecciones redundantes de baja confianza. [32]

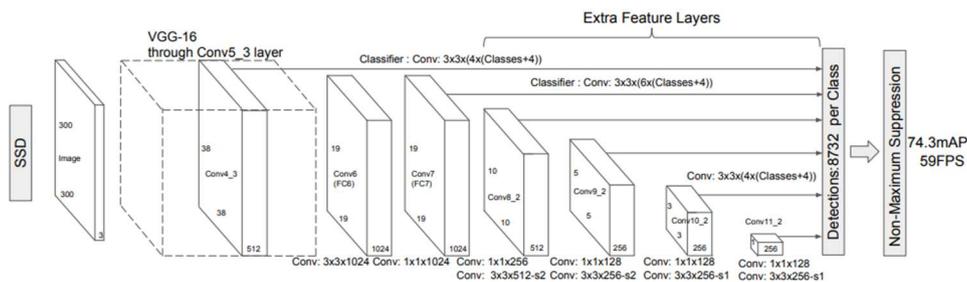


Figura 32. Algoritmo SSD, funcionamiento mediante Redes Convolucionales, Fuente [31]

MobileNet SSD

MobileNet SSD combina la eficiencia de MobileNet con la capacidad de detección de objetos de SSD. El resultado es un modelo de detección de objetos que es rápido y preciso, adecuado para su implementación en dispositivos que tengan pocos recursos. Se ha convertido en una opción popular para aplicaciones de detección de objetos en tiempo real, como sistemas de seguridad, vehículos autónomos, seguimiento de objetos y mucho más.

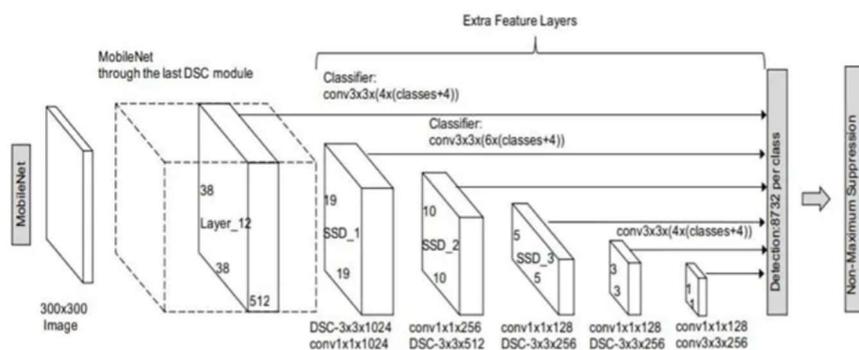


Figura 33. Algoritmo MobileNet SSD, Fuente [33]

4.4 Sistema de tracking de objetos en una imagen

El seguimiento de centroides (tracking) es una técnica que se utiliza para rastrear objetos en secuencias de video. Su funcionamiento se basa en la identificación y seguimiento de los centroides de los objetos detectados en cada frame del video. El centroide es simplemente el punto medio de un objeto, que se utiliza como una representación efectiva de su posición.

La biblioteca de visión por computadora de código abierto, **OpenCV**, desempeña un papel esencial en la implementación eficiente y precisa de sistemas de seguimiento de centroides. Para lograrlo, se emplean dos módulos clave [34]:

- El módulo **'centroidtracker.py'** es una implementación de un algoritmo de seguimiento de centroides que hace uso óptimo de las funciones de OpenCV. Este módulo permite el seguimiento de múltiples objetos en tiempo real, lo que lo hace especialmente útil en aplicaciones de seguimiento de personas. Contiene la lógica principal para el seguimiento, utiliza métodos de detección de objetos en cada frame del video y asignarles un identificador único. Además, realiza un seguimiento de la trayectoria de cada objeto a lo largo del tiempo.

- El módulo **'trackableobject.py'** es responsable de mantener un registro de los objetos detectados, incluyendo su ID, posición y la cantidad de frames consecutivos en los que han sido rastreados.

El proceso de seguimiento de centroides se desarrolla en los siguientes pasos:

1. **Detección de objetos:** En la primera etapa, se utiliza un algoritmo de detección de objetos, como YOLO o MobileNet SSD, para identificar y localizar objetos en cada frame del video. Este algoritmo analiza el contenido visual y en un tensor numpy 'detections', que es una colección de datos numéricos en varias dimensiones, se almacenan las coordenadas (posición) de los objetos detectados, así como sus etiquetas (clases de objetos). La detección inicial es fundamental para iniciar el seguimiento de los objetos.
2. **Identificación de personas:** Tras la detección de los objetos, se seleccionan las detecciones que superan el umbral mínimo de confianza. En el caso de este proyecto si se supera el umbral mínimo de confianza, se extrae la etiqueta almacenada en el tensor 'detections' y si coincide con 'persona' el seguimiento de centroides comenzará.

```
35 CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",  
36           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",  
37           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",  
38           "sofa", "train", "tvmonitor"]  
39
```

Figura 34. Posibles etiquetas de las clases

3. **Cálculo de centroides:** Una vez que los objetos han sido detectados y se confirma que son personas, se procede al cálculo de los centroides. El centroide de un objeto es simplemente el punto medio de dicho objeto en el plano XY. Se calcula a partir de las coordenadas proporcionadas por el algoritmo de detección. Este punto se utiliza como una representación efectiva de la posición del objeto y es esencial para el seguimiento.
4. **Asignación de IDs y creación de rastreadores:** En esta etapa, el módulo 'centroidtracker.py' entra en acción. Su función principal es asignar un identificador único (ID) a cada objeto detectado, con objeto se refiere a persona. Este proceso se basa en las coordenadas de los centroides en el frame actual y en los frames anteriores. Si un objeto se ha movido de un frame a otro, el módulo intenta asignarle el mismo ID, lo que permite realizar un seguimiento continuo de la trayectoria del objeto. Esta asignación se basa en criterios de proximidad y otras estrategias para garantizar una identificación precisa. Además, se crea una lista de rastreadores para objetos 'persona' que se utilizarán en frames posteriores
5. **Seguimiento de trayectorias:** Con cada frame subsiguiente, se recorren todos los rastreadores de objetos previamente creados y el sistema actualiza la posición de los centroides y mantiene un registro de las trayectorias de los objetos a lo largo del tiempo. Esto implica mantener un registro de las coordenadas de los centroides en cada frame y vincularlas al objeto correspondiente mediante su ID. Esta información acumulativa se convierte en una representación histórica de la trayectoria del objeto. La información resultante se almacena en un diccionario 'objects' para su posterior uso en el conteo de personas.
6. **Uso de 'trackableobject.py':** El módulo 'trackableobject.py' es esencial en este proceso. Después de que 'centroidtracker.py' haya asignado un ID único a cada objeto detectado, se crea una instancia de 'trackableobject.py' para ese objeto en particular. Esta instancia

se utiliza para mantener un registro de información importante sobre el objeto, incluyendo:

- **ID del objeto:** El ID único asignado al objeto.
 - **Posición del Objeto:** La posición actual del objeto, que se actualiza a medida que el objeto se mueve en frames sucesivos.
 - **Historial de Trayectoria:** Se mantiene un registro de las coordenadas de los centroides del objeto en frames anteriores. Esto permite el cálculo de trayectorias históricas del objeto.
7. **Actualización de información de objetos:** Posteriormente, se entra en un bucle que itera el diccionario 'objects', que contiene información sobre los objetos detectados en el frame actual. En cada iteración, se obtiene el identificador y el centroide de cada objeto detectado. Se busca en el diccionario 'trackableObject' para comprobar si existe un objeto rastreado correspondiente al identificador mencionado anteriormente. Si no existe, se crea un nuevo objeto rastreado de la clase 'TrackableObject', que contiene información sobre el objeto detectado y su trayectoria. Si ya existe el objeto rastreado correspondiente al identificador, se actualiza su información de trayectoria con el nuevo centroide.
8. **Visualización de resultados y detección de cruce de línea:** Finalmente, el sistema puede visualizar los objetos rastreados en el video. Este incluye la visualización de su ID y trayectoria. Además, se ha añadido la funcionalidad de detección de cruce de línea para conocer la dirección de movimiento de los objetos. El sistema verifica si un objeto ha cruzado una línea de conteo, lo que depende de la dirección de movimiento y permite saber si la persona está ingresando o saliendo de la biblioteca

Este proceso integral de seguimiento de centroides se aplica específicamente a objetos clasificados como 'persona', y su objetivo es proporcionar un análisis detallado del movimiento y conocer el número de personas que acceden y salen de la biblioteca.

4.5 MobilNetSSD vs YOLO3

	MobilNet SSD	YOLO3
Eficiencia y velocidad	Altamente eficiente y rápido en dispositivos móviles y embebidos.	Eficiente en términos de velocidad, pero más pesado en recursos.
Precisión	Buena precisión, pero puede ser menor en comparación con modelos más grandes.	Ofrece una alta precisión en la detección de objetos.
Consumo de recursos	Requiere menos recursos computacionales y memoria.	Requiere más recursos en términos de CPU y memoria.
Detección en tiempo real	Capacidad de detección en tiempo real en dispositivos con recursos limitados.	Realiza la detección en tiempo real, pero con mayores requisitos de hardware.
Aplicaciones variadas	Adecuado para una variedad de aplicaciones con restricción de recursos.	Utilizando en una amplia gama de aplicaciones, incluyendo aplicaciones críticas.

Tabla 1. Tabla MobileNet SSD vs YOLO3

En este proyecto se han probado ambos modelos comentados anteriormente, mientras se realizaban estas pruebas, se notaban diferencias notables en el rendimiento de ambos modelos, especialmente en el contexto de un servidor con recursos limitados.

Se realizaron pruebas con el modelo **YOLO3** utilizando vídeos de prueba offline, y se identificaron ciertos problemas significativos. En particular, se observó que el procesamiento de las imágenes era lento, lo que resultaba en una mayor latencia. Dado que el servidor cuenta con recursos limitados y se necesita un modelo que no imponga una carga elevada en términos de recursos, se tomó la decisión de no continuar con las pruebas de este modelo.

En el caso de **MobileNet SSD**, se encontró que el modelo procesaba las imágenes de manera más eficiente, lo que resultaba en menor latencia y una capacidad más constante para detectar objetos en tiempo real. Además, dado que este modelo está diseñado específicamente para dispositivos con recursos limitados y aplicaciones de tiempo real, se observó un rendimiento más fluido y rápido, lo cual era altamente beneficioso considerando las características y capacidad de cómputo del equipo servidor; como se busca una solución de bajo coste se ha probado a reutilizar equipos dados de baja por el servicio CRAI de la UPCT.

Debido a que ambos módulos de detección de objetos, en particular la clase 'persona', han sido entrenados utilizando el conjunto de datos COCO, las características de entrenamiento relacionadas con las imágenes de personas no coinciden exactamente con las que se encuentran en la biblioteca. Como resultado, ambas arquitecturas pueden mostrar errores considerables en la detección de personas, principalmente debido a que los datos y las características de aprendizaje del patrón "persona" no se asemejan lo suficiente a las imágenes de personas que se generan en la biblioteca.

Con el objetivo de mejorar la tasa de éxito en la detección de personas, se han identificado dos variables principales: el ángulo de la imagen (es decir, la inclinación de la cámara) y el área que se detecta en una persona.

En resumen, una vez entendido el principio de funcionamiento de la arquitectura de MobileNet SSD y que requiere menor carga computacional, la solución de diseño ha sido adaptar y customizar el módulo de conteo basado en MobileNet SSD y el módulo de tracking descrito. Debido a la influencia en el ángulo de inclinación de la cámara y su posición relativa respecto a la zona de paso para la tasa de éxito de detección de personas, se han hecho pruebas empíricas para encontrar la ubicación óptima de la cámara y el ajuste de parámetros de los módulos de detección y tracking.

4.6 Pruebas y ajustes de parámetros del módulo de conteo y de la ubicación y orientación de la cámara.

A partir de un análisis de datos reales de imágenes que muestran personas transitando por la zona de detección en la Biblioteca de Antigones, se ha observado empíricamente un tiempo promedio de aproximadamente dos segundos desde que una persona entra en el campo de visión de la cámara hasta que sale de él. Esta estimación es relevante debido a que el módulo de conteo realiza las operaciones de detección y seguimiento, y para que el seguimiento sea exitoso, una persona debe ser detectada de manera continua en varios frames.

Considerando que la cámara Tapo en la Biblioteca funciona a un promedio de 15,39 frames por segundo (**fps**), se estima que una persona debería aparecer en al menos 25-30 frames consecutivos durante su paso por la zona de detección.

La personalización de la solución tiene tres ejes principales:

1. De forma empírica, probar ángulos de inclinación y distancias de la cámara con respecto a la zona de paso en la biblioteca para lograr la mejor posición posible para la detección con éxito. Es decir, se trata de lograr que las imágenes captadas se asemejen más a los datos de aprendizaje del dataset COCO.
2. Zona de detección, posicionamiento línea boundary y tiempo medio de paso de una persona (para lograr un número mínimo de fps donde aparece una persona para su tracking).
3. Ajuste de parámetros de los modelos MobileNet SSD y tracking.

4.6.1 Ajuste de parámetros en los módulos de detección y tracking

En esta sección, se detallarán las pruebas realizadas para evaluar la calidad de la detección y el seguimiento de personas usando videos mp4 patrón. Para comprender plenamente los resultados, es importante considerar las diferentes posturas de la cámara y su posible impacto en los resultados. Antes de adentrarnos en los detalles de las pruebas, se proporcionará una breve introducción a las diferentes posturas de la cámara y como pueden influir en la eficacia del modelo.

Con el fin de llevar a cabo las pruebas de manera más eficiente, se desarrolló un código automatizado en Python que genera un csv con el resultado de las pruebas. Esta decisión se tomó en consideración del tiempo que requeriría realizar las pruebas manualmente, dada la variedad de videos de prueba disponibles de la biblioteca. En este contexto, se deben tener en cuenta tres parámetros ajustables que influyen en el rendimiento. En consecuencia, se asignaron valores a cada parámetro y se automatizó el proceso para evaluar todas las combinaciones posibles. Los tres parámetros son “skip_frames”, “confidence” y “media”:

- **“skip_frames”**: Se refiere al número de frames que deben omitirse antes de llevar a cabo una nueva detección de personas. Si se establece un valor más alto, se omitirán más frames, lo que resultará en una velocidad de procesamiento más rápida, pero con una precisión reducida. Por otro lado, si se establece un valor más bajo, se omitirán menos frames, lo que aumentará la precisión, pero a expensas de una velocidad de procesamiento más lenta.
- **“confidence”**: Su valor oscila en el rango de 0 a 1, lo que indica la probabilidad mínima requerida para filtrar detecciones de baja confiabilidad. En situaciones en las que la imagen de entrada es de alta calidad y los objetos son nítidos y de gran tamaño, es posible reducir el valor de confianza (confidence) con el fin de detectar un mayor número de objetos. No obstante, cuando la imagen presenta baja calidad o los objetos son pequeños y poco definidos, se recomienda aumentar el valor de confianza para minimizar los falsos positivos
- **“media”**: Su valor varía entre 0 y 255 y se refiere a un parámetro utilizado para escalar la imagen de entrada antes de realizar la detección de objetos. Variar este parámetro afectará cómo se procesa la imagen, lo que puede aumentar la sensibilidad a los detalles de la imagen, mientras que reducirlo puede disminuir la sensibilidad y el ruido en la detección. Aumentar el valor puede ser beneficioso si las imágenes de entrada tienen una calidad baja o están subexpuestas, mientras que disminuir el valor es adecuado para imágenes de alta calidad con un buen contraste.

También se realizaron pruebas con diferentes posiciones de la línea divisoria entre la parte superior e inferior. En las imágenes donde había muchos obstáculos se optó por recortar el

encuadre con el fin de centrarlo más y reducir la presencia de obstáculos. Estos ajustes se pueden hacer desde un archivo de configuración.

4.6.2 Estudio de la posición y ángulo de inclinación de la cámara respecto al área de detección y paso

La cámara utilizada tiene la capacidad de rotar 360° en el plano horizontal y 114° en el plano vertical, lo que permite ajustar el ángulo de visión según las necesidades. La principal dificultad que se presentó fue la instalación de la cámara en el techo, para la cual se requirió la asistencia de técnicos especializados. Se realizaron diversas pruebas con distintos ángulos de la cámara. A continuación, se explican los tipos de planos que puede tomar una cámara más interesante.



Figura 35. Posibles planos de fotografía

- **Plano cenital:** En este plano, la cámara está posicionada directamente sobre el sujeto o la escena, apuntando hacia abajo perpendicularmente. Esta perspectiva ofrece una visión despejada sin preocupaciones acerca de posibles obstrucciones por parte de otros individuos
- **Plano picado:** En este caso, la cámara se coloca por encima del sujeto o la escena, pero en un ángulo inclinado hacia abajo. Aunque sigue proporcionando una vista clara, puede haber momentos en los que la obstrucción se convierta en un factor a considerar
- **Plano Normal:** Este plano, que muestra a los personajes o la acción desde una distancia media, no garantiza una visión totalmente despejada, ya que podría darse el caso de obstrucciones entre individuos.

En la Figura 36, se muestra el esquema que representa la posición y el ángulo de inclinación de la cámara con respecto al arco de entrada y a la zona de paso y detección objetivo en la que se ha colocado. Este esquema ilustra la posición que la cámara mantuvo durante las pruebas realizadas, aunque se varió el ángulo para simular diferentes planos.

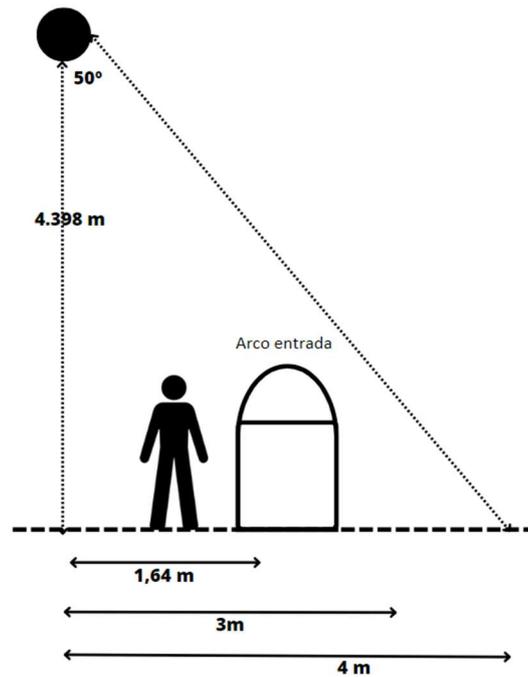


Figura 36. Esquema de posición de la cámara

Como se ha comentado previamente, remarcar que el modelo ha sido pre-entrenado con el dataset COCO. Por lo tanto, existen situaciones en las que es posible que no detecte a las personas de manera precisa, ya que las imágenes de personas en la biblioteca pueden diferir significativamente de las imágenes utilizadas durante el proceso de aprendizaje del modelo. Esto puede dar lugar a detecciones e identificaciones imprecisas de las personas. A continuación, se analizarán los resultados obtenidos en función de los diversos ángulos de la cámara.

Pruebas con ángulo cenital

En esta sección, se comentarán los resultados de las pruebas realizadas con el ángulo de cámara cenital. Como se mencionó anteriormente, el plano cenital implica tener una imagen desde arriba.

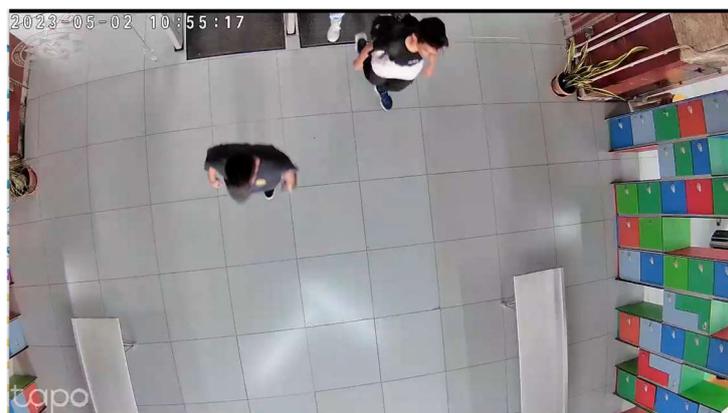


Figura 37. Plano cenital, imágenes de la biblioteca

Se llevaron a cabo pruebas con videos tanto en blanco y negro como en color. En ninguno de los casos se detectaron personas, ya que el modelo pre-entrenado no las identificaba como tales debido a la inclinación de la cámara.

Skip Frames	Confidence Config	Media	Count In Real	Count Out Real	Count In Estimated	Count Out Estimated	Error In Abs	Error Out Abs	Error Global	Detecciones
15	0.3	127	10	23	8	5	2	18	0.9826087	13
15	0.3	135	10	23	7	4	3	19	1.12608696	10
15	0.3	135	10	23	7	4	3	19	1.12608696	10
15	0.3	115	10	23	7	3	3	20	1.16956522	10
15	0.4	127	10	23	7	3	3	20	1.16956522	10
15	0.4	135	10	23	7	2	3	21	1.21304348	8
15	0.4	115	10	23	6	3	4	20	1.26956522	9
25	0.3	127	10	23	6	3	4	20	1.26956522	8
20	0.3	115	10	23	7	1	3	22	1.25652174	7

Tabla 2. Resultados de las pruebas en plano cenital

Como se puede observar, los resultados son insatisfactorios, ya que se presentan errores de detección que afectan negativamente a la precisión del conteo.

Pruebas con ángulo normal

En esta sección se comentarán los resultados obtenidos tras las pruebas realizadas con el ángulo de cámara normal. Como se comentaba es un plano desde un ángulo más recto.

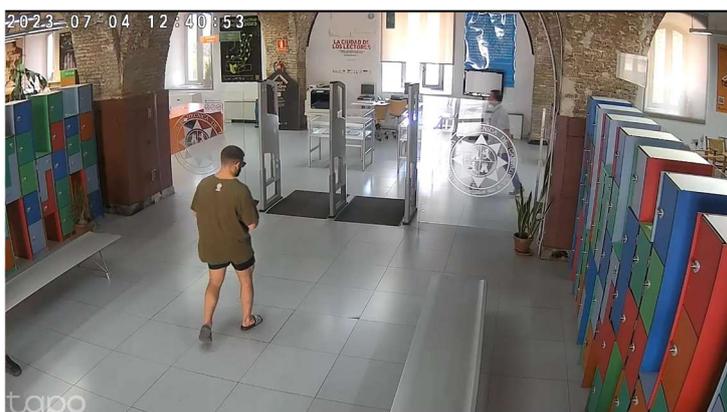


Figura 38. Plano normal, imágenes de la biblioteca

Se llevaron a cabo pruebas con respecto a este plano de captura. Es cierto que este plano es eficaz en la detección de todas las personas presentes. Sin embargo, se ha observado que la detección se ve afectada en casos de oclusión, donde la visión de las personas se bloquea mutuamente. La detección es más efectiva cuando las personas pasan individualmente, evitando así cualquier interferencia entre ellas. Una perspectiva más centrada podría ser preferible en términos de detección, pero debido a restricciones en la ubicación de la cámara, esta disposición resultó ser la más conveniente.

A pesar de su capacidad de detección precisa, se ha identificado una limitación en el seguimiento de las personas en esta perspectiva frontal. Debido a la vista frontal, la dirección vertical del movimiento (hacia arriba o hacia abajo) no puede ser adecuadamente apreciada, lo que afecta negativamente la calidad del seguimiento. Sería posible mejorar tanto el conteo como el seguimiento si se pudiera centrar la cámara y elevarla para obtener una perspectiva óptima.

Debido a que se logró una detección exitosa, pero se experimentó dificultades en el tracking de algunas personas detectadas, los resultados de las pruebas se consideraron insatisfactorios ya que no se hace bien el conteo.

Skip Frames	Confidence	Media	Count In Real	Count Out Real	Count In Estimated	Count Out Estimated	Error In Abs	Error Out Abs	Error Rel Global	Detecciones
15	0.5	85	8	3	6	6	2	3	1.25	11
15	0.4	127	8	3	3	9	5	6	2.625	11
15	0.5	127	8	3	3	9	5	6	2.625	11
30	0.4	135	8	3	5	7	3	4	1.70833333	11
30	0.5	135	8	3	5	7	3	4	1.70833333	11
15	0.4	115	8	3	4	8	4	5	2.16666667	11
15	0.5	115	8	3	4	8	4	5	2.16666667	11
30	0.3	135	8	3	4	8	4	5	2.16666667	11
30	0.4	115	8	3	3	8	5	5	2.29166667	11
30	0.5	115	8	3	3	8	5	5	2.29166667	11
30	0.4	127	8	3	2	10	6	7	3.08333333	11
30	0.5	85	8	3	2	10	6	7	3.08333333	11
30	0.5	127	8	3	2	10	6	7	3.08333333	11
30	0.3	127	8	3	1	11	7	8	3.54166667	11

Tabla 3. Resultados de las pruebas en plano normal

En la tabla 3 se muestra un ejemplo de los mejores resultados obtenidos. Como podemos observar, las detecciones equivalen a la suma de entradas y salidas reales, sin embargo, debido al error de seguimiento que se comentaba, el error relativo global es muy alto.

Pruebas con ángulo picado

En esta sección, se comentarán los resultados obtenidos tras las pruebas realizadas con el ángulo de cámara picado. Como se comentaba es un plano desde un ángulo inclinado desde arriba.



Figura 39. Plano picado, imágenes de la biblioteca 1

En relación a la perspectiva en picado, se llevaron a cabo pruebas desde diversas distancias y ángulos con el objetivo de determinar la configuración más adecuada.

Los resultados obtenidos en los videos capturados desde la perspectiva representada en la Figura 39 fueron más satisfactorios en comparación con los planos anteriores. No obstante, se observó un porcentaje reducido de detecciones falsas, principalmente debido a obstáculos como las alfombrillas negras y al ángulo de captura desde una posición elevada. Además, es importante destacar que esta perspectiva presenta limitaciones en términos de su margen de detección, ya que no abarca una gran distancia.

Skip Frames	Confidence Config	Media	Count In Real	Count Out Real	Count In Estimated	Count Out Estimated	Error In Abs	Error Out Abs	Error Global	Detecciones
25	0.4	135	3	7	2	10	1	3	0.76190476	14
20	0.4	95	3	7	1	12	2	5	1.38095238	13
30	0.3	127	3	7	7	6	4	1	1.47619048	14
30	0.3	135	3	7	6	7	3	0	1	14
30	0.4	85	3	7	6	7	3	0	1	13
30	0.4	95	3	7	6	7	3	0	1	12
30	0.4	127	3	7	7	6	4	1	1.47619048	12

Tabla 4. Resultados de las pruebas en plano picado posición 1

Como se puede observar, se producen detecciones de falsos positivos, lo cual tiene un impacto negativo en los resultados. Sin embargo, es importante destacar que el seguimiento de las detecciones reales se realiza correctamente.



Figura 40 Plano picado, imágenes de la biblioteca 2

Los resultados obtenidos en los videos capturados desde la perspectiva representada en la Figura 40 fueron más satisfactorios en comparación con el plano de la Figura 39. A pesar de que persisten detecciones falsas debido a la presencia de obstáculos, se registró una disminución en su frecuencia. Esto se debe a la mayor distancia y la mayor inclinación de la cámara, lo que permite una mejor visualización del recorrido de las personas y, como resultado, un seguimiento más preciso de las detecciones.

Skip Frames	Confidence Config	Media	Count In Real	Count Out Real	Count In Estimated	Count Out Estimated	Error In Abs	Error Out Abs	Error Global	Detecciones
15	0.1	135	2	3	2	3	0	0	0	5
15	0.2	135	2	3	2	3	0	0	0	5
5	0.5	135	2	3	3	3	1	0	0.5	5
15	0.1	95	2	3	3	3	1	0	0.5	5
15	0.2	95	2	3	3	3	1	0	0.5	5
15	0.3	95	2	3	3	3	1	0	0.5	5
15	0.3	115	2	3	4	3	2	0	1	6

Tabla 5. Resultados de las pruebas en plano picado posición 2

Como se puede observar, los resultados muestran una notable mejora con respecto a las configuraciones anteriores. Las detecciones se realizan de manera precisa, aunque se presentan pequeños errores en el seguimiento.



Figura 41. Plano picado, imágenes de la biblioteca 3

Los resultados obtenidos en los videos capturados desde la perspectiva representada en la Figura 41 superaron significativamente a las configuraciones de la Figura 40 y Figura 39 en términos de satisfacción. La mayor distancia y el ángulo de captura más pronunciado contribuyeron de manera notable a una mejora en la precisión de las detecciones. En este escenario, la presencia de numerosos obstáculos en la imagen y una región no relevante requirieron un ajuste del encuadre, mediante el recorte de la imagen desde el archivo de configuración, con el fin de centrarse en la detección y seguimiento de personas. Esta perspectiva ha arrojado los resultados más destacados, como se puede apreciar en la siguiente tabla:

Skip Frames	Confidence Config	Media	Count In Real	Count Out Real	Count In Estimated	Count Out Estimated	Error In Abs	Error Out Abs	Error Global	Detecciones
5	0.1	95	6	6	6	6	0	0	0	12
5	0.2	95	6	6	6	6	0	0	0	12
5	0.3	95	6	6	6	6	0	0	0	12
5	0.4	85	6	6	6	6	0	0	0	12
25	0.1	135	6	6	6	6	0	0	0	12
25	0.2	135	6	6	6	6	0	0	0	12
25	0.3	127	6	6	6	6	0	0	0	12
25	0.3	135	6	6	6	6	0	0	0	12
25	0.4	127	6	6	6	6	0	0	0	12
25	0.4	135	6	6	6	6	0	0	0	12
30	0.1	95	6	6	6	6	0	0	0	12

Tabla 6. Resultados de las pruebas en plano picado posición 2

Como se puede observar, los resultados son altamente satisfactorios, ya que tanto la detección como el seguimiento se realizan de manera precisa. Tras todas las pruebas realizadas bajo esta perspectiva angular, se llega a la conclusión de que los mejores resultados se obtienen al establecer los siguientes parámetros: un valor de "skip_frames" de 25, una confianza ("confidence") de 0.3 o 0.4, y un valor de "media" de 135. Estos valores, han demostrado ser los más efectivos.

4.6.2 Conclusiones de las pruebas

Se han generado videos de prueba mp4 de corta duración para distintos ángulos y posiciones de la cámara, donde se conoce el dato de total de entradas y total de salidas de personas, y que sirven como vídeos patrón para evaluar las configuraciones del módulo de conteo.

A partir de la observación del comportamiento de las personas en la zona de paso y control, se ha calculado un tiempo promedio de 2 segundos para el paso de una persona. Esto se traduce en una estimación de que una persona aparece en la imagen durante un intervalo de 20 a 30 fps.

Durante la realización de estas pruebas de detección y seguimiento de personas se han realizado varias pruebas con diferentes videos, utilizando diversas perspectivas de cámara, se ha observado que la tercera posición en picado, representada en la **Figura 41**, ha proporcionado los resultados más satisfactorios. Esta perspectiva, se caracteriza por una mayor distancia y un ángulo de captura más pronunciado, lo que se tradujo en una mejora significativa en la precisión de las detecciones y un seguimiento más confiable de las personas detectadas.

Las pruebas realizadas con otras perspectivas, como las mostradas en las Figuras 39 y 40, revelaron desafíos relacionados con la detección de obstáculos y la necesidad de realizar recortes en el encuadre para optimizar la detección y el seguimiento. Sin embargo, la tercera posición en picado minimizó estos desafíos y proporcionó una mayor coherencia en los resultados.

Basándonos en los datos recopilados y la consistencia de los resultados, se determina que la perspectiva en picado, las imágenes en color, desde la tercera posición, es la opción preferida para llevar a cabo el conteo de personas utilizando imágenes de cámara en este contexto específico. Además, se han identificado los siguientes valores óptimos para los parámetros clave del módulo de conteo: un valor de 'skip_frames' de 25, un valor de 'confidence' de 0.3 o 0.4, y un valor de 'media' de 135. Estos valores han demostrado ser los más efectivos en las pruebas realizadas.

También se proponen las siguientes directrices y recomendaciones para orientar la ubicación de la cámara en el contexto de los arcos de entrada a una biblioteca:

- Se sugiere que la cámara enfoque el área de paso antes de los arcos de entrada, ya que proporciona una imagen más despejada de obstáculos. Se plantea ubicar la línea de límite (boundary) aproximadamente a una distancia de 1 a 1,5 metros del arco de entrada.
- Se parte de una distancia de paso estimada para la detección de una persona de alrededor de 2 metros, con un tiempo promedio de paso de 2 segundos. Teniendo en cuenta el promedio de 15 fps de la cámara Tapo en un streaming RTSP con la red Wifi IoTUT, se estiman unos 20-30 fps seguidos donde se muestre una persona.
- Se recomienda colocar la cámara a una distancia que oscile entre 4 y 5 metros del arco de entrada, con el fin de garantizar una visualización adecuada de la zona de paso.

Además, se aconseja fijar la altura de la cámara en al menos 2,5 metros con respecto al suelo y ajustar su inclinación a unos 30 grados, orientándola hacia la zona de paso.

Estas recomendaciones son directrices que deben ser refinadas y validadas según las condiciones específicas de instalación en otra biblioteca. No obstante, se ha verificado que son requisitos asumibles y factibles para una implementación de bajo costo.

La instalación de la cámara ha requerido la debida autorización tanto del servicio CRAI como del Vicerrectorado de Campus y Sostenibilidad, que es responsable de las infraestructuras de la UPCT. En este sentido, únicamente los técnicos de la Unidad Técnica cuentan con la autorización necesaria para llevar a cabo la instalación en el techo de la cámara.

A pesar de que las pruebas en un entorno real y con imágenes reales resultan esenciales, la tarea de ajustar y modificar la posición de la cámara en la Biblioteca de Antigones ha sido un desafío considerable, con limitaciones que hemos tenido que abordar. Sin embargo, hemos llegado a la conclusión de que existe una posición y orientación óptima para la cámara, lo que maximiza la tasa de éxito en el conteo. En esta configuración, las imágenes capturadas de las personas se asemejan de manera más cercana al conjunto de datos de entrenamiento COCO.

Finalmente, es importante destacar que, dado que la cámara Tapo es una solución de bajo costo que permite ajustar el ángulo y el zoom a través de una aplicación móvil, los requisitos de posición y orientación resultan viables no solo para la Biblioteca de Antigones, sino también para otras dependencias de la UPCT.

Capítulo 5. Diseño e implementación del servidor del sistema de monitorización de la ocupación de la biblioteca

Destacar que se ha planificado la implementación de una red **WIFI IoTUT** independiente, la cual será utilizada para las cámaras de visión y para el servidor de conteo de personas.

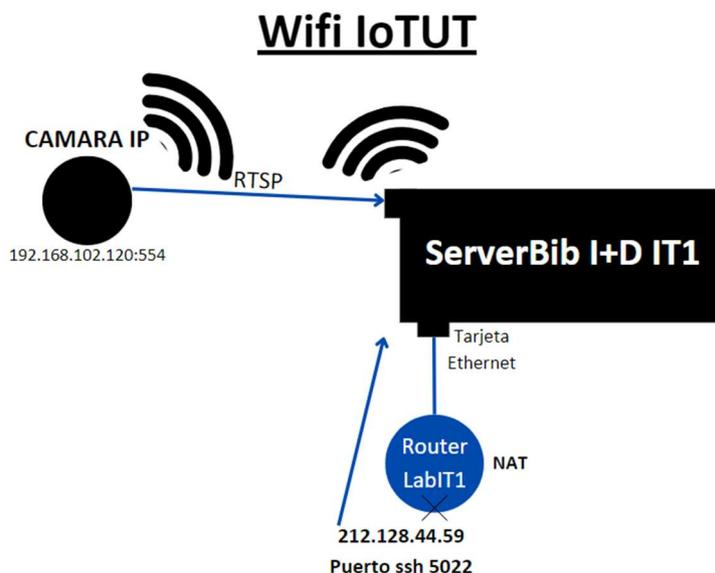


Figura 42. Esquema conexiones

En el Server Biblioteca hay dos tarjetas de red:

1. **Tarjeta Wifi:** configurada para asociarse a la Wifi IoTUT exclusiva de la Unidad Técnica de la UPCT y gestionada y monitorizada por el Servicio de Informática. El uso de esta red Wifi permite la comunicación segura con la cámara IP instalada en la biblioteca de Antigones. Se han pedido los permisos oportunos y notificado datos de uso para que el tráfico del servidor no sea filtrado por el servicio de informática.
2. **Tarjeta Ethernet:** conectada al router del laboratorio I+D IT1. En el router del laboratorio IT1 se ha configurado la redirección del puerto 22 SSH para el acceso remoto y seguro al servidor.

En el servidor se encuentra el programa de conteo, en este mismo se almacenan los datos de ocupación y se puede visualizar la ocupación que ha habido durante el día en la biblioteca, además, para poder informar a los alumnos de la ocupación que hay y que los administradores sepan el estado de la cámara se ha implementado dos bots de Telegram.

A continuación, se muestra el esquema del proyecto, se representan las cámaras, la estructura del backend y la base de datos:

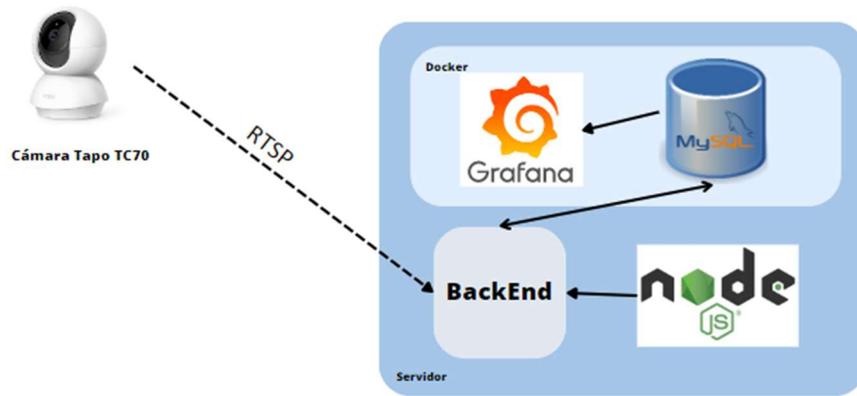


Figura 43. Esquema general cámara y servidor

La cámara envía las imágenes en tiempo real al servidor, el programa de conteo se encarga de procesarlas y almacenar la ocupación en la base de datos **MySQL**. Los datos de ocupación almacenados en la BBDD son utilizados por Grafana para mostrarlos en una gráfica y para enviar a través del bot de Telegram la ocupación. **NodeJS**, es el framework que se usa para poder usar el administrador **PM2**.

5.1 Arquitectura global del sistema

Los tres elementos principales de este sistema son los siguientes:

- **Cámara Tapo**, que se utiliza para obtener las imágenes y debido a que permite RTSP se puede leer las frames que se envían.
- **BBDD MySQL**, es la base en la que se almacenan todos los registros de ocupación y posteriormente de la que se obtienen para visualizarlos en Grafana o para enviarlos por Telegram.
- **Back-End**, dentro de este tenemos varias funcionalidades:
 - La más importante es que aloja el programa principal de conteo, que es el encargado de realizar el conteo de personas que acceden y salen de la biblioteca en la BBDD.
 - También tenemos la funcionalidad de envío de alertas y notificaciones al bot de **Telegram** para ambos canales.
 - La tercera funcionalidad que tiene es un **HTML** que define una página web que incluye un formulario. El formulario permite a los administradores enviar una solicitud POST a una URL API de Telegram.

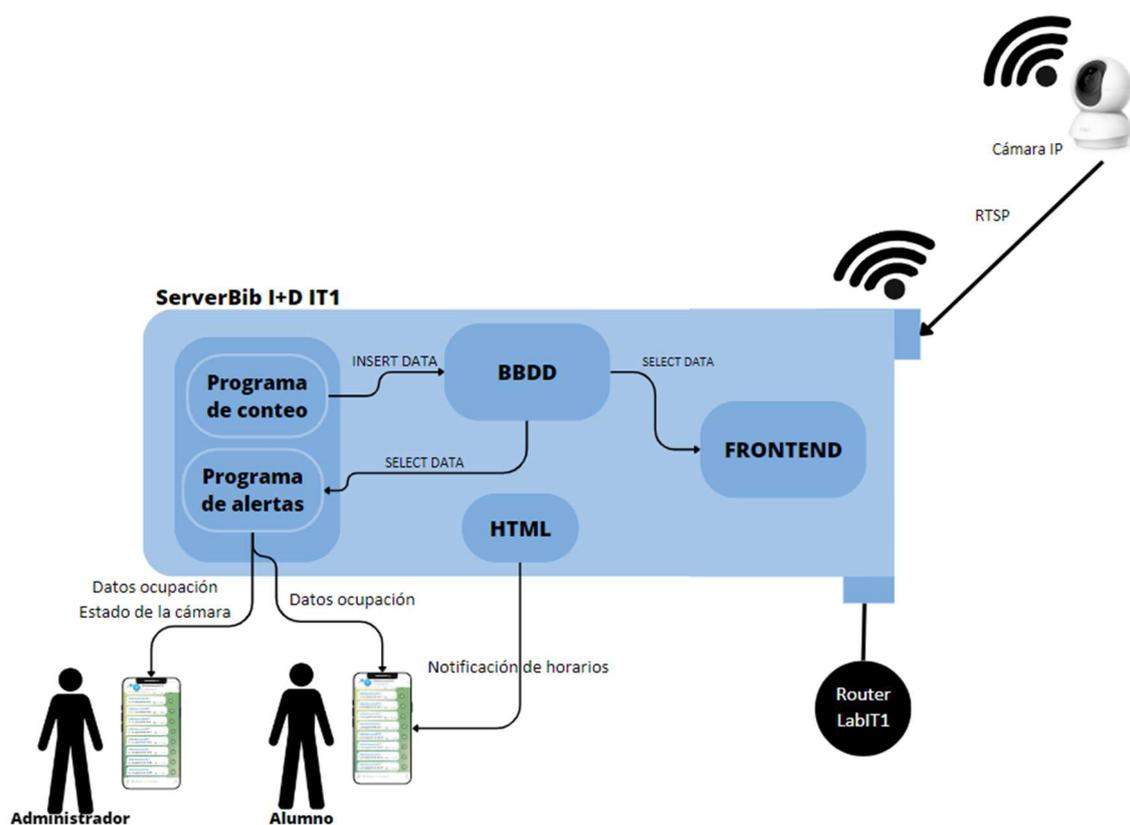


Figura 44. Esquema del sistema

A continuación, se explica en detalle el funcionamiento global del sistema:

1. La **cámara** utiliza el método RTSP (Protocolo de Control de Transmisión en Tiempo Real) para enviar imágenes en forma de frames en tiempo real al servidor. **RTSP** es un protocolo de comunicación que se emplea para la transmisión de contenido multimedia, como video y audio, a través de redes IP. Funciona como un mecanismo de control que establece y gestiona las conexiones para la entrega de datos multimedia en tiempo real. [35] En este contexto es interesante ya que el servidor puede procesar y utilizar las imágenes capturadas en tiempo real, lo que resulta fundamental para el funcionamiento del sistema de conteo de personas.
2. En el **backend** se encuentra el programa de conteo, que recibe los frames de la cámara ya que se inicia una transmisión de video desde la URL RTSP, estas imágenes que se reciben son procesadas por el programa de conteo y almacena datos de ocupación en la BBDD.
3. La **BBDD** se encarga de almacenar todos los registros de ocupación recibidos del programa de conteo y esta información queda accesible para otras funcionalidades para el uso de dichos registros.
4. En el backend está la funcionalidad de envío de alertas y notificaciones por **Telegram** que habíamos comentado antes, esta utilizará el último dato almacenado en la BBDD, para poder informar a los usuarios de la ocupación.
5. **Grafana** también utilizará los registros almacenados en BBDD para mostrarlos en una gráfica.

6. El **HTML**, se refiere a un formulario que se encuentra en el servidor, el cual permite introducir los horarios en caso de sufrir cambios y se envían por Telegram.

Los dos programas que se tienen en el backend, los cuales son el encargado de envío de alertas y el programa principal de conteo, son gestionadas con **PM2**, este permite mantenerlas vivas, y en caso de algún error lo podemos monitorizar por el log de PM2, además se encarga de recargarlas evitando tiempos de inactividad.

id	name	namespace	version	mode	pid	uptime	♻	status	cpu	mem	user	wa
0	Bots	default	N/A	Fork	4314	84s	0	online	0%	94.0mb	adm...	disabl
1	Run	default	N/A	Fork	4356	66s	0	online	100%	205.6mb	adm...	disabl

Figura 45. Lista de programas gestionadas por PM2.

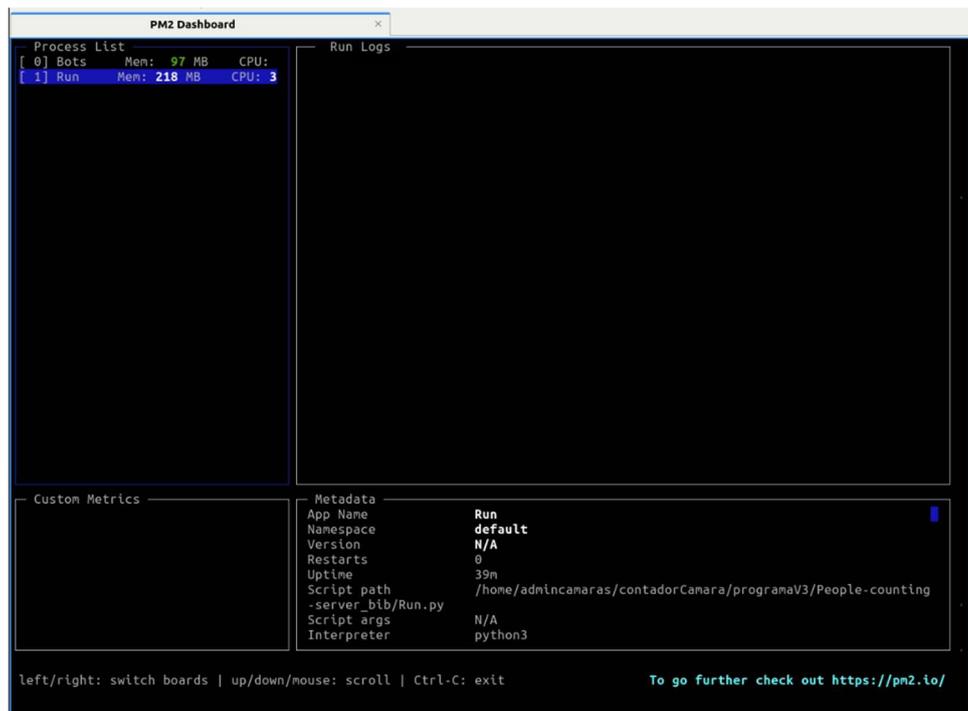


Figura 46. Imagen de Dashboard PM2

En la Figura 46 se puede observar la pestaña que aparece al ejecutar el comando 'pm2 monit', que permite visualizar el uso de recursos (memoria y CPU), también se puede ver el estado, el tiempo de actividad y el número de reinicios de cada proceso, a parte tenemos la ventana de 'Run Logs', que en caso de saltar algún error se mostrará ahí.

5.2 Arquitectura del backend y BBDD MySQL

Este programa de conteo que se presenta en este proyecto se deriva de un código fuente preexistente, accesible en el repositorio de GitHub del usuario Sai Subhakar T. [36] Este repositorio sirvió como punto de partida, sobre el cual se aplicaron mejoras y adaptaciones específicas para satisfacer los requerimientos y objetivos del proyecto.

Hay que tener en cuenta que el código está utilizando un modelo pre-entrenado de detección de objetos que se carga desde un archivo de configuración y pesos pre-entrenados.

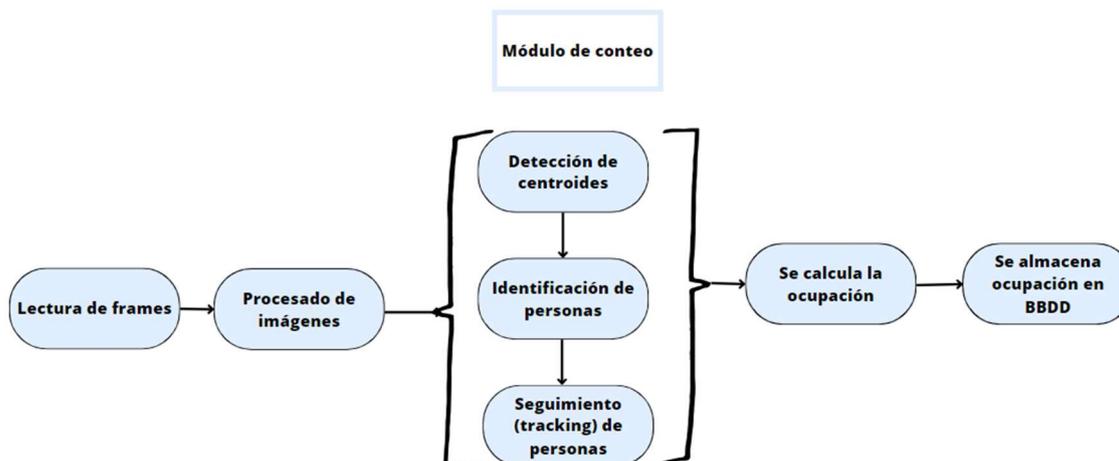


Figura 47. Esquema del funcionamiento del programa de conteo Run.py

El esquema muestra el funcionamiento interno del programa principal, a continuación, se explica:

1. **La lectura de frames:** Se extrae el flujo de imágenes de manera continua de la cámara IP a través de RTSP.
2. **Módulo de conteo de personas:** Esto se ha explicado en detalle en el capítulo anterior en la sección 4.4.
 - **Detección de centroides:** Al recibir suficientes frames para poder empezar a procesar las imágenes, se comienza a realizar la detección de centroides.
 - **Identificación de personas:** Tras la detección de centroides para cada frame, se identifica los objetos que sean 'persona'.
 - **Seguimiento (tracking) de personas:** Tras la identificación de personas se realiza el seguimiento de estas. Luego se comprueba si la persona ha cruzado la línea de conteo, dependiendo de la dirección del movimiento, indicara si la persona está accediendo o saliendo de la biblioteca
3. **Se calcula la ocupación:** Como se ha comentado en el punto anterior, dependiendo de la dirección de la línea y del movimiento del objeto, se actualiza los datos de entrada y salida y se consiguen los datos de ocupación.
4. **Se almacena ocupación en BBDD:** Al obtener la ocupación si su valor ha cambiado desde el último dato obtenido, se llama a una función que se encarga de almacenar este registro en la BBDD.

fecha_hora	ocupacion
2023-07-17 21:11:43	9
2023-07-17 21:11:44	8
2023-07-17 21:12:05	7
2023-07-17 21:12:07	6
2023-07-17 21:12:09	5
2023-07-17 21:12:09	4
2023-07-17 21:12:11	3
2023-07-17 21:12:13	2
2023-07-17 21:12:15	1
2023-07-17 21:12:16	0

Figura 48. Columnas de la BBDD



Figura 49. Imagen de la cámara de la biblioteca

Además, se ha incorporado una funcionalidad que regula su ejecución en base a una franja horaria predefinida. Esta función asegura que el programa solo pueda ejecutarse dentro del periodo de tiempo específico.

El funcionamiento de esta implementación es el siguiente: cuando el programa se inicia, se llama a una función donde se verifica si está en la franja horaria permitida. Si se encuentra en la franja, el programa se ejecuta como normalmente hasta que termine la franja horaria. Si no está en la franja horaria permitida, el programa de conteo deja de ejecutarse, quedando en estado de espera hasta que se encuentre dentro del intervalo de tiempo válido.

Esto garantiza que solo se ejecute durante los días que va a estar abierta la biblioteca y en el horario que hay configurado.

En el diagrama de flujo de la Figura 50, se representa el proceso que sigue el programa de alertas. La decisión de crear un programa independiente de alertas se tomó debido a su funcionamiento asíncrono, lo que resultó ser la opción más adecuada para evitar interferir con el programa de conteo. El proceso es simple: se verifica el estado de la cámara, y si este cambia, se envía una alerta al bot de administradores. Además, se verifica el dato más reciente de ocupación y se envía tanto al bot de administradores como al de alumnos. Estas notificaciones se envían de forma periódica, cada media hora.

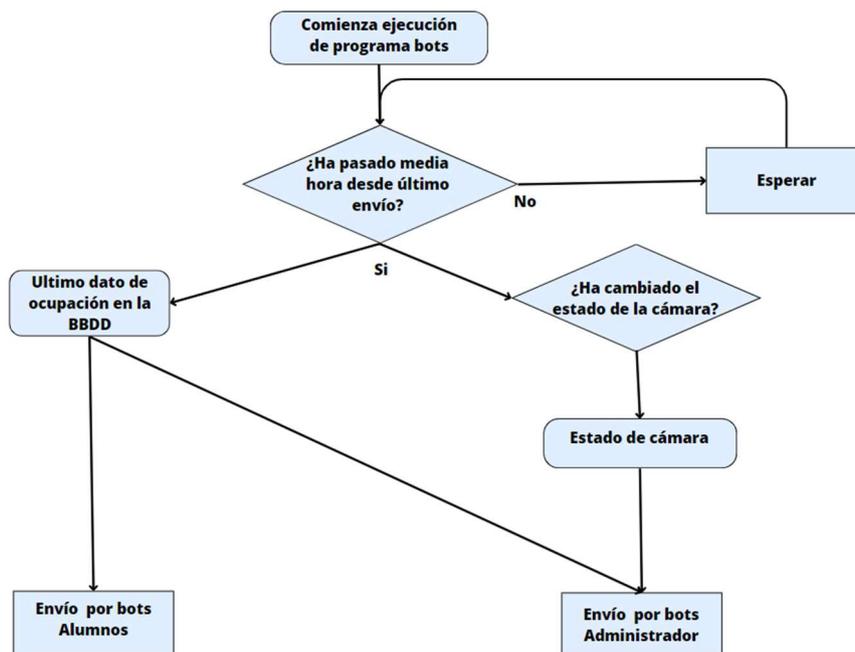


Figura 50. Diagrama de flujo programa envío de alertas por bots

Añadir, que el servidor ha sido diseñado con un enfoque en la preservación de la privacidad y el cumplimiento de la Ley Orgánica de Protección de Datos (LOPD). Es importante destacar que el servidor no almacena ningún video del streaming ni ninguna imagen de personas de forma permanente. Tan pronto como se reciben los frames del streaming en tiempo real, son procesados para llevar a cabo el conteo de personas, y no son almacenadas.

5.3 Frontend de visualización de datos de conteo con Grafana

Para poder visualizar los datos de ocupación en una gráfica se ha utilizado **Grafana**, Grafana es una plataforma que permite a los usuarios ver sus datos a través de tablas y gráficos que se unifican en un panel de control (o en varios) para facilitar la interpretación y la comprensión, además es de código abierto [38], el usuario descarga la aplicación y ya la puede usar, se puede configurar para personalizar la forma de visualizar los datos de interés de la manera que se desee.

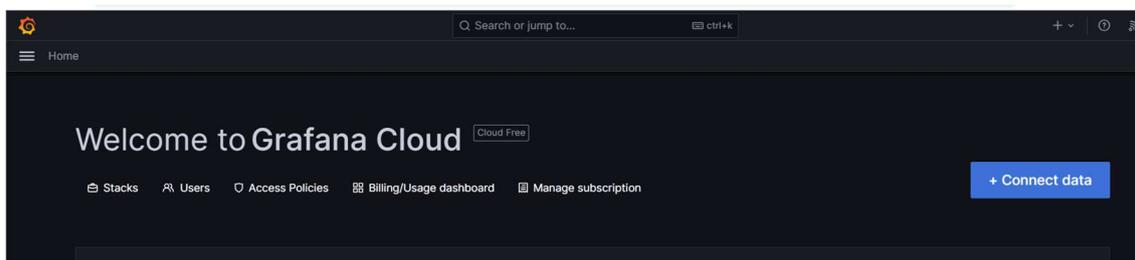


Figura 51. Página de inicio de Grafana

Para conectar la base de datos, se debe acceder a la opción 'Data Source' en la sección de configuración de la barra lateral izquierda y se selecciona el botón 'Add data source'.

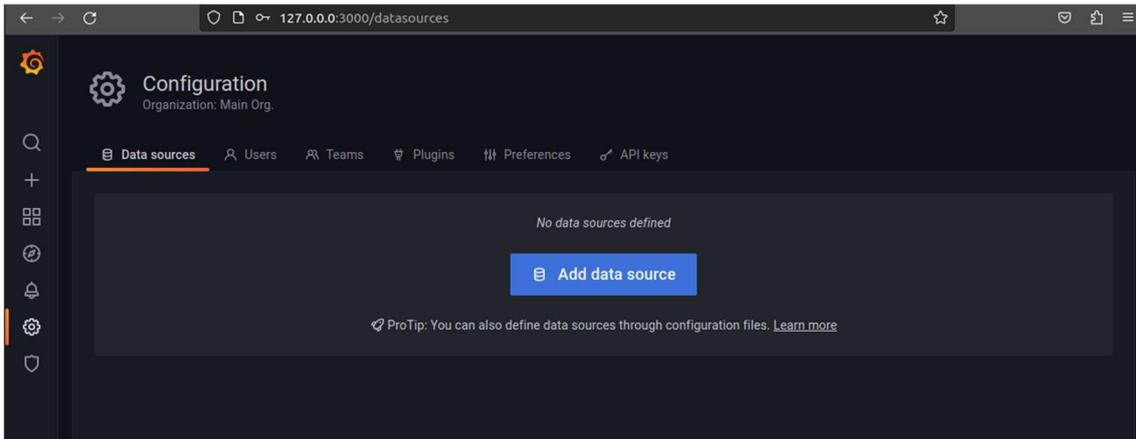


Figura 52. Configuración de la conexión BBDD parte 1

En el menú que aparece se debe buscar el tipo de base de datos se está utilizando, en este caso será **MySQL**.

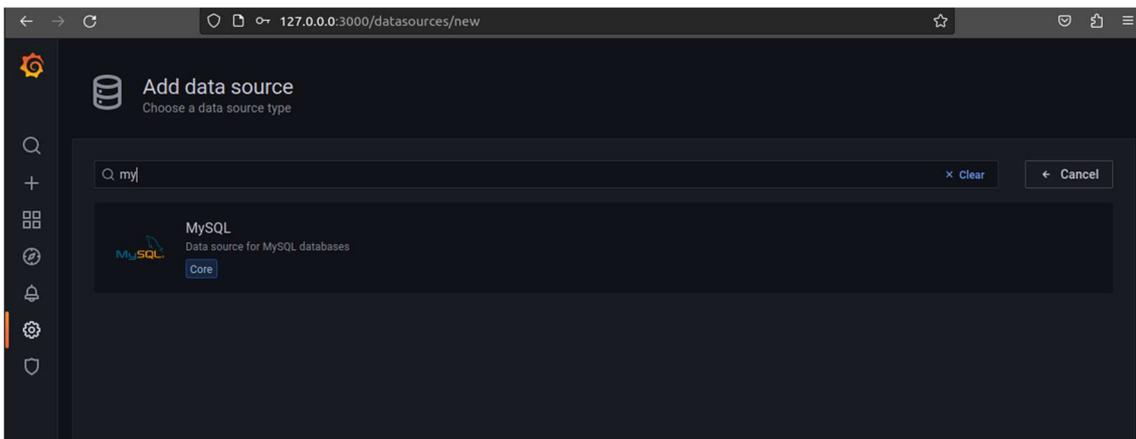


Figura 53. Configuración de la conexión BBDD parte 2

Al seleccionar MySQL, se tendrá que rellenar los campos para la conexión.

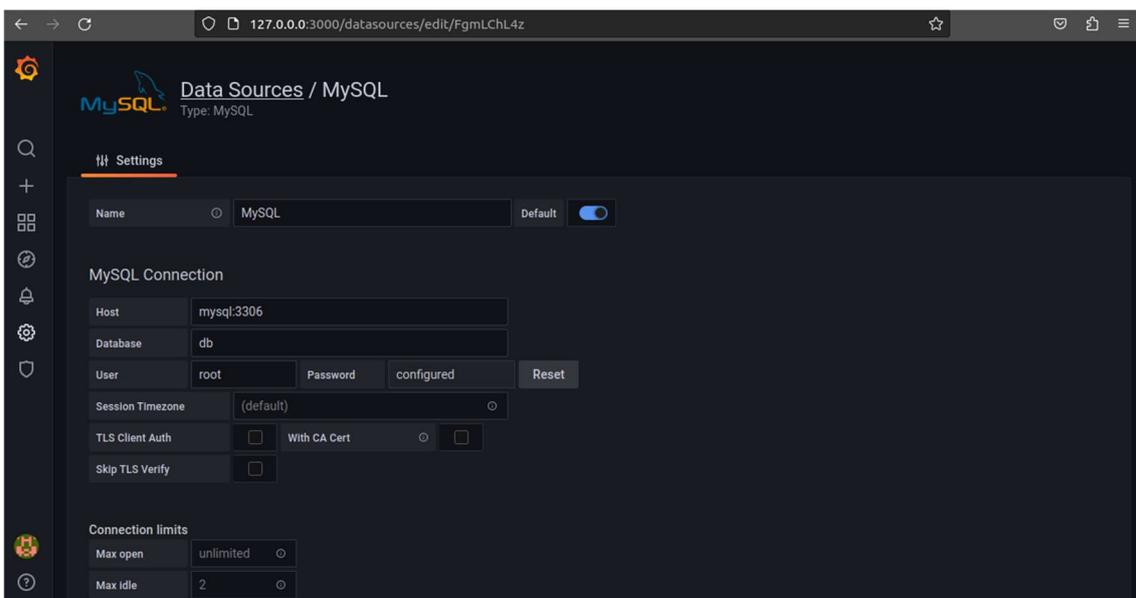


Figura 54. Configuración de la conexión BBDD parte 3

Tras rellenar los campos se tiene que comprobar la conexión, para ello se selecciona la opción de 'Save & Test' y se podrá ver como la conexión esta 'OK'.

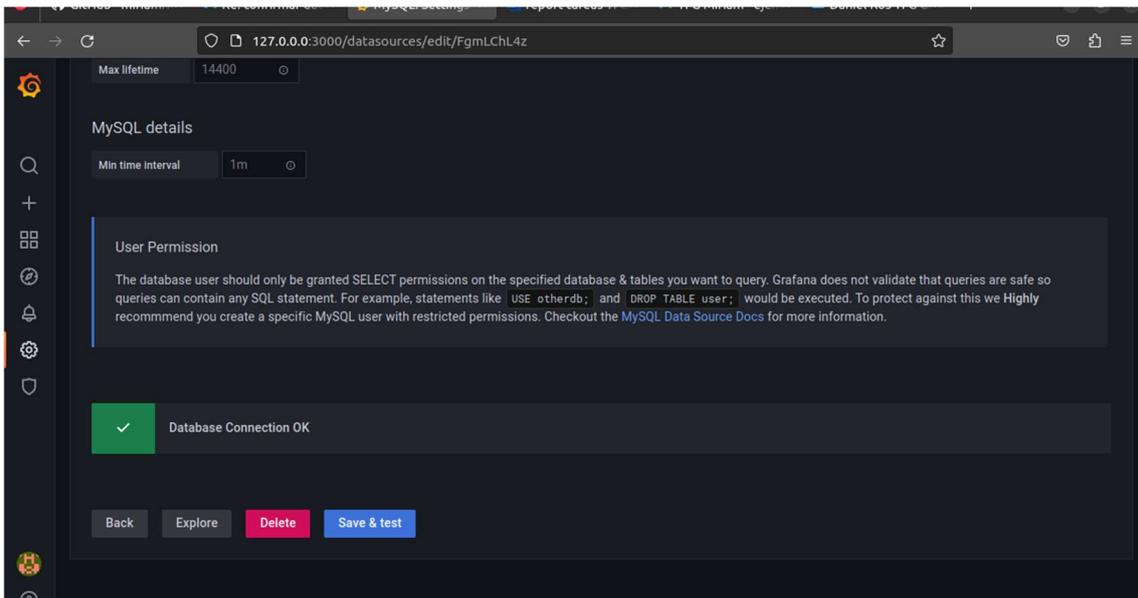


Figura 55. Configuración de la BBDD parte 4

Para crear paneles o graficas con los datos hay que acceder al menú y seleccionar 'Create' y dentro seleccionar 'Dashboard' y finalmente se selecciona 'Add new panel'.

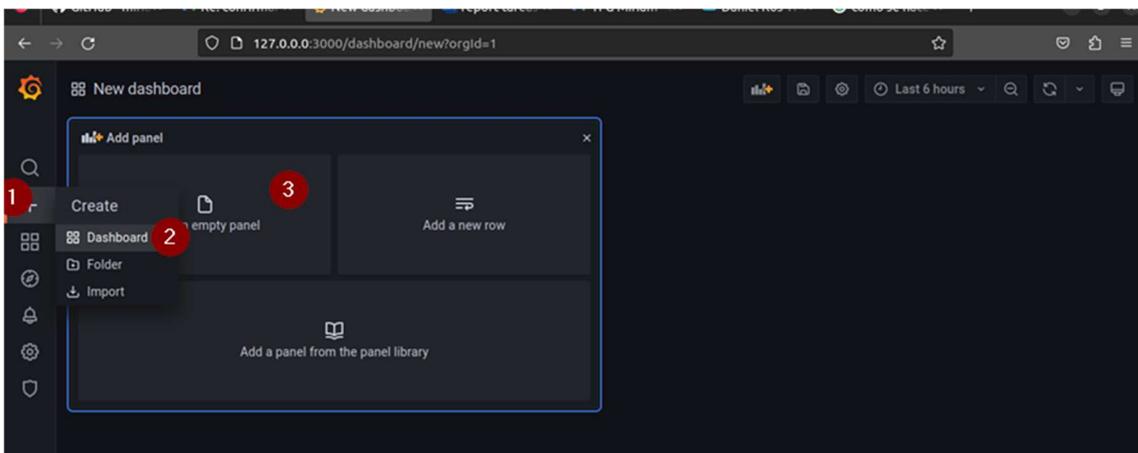


Figura 56. Configuración gráfica parte 1

Tras seleccionar 'Add new panel', se escribe la consulta que coincida con lo que se quiera mostrar.

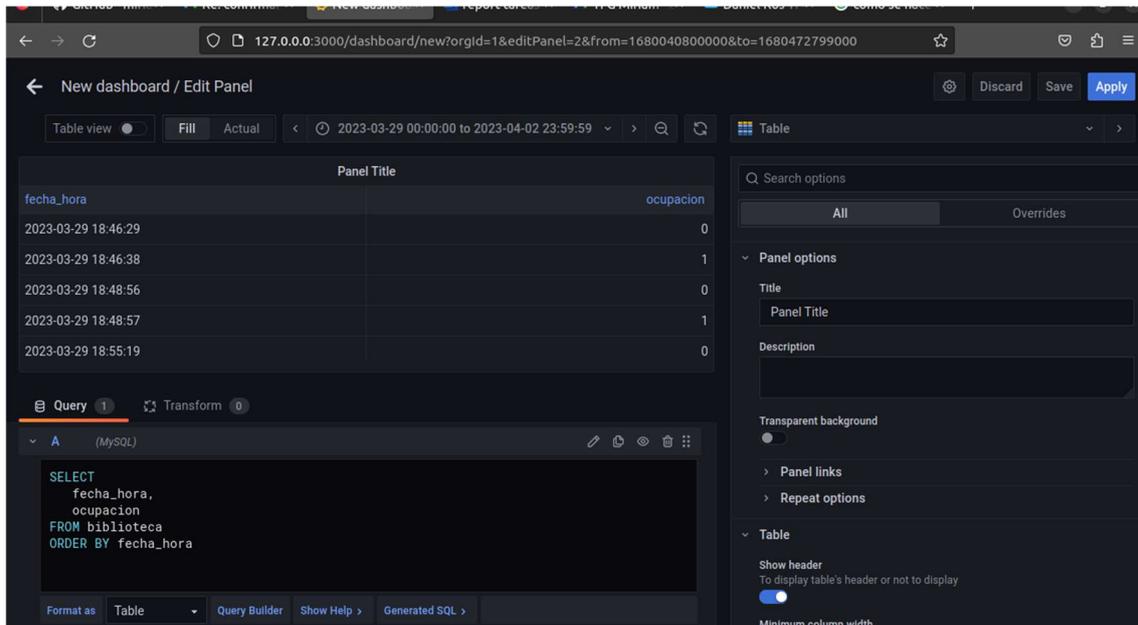


Figura 57. Configuración gráfica parte 2

Se accede a la pestaña 'Transform', se elige la opción 'Convert Field Type', se elige el campo 'fecha_hora' y se convierte en tipo time en el formato 'YYYY-MM-DD HH:mm:ss'. Y el tipo de gráfica ha de ser 'Time Series'.

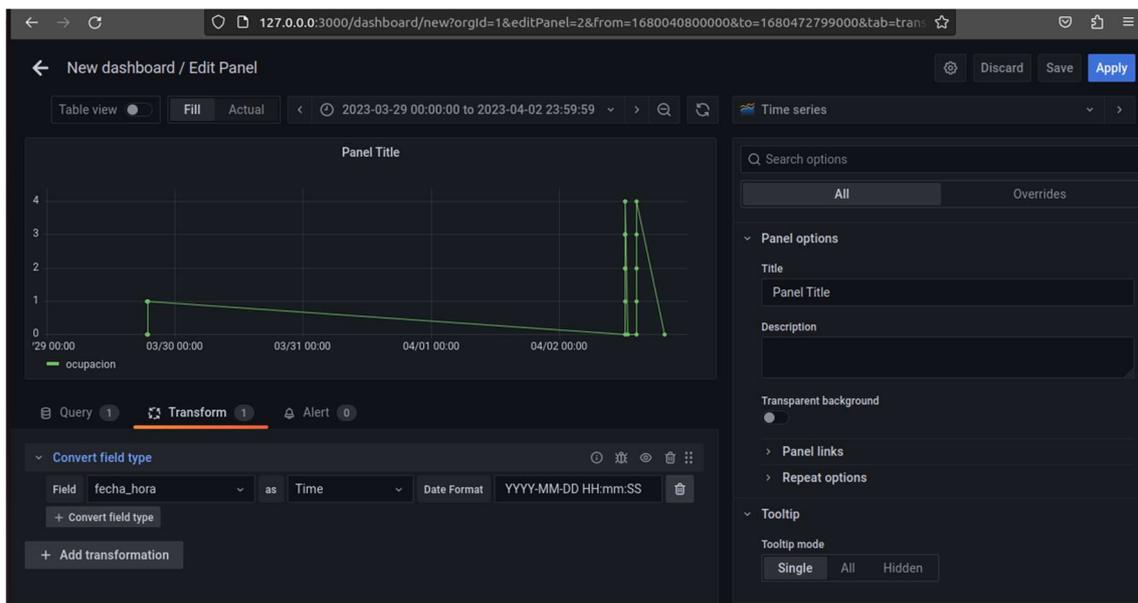


Figura 58. Configuración gráfica parte 3

En el panel se verá la gráfica que hemos configurada, donde muestra la ocupación que hay en cada fecha y hora.



Figura 59. Ejemplo grafica Grafana 25-07-2023

5.4 Servicio de monitorización mediante bot de Telegram

Para que los alumnos estén informados acerca de la ocupación y los cambios de horario que haya en la biblioteca se ha implementado un **bot** en **Telegram**. Se ha creado un canal de difusión en la plataforma de Telegram, donde se encuentra este bot que se encarga de enviar las notificaciones, a través de este canal, los estudiantes tienen la oportunidad de unirse y recibir las notificaciones. Esto garantiza a los alumnos estén informados de cualquier modificación que pueda surgir, asegurando así una experiencia optima en el uso de las instalaciones de la biblioteca.

Para que los administrados puedan monitorizar el servicio de conteo en caso de que deje de funcionar, se ha implementado un bot que envía alertas cuando la cámara no está conectada o no se reciben frames, también se encarga de informar el número de ocupación. Al igual que para el caso de alumnos, se crea un canal de difusión privado donde solo estarán los administradores y el bot.

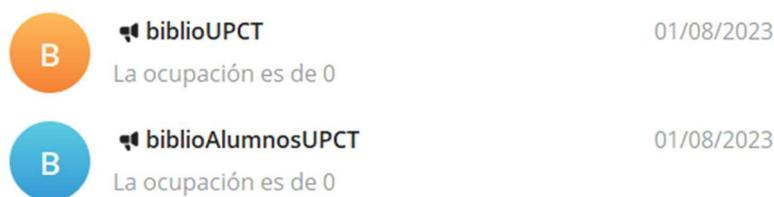


Figura 60. Bots de alumnos y administradores

En la Figura 61, se puede observar un fragmento de código que muestra el proceso de envío de notificaciones a través de bots. Para iniciar este proceso, primero se carga el bot utilizando su **token**, que se obtiene gracias a BotFather, la herramienta encargada de gestionar los bots creados. Una vez que el bot está cargado, se utiliza la función 'send_message' del bot, proporcionando el **chat_id** que corresponde al canal donde el bot se encuentra. Para obtener este chat_id, se hizo un pequeño código que toma como entrada el token del bot y el nombre del canal. Este código utiliza la función 'get_chatid' del bot para recuperar el chat_id necesario para el envío de mensajes al canal deseado. Este proceso es esencial para el funcionamiento eficiente de las notificaciones por bots.

```
async def send_telegram_message(message):  
    print("estoy en el send_message")  
    botAdmin = telegram.Bot(token="6[redacted]nqig")  
    await botAdmin.send_message(chat_id = "[redacted]79", text = message)
```

Figura 61. Función para enviar mensajes por el bot de Administradores

A continuación, se explicará cómo se han implementado las diferentes alertas:

- **Número de ocupación:** Para el número de ocupación se accede a la base de datos y se selecciona la última ocupación registrada, y se envía esta información a través de ambos bots tanto el de alumnos como administradores.



Figura 62 ejemplo canal alumnos envió de N.º ocupación

- **Horario:** El administrador tiene acceso a un enlace donde indica el nuevo horario y el bot se encarga de enviar el nuevo horario al canal de alumnos.

Introduce el mes o meses:
Septiembre-Octubre

Introduce los días:
Lunes a Viernes

Introduce el mes:
8:30-21:00

Submit

Figura 63. Formulario para notificar cambios de horario



Figura 64. Chat ejemplo envío de cambio de horario

- **Cámara On/Off:** Para informar del estado de la cámara, se ha implementado un bucle con las siguientes condiciones: Si se recibe frames y se hace ping correctamente a la cámara se envía un mensaje informando de que ya se ha conectado la cámara, no se vuelve a enviar ningún mensaje a menos que una de las condiciones anteriormente comentadas ya no se cumplan, es decir, no se reciban frames o no se haga ping, en este caso se enviará una alerta avisando que no está conectada la cámara.

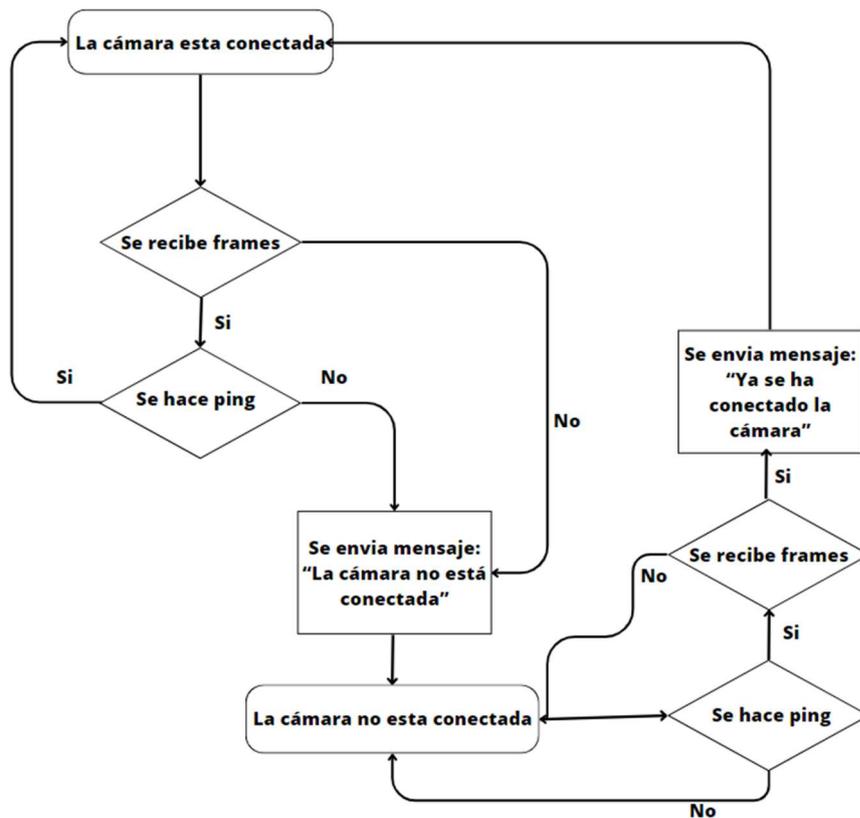


Figura 65. Diagrama de flujo envío de alertas cámara On/Off



Figura 66. Chat ejemplo envío alertas del estado de la cámara

En todas las alertas se ha establecido un intervalo de notificación cada media hora para lograr un equilibrio entre la frecuencia de las alertas y la oportunidad de mantener a los miembros de los canales debidamente informados. Así se busca proporcionar una experiencia informacional eficaz y cómoda para todos los usuarios.

5.5 Recursos computacionales requeridos

El servidor que se utiliza tiene cuatro núcleos de CPU y si un programa usa los cuatro núcleos al 100%, el uso de la CPU sería 400%. Esto no significa que el programa este consumiendo más recursos de los que tiene, sino que está aprovechando la capacidad de procesamiento paralelo del ordenador.

Durante el desarrollo de nuestras pruebas de detección y seguimiento de personas, resulta fundamental comprender en detalle los recursos computacionales que se emplean en el proceso. Para este propósito, hemos recurrido a la herramienta de monitoreo en tiempo real conocida como el comando **top**. Este análisis proporciona información esencial sobre el uso de CPU, memoria y otros recursos, lo que permite evaluar la eficiencia y el impacto del sistema en el entorno operativo.

```
File Actions Edit View Help
admincamaras@servercamaras: .../People-counting-server_bib x
top - 16:02:27 up 33 days, 2:53, 2 users, load average: 3,99, 4,50, 4,62
Tasks: 228 total, 2 running, 225 sleeping, 1 stopped, 0 zombie
%Cpu(s): 90,7 us, 6,0 sy, 0,0 ni, 3,3 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 15788,3 total, 4964,8 free, 3106,0 used, 7717,5 buff/cache
MiB Swap: 4096,0 total, 4096,0 free, 0,0 used, 12224,8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
1267542 adminca+  20   0 1381812 221496 60848 R 378,7  1,4 101:16.91 python3
91478  adminca+  20   0 2142964 633452 55308 S  5,0  3,9 856:45.63 anydesk
1428  root      20   0 737448 174016 125280 S  1,3  1,1 902:13.84 Xorg
1450  root      20   0 1743560 92664 54384 S  0,7  0,6 196:52.19 dockerd
14  root      20   0 0 0 0 I  0,3  0,0 86:38.75 rcu_sched
2338  lxd       20   0 1754428 220188 20092 S  0,3  1,4 38:05.24 mysqld
85465  adminca+  20   0 498824 107536 87336 S  0,3  0,7 0:45.68 qterminal
1267729 root      20   0 0 0 0 I  0,3  0,0 0:01.88 kworker/3:1-events
1268185 adminca+  20   0 10920 4396 3612 R 0,3 0,0 0:00.03 top
1  root      20   0 168292 13856 8396 S  0,0  0,1 1:03.01 systemd
2  root      20   0 0 0 0 S  0,0  0,0 0:01.04 kthreadd
```

Figura 67. Resultados del comando top

ATOP - servercamaras 2023/07/31 16:03:13													10s elapsed				
PRC	sys	3.66s	user	34.22s	#proc	228	#trun	5	#tslpi	753	#tslpu	62	#zombie	0	#exit	2	
CPU	sys	36%	user	343%	irq	1%	idle	21%	wait	0%	guest	0%	avgf	3.33GHz	avgscal	99%	
cpu	sys	9%	user	86%	irq	1%	idle	4%	cpu003	w 0%	guest	0%	avgf	3.33GHz	avgscal	99%	
cpu	sys	9%	user	86%	irq	0%	idle	5%	cpu002	w 0%	guest	0%	avgf	3.33GHz	avgscal	100%	
cpu	sys	10%	user	84%	irq	0%	idle	5%	cpu001	w 0%	guest	0%	avgf	3.33GHz	avgscal	100%	
cpu	sys	8%	user	86%	irq	0%	idle	6%	cpu000	w 0%	guest	0%	avgf	3.33GHz	avgscal	100%	
CPL	avg1	4.00	avg5	4.43	avg15	4.59			csw	113335	intr	25976			numcpu	4	
MEM	tot	15.4G	free	4.8G	cache	6.5G	dirty	0.1M	buff	717.8M	slab	502.4M	shrss	38.6M	numnode	1	
SWP	tot	4.0G	free	4.0G	swcac	0.0M							vmcom	7.5G	vnlim	11.7G	
PSI	cpusone	7%	mensone	0%	menfull	0%	iosone	0%	iofull	0%	cs	6/5/8	ms	0/0/0	ls	0/0/0	
LVM	g-ubuntu--lv	busy	1%	read	0	write	50	discrd	0	MBr/s	0.0	MBw/s	0.0	avio	2.32	ms	
DSK	sda	busy	1%	read	0	write	26	discrd	0	MBr/s	0.0	MBw/s	0.0	avio	4.46	ms	
NET	transport	tcp	920	tcpo	677	udpi	1	udpo	9	tcpao	0	tcppo	0	tcprs	13		
NET	network	ipi	921	ipo	667	ipfrw	0	deliv	921			icmpi	0	icmpo	0		
NET	wlx88d7	0%	pcki	473	pcko	326	sp	72	Mbps	si	271	Kbps	so	24	Kbps	erri	0
NET	enp0s25	0%	pcki	455	pcko	371	sp	100	Mbps	si	41	Kbps	so	111	Kbps	erri	0
NET	veth119	----	pcki	2	pcko	2	sp	0	Mbps	si	0	Kbps	so	0	Kbps	erri	0
NET	docker_	----	pcki	2	pcko	2	sp	0	Mbps	si	0	Kbps	so	0	Kbps	erri	0

PID	CID	SYSCPU	USRCPU	RDELAY	VGROW	RGROW	RUID	EUID	ST	EXC	THR	S	CPUNR	CPU	CMD	1/3
1267542	host	2.60s	33.19s	2.91s	0B	0B	admincam	admincam	--	-	9	R	0	360%	python3	
91478	host	0.45s	0.36s	0.58s	0B	1.3M	admincam	admincam	--	-	22	S	3	8%	anydesk	
1428	host	0.39s	0.25s	0.03s	5.0M	0B	root	root	--	-	6	S	1	6%	Xorg	
85465	host	0.02s	0.23s	0.19s	0B	0B	admincam	admincam	--	-	4	S	1	3%	qterminal	
90946	host	0.04s	0.08s	0.01s	0B	0B	root	root	--	-	4	S	3	1%	anydesk	
1268203	host	0.04s	0.03s	0.00s	5.3M	4.2M	admincam	admincam	--	-	1	R	2	1%	atop	
1020885	host	0.01s	0.02s	0.01s	0B	0B	admincam	admincam	--	-	20	S	0	0%	Isolated Web C	
91137	host	0.02s	0.01s	0.03s	0B	0B	admincam	admincam	--	-	6	S	2	0%	anydesk	
1450	host	0.01s	0.01s	0.01s	0B	0B	root	root	--	-	14	S	3	0%	dockerd	

Figura 68. Resultados del comando atop

Al analizar los resultados proporcionados por el comando top, se destaca que el proceso de detección y seguimiento de personas, ejecutado por el entorno de Python que engloba tanto el programa de conteo como el de seguimiento, genera una carga significativa en la unidad de procesamiento central (CPU). Durante las pruebas, el uso de CPU alcanza regularmente valores superiores al 300%, acercándose, pero sin sobrepasar el 380%. Esta alta utilización de la CPU es coherente con la complejidad del procesamiento de imágenes y la detección de objetos en tiempo real.

En contraste, el uso de memoria se mantiene en niveles bajos, representando aproximadamente el 1,4% del total disponible. Esto indica que el proceso de detección y seguimiento no presenta demandas significativas en términos de memoria RAM y que el sistema tiene suficiente margen en este aspecto.

5.6 Herramientas de gestión remota del servidor Zerotier y AnyDesk

Durante la ejecución de este proyecto, el servidor ha estado ubicado en el laboratorio I+D IT1, el entorno utilizado es Linux Ubuntu. Se ha tenido que trabajar en remoto y usar herramientas de acceso al servidor para su administración, actualización de los programas y las pruebas del sistema. Este servidor se puede acceder a él de manera remota a través de distintas opciones, como SSH, ZeroTier o AnyDesk.



Figura 69. Servidor

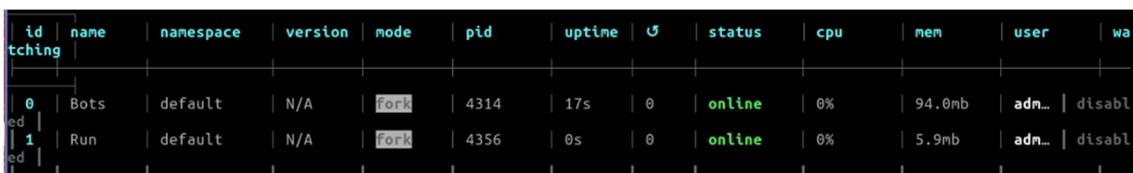
Durante todo el proceso de desarrollo y configuración del entorno, se hizo uso de ZeroTier para acceder al servidor. Sin embargo, al avanzar hacia las etapas de pruebas y validación, se tomó la decisión de cambiar a la herramienta AnyDesk. Esta decisión se tomó debido a ciertas limitaciones observadas al usar las otras opciones, específicamente en relación al concepto de reenvío de puertos (**forwarding**).

Para un mejor entendimiento de estas limitaciones, es necesario comprender que el forwarding involucra el proceso de redirigir el tráfico de datos desde un puerto local a un puerto remoto, permitiendo así la comunicación entre diferentes sistemas.

En el contexto de **ZeroTier**, esta redirección puede generar complicaciones al manejar datos visuales complejos como imágenes. Estas herramientas aplican una encapsulación y cifrado de datos, lo que puede introducir cierta sobrecarga en la transferencia de imágenes. Por ello se tomó la decisión de usar esa alternativa en las fases de prueba, ya que ofrece una experiencia más fluida, evitando así las problemáticas asociadas al reenvío de puertos.

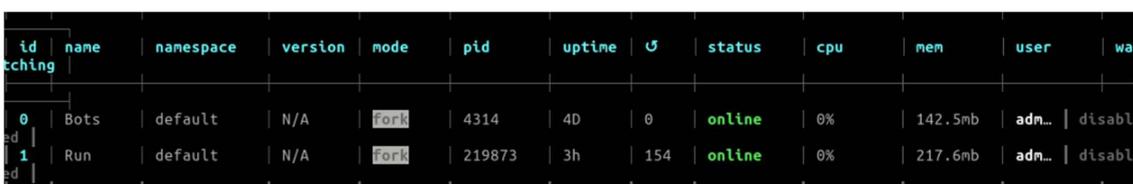
5.7 Pruebas en el servidor

Para comprobar la robustez y estabilidad del servidor se llevó a cabo una prueba de ejecución continua de los dos programas, el programa de conteo y el de envío de alertas a través de bots, durante dos semanas, utilizando la gestión de PM2. Durante este período, se observó que el servidor se mantuvo en un estado estable. En situaciones en las que la cámara se apagaba antes de lo previsto o se producía algún error, PM2 se encargaba de reiniciar el programa de manera automática hasta que funcionara correctamente. Se vio que el uso de la CPU aumentaba en proporción al flujo de personas, lo que generaba una mayor carga de trabajo para el programa de conteo. Sin embargo, a pesar de esta variabilidad, el servidor se mantuvo sólido y sin fallos.



id	name	namespace	version	mode	pid	uptime	U	status	cpu	mem	user	wa
0	Bots	default	N/A	fork	4314	17s	0	online	0%	94.0mb	adm...	disabl
1	Run	default	N/A	fork	4356	0s	0	online	0%	5.9mb	adm...	disabl

Figura 70. Lista de programas gestionados por PM2 primer día



id	name	namespace	version	mode	pid	uptime	U	status	cpu	mem	user	wa
0	Bots	default	N/A	fork	4314	4D	0	online	0%	142.5mb	adm...	disabl
1	Run	default	N/A	fork	219873	3h	154	online	0%	217.6mb	adm...	disabl

Figura 71. Lista de programas gestionados por PM2 cuarto día

Como se puede ver en la Figura 70 y 71, el sistema continuaba operando de manera estable y sin interrupciones, después de los cuatro. Estos resultados confirman la robustez y estabilidad del servidor bajo condiciones de uso intensivo y demuestran la efectividad de la gestión de PM2 en el mantenimiento del sistema en funcionamiento continuo.

Durante el transcurso de estas semanas de prueba se pudo ver que en Grafana se actualizaba el número de ocupación que había y también se recibirán las notificaciones con la ocupación y las alertas en caso de que la cámara dejase de funcionar.

A lo largo de estas semanas de prueba, se observó que en Grafana se actualizaba constantemente el número de ocupación actual, además de recibir notificaciones sobre la ocupación y alertas en caso de que la cámara dejara de funcionar.



Figura 72. Ejemplo Grafana 28-07-2023



Figura 73. Notificaciones de ocupación 28-07-2023



Figura 74. Notificación cámara no conectada 28-07-2023

Capítulo 6. Conclusiones y trabajos futuros

6.1 Conclusiones

Durante el desarrollo de este proyecto se ha analizado la problemática del control de aforo en espacios de interior, con un enfoque personalizado y de bajo coste diseñado específicamente para la Biblioteca de Antigonos de la Universidad Politécnica de Cartagena (UPCT). Los resultados técnicos obtenidos indican claramente que la implementación de este servicio es viable y podría ser gestionado eficazmente por el Servicio CRAI en colaboración con la Unidad Técnica y el Servicio de Informática de la UPCT. Es fundamental destacar que esta solución no almacena ni registra ningún tipo de imagen de personas ni recopila datos personales, cumpliendo así con la Ley Orgánica de Protección de Datos (LOPD). La única consideración importante radica en la correcta gestión y mantenimiento del servidor por parte del personal técnico capacitado, siguiendo los procedimientos establecidos en la UPCT.

El principal objetivo de este proyecto consistió en automatizar el proceso de conteo, sustituyendo así el trabajo manual que previamente llevaban a cabo el personal de Biblioteca. Para lograrlo, se ha tenido que hacer un estudio y comprensión de tecnologías de visión artificial y las últimas tendencias en soluciones neuronales de aprendizaje profundo. Esta parte ha sido bastante compleja y laboriosa ya que, los campos de sistemas de visión artificial, inteligencia artificial y redes neuronales de aprendizaje profundo, no se estudian en el Grado de Ingeniería Telemática.

Durante el análisis de soluciones de software libre se escogió y estudio los modelos CNN de detección basados en Yolo3 y MobileNet SSD y el sistema de seguimiento que se tuvo como punto de partida. Se ha escogido y justificado el uso de la solución MobileNet SSD debido a su eficiencia en términos de recursos computacionales y por ser apta para una solución de bajo coste. A su vez, se han realizado pruebas exhaustivas de la influencia de la colocación y orientación de la cámara con respecto a la zona de paso para maximizar la tasa de detección de personas y que las imágenes capturadas se asemejen más a las imágenes patrón de personas del dataset de entrenamiento COCO. Se han encontrado recomendaciones y pautas para instalar la cámara en otras bibliotecas de la UPCT.

Respecto al módulo de conteo, basado en MobileNet SSD y el algoritmo de tracking, se adaptó y se configuró este código para que funcionase en el entorno del servidor, incorporando funcionalidades adicionales de valor, como la capacidad de registrar la ocupación en una base de datos centralizada y presentar visualmente estos datos en Grafana. Además, implementamos un sistema de alertas mediante bots de Telegram para una comunicación efectiva.

La realización de este proyecto representó una experiencia enriquecedora desde una perspectiva tanto académica como profesional. Durante su desarrollo, se adquirieron conocimientos sobre diversas tecnologías y conceptos, que abarcaron desde la programación en Python, o el aprendizaje sobre diferentes modelos de detección.

La exploración del campo de la visión artificial y la implementación de sistemas de conteo de personas resultaron particularmente interesantes. Esta experiencia proporcionó una comprensión de un ámbito tecnológico en constante evolución y fortaleció la apreciación de la aplicabilidad práctica de estas tecnologías en entornos del mundo real.

Este proyecto ha representado una oportunidad única de crecimiento y desarrollo tanto a nivel personal como profesional. Ha resaltado la importancia de la tecnología en entornos educativos

y públicos, subrayando la necesidad constante de proyectos similares que impulsen la excelencia y la eficiencia en nuestras instituciones y comunidades.

6.2 Grado de consecución de los objetivos y formación adquirida

Resumen de objetivos propuestos y alcanzados

Los objetivos alcanzados son los siguientes:

1. Se ha estudiado y adaptado un módulo de conteo de personas, basado en visión artificial con redes neuronales de aprendizaje profundo, a medida para una biblioteca.
2. Diseño y configuración de un backend con una BBDD para el almacenamiento de los datos generados por el sistema de conteo.
3. Desarrollo de un frontend, configuración de Grafana para mostrar los datos de ocupación en tiempo real, y la creación de bots de Telegram para enviar alertas a un administrador o informar a alumnos de la ocupación de la biblioteca.
4. Se ha realizado un prototipo real de bajo coste a medida para las características de una Biblioteca de una universidad.

Todo el proyecto está documentado técnicamente para reproducir la solución. Por los resultados obtenidos, se considera que la solución se puede pasar a producción y extender a las 3 Bibliotecas que tiene la UPCT.

Aprendizaje y dificultades

En el transcurso de este proyecto, se ha tenido que adquirir un conjunto de nuevas habilidades y conocimientos. En primer lugar, se ha explorado el lenguaje de programación Python, el cual era un terreno desconocido hasta el inicio de esta iniciativa. Asimismo, se ha profundizado en la operación de entornos Linux, el cual representaba un sistema operativo poco familiar previamente. También, se ha experimentado con Docker, permitiendo una introducción inicial a esta tecnología.

Además, se ha dedicado tiempo a la revisión y comprensión de documentos relevantes relacionados con la visión artificial, OpenCV y diversos modelos de detección, así como a su funcionamiento. También se ha adentrado en la comprensión de sistemas de seguimiento (tracking).

Uno de los desafíos más significativos a lo largo de este proyecto fue la comprensión y la implementación de las soluciones YOLO3 y MobileNet, así como la capacidad para ajustar sus parámetros, dado que no se tenía experiencia previa en el campo de la visión artificial y los modelos de detección. Adaptar dicho programa al entorno específico de la biblioteca también resultó ser una tarea muy exigente.

6.3 Trabajos futuros

En el proceso de desarrollo y puesta en marcha del sistema de conteo de personas en la Biblioteca Antigones de la UPCT, se han alcanzado buenos resultados y logros. Sin embargo, como en cualquier proyecto tecnológico, existen oportunidades para la mejora continua y la expansión del sistema. A continuación, se comentan posibles vías para futuros trabajos y desarrollos que podrían enriquecer y perfeccionar aún más este proyecto:

1. **Evaluación de otros modelos de detección e investigación en técnicas de Transfer Learning.** Se trataría de ver cómo reentrenar los modelos de detección pero con

imágenes reales etiquetas de una Biblioteca. Este aspecto se trató someramente durante el TFG pero debido a la alta complejidad y dificultad se propone como posible trabajo futuro.

2. **Pruebas en otras bibliotecas de la UPCT:** Se proponer probar en las bibliotecas del Campus de Alfonso XIII y el CIM. Lo principal sería confirmar el estudio de la ubicación y orientación e instalación de la cámara.
3. **En el servidor, configuración de franja horaria dinámica:** Una mejora importante podría ser permitir que los administradores configuren la franja horaria de funcionamiento del sistema de manera dinámica a través de un formulario. Esto sería útil en situaciones de festivos, imprevistos u horarios especiales, lo que garantizaría una adaptación más precisa a las necesidades cambiantes.
4. **En el servidor, hacer un API REST de acceso a los datos de aforo en tiempo real:** con este API REST, el servicio de Informática podría integrar el dato del aforo en la web de la UPCT o en el appUPCT Android de alumnos.

Bibliografía y referencias

- [1] «Protocol Design Whitepaper | ZeroTier Documentation». <https://docs.zerotier.com/zerotier/manual/> (accedido 31 de agosto de 2023).
- [2] «GitHub: Let's build from here · GitHub». <https://github.com/> (accedido 31 de agosto de 2023).
- [3] «Visual Studio Code - Code Editing. Redefined». <https://code.visualstudio.com/> (accedido 31 de agosto de 2023).
- [4] «Descargar Visual Studio Tools: instalación gratuita para Windows, Mac, Linux». <https://visualstudio.microsoft.com/es/downloads/> (accedido 31 de agosto de 2023).
- [5] «Welcome to Python.org». <https://www.python.org/> (accedido 31 de agosto de 2023).
- [6] «El tutorial de Python — documentación de Python - 3.11.5». <https://docs.python.org/es/3/tutorial/index.html> (accedido 31 de agosto de 2023).
- [7] «Docker: Accelerated Container Application Development». <https://www.docker.com/> (accedido 31 de agosto de 2023).
- [8] «Grafana: The open observability platform | Grafana Labs». <https://grafana.com/> (accedido 31 de agosto de 2023).
- [9] «Qué es Grafana, cómo podemos emplearlo en la monitorización». <https://pandorafms.com/blog/es/que-es-grafana/> (accedido 31 de agosto de 2023).
- [10] «¿Qué Es MySQL? Explicación Detallada Para Principiantes». <https://www.hostinger.es/tutoriales/que-es-mysql> (accedido 31 de agosto de 2023).
- [11] «Qué es NodeJS y para qué sirve | OpenWebinars». <https://openwebinars.net/blog/que-es-nodejs/> (accedido 31 de agosto de 2023).
- [12] «pm2 - npm». <https://www.npmjs.com/package/pm2> (accedido 31 de agosto de 2023).
- [13] «HTML: Lenguaje de etiquetas de hipertexto | MDN». <https://developer.mozilla.org/es/docs/Web/HTML> (accedido 31 de agosto de 2023).
- [14] «CSS básico - Aprende desarrollo web | MDN». https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics (accedido 31 de agosto de 2023).
- [15] «Telegram Messenger». <https://telegram.org/> (accedido 31 de agosto de 2023).
- [16] «Visión por computadora ▷ Información, Historia, Biografía y más.». <https://es.wikidat.com/info/vision-por-computadora> (accedido 29 de agosto de 2023).
- [17] «¿Qué es la visión por computadora? Sistemas de visión artificial y aplicaciones: ejemplos de soluciones — RecFaces». <https://recfaces.com/es/articles/vision-computador-soluciones> (accedido 29 de agosto de 2023).
- [18] «Detección de Bordes». <https://view.genial.ly/62c4c7ccd27e63001ace2636/interactive-content-deteccion-de-bordes> (accedido 28 de agosto de 2023).
- [19] L. Gervasoni *et al.*, «Mecánica Computacional Vol XXXIII», 2014.

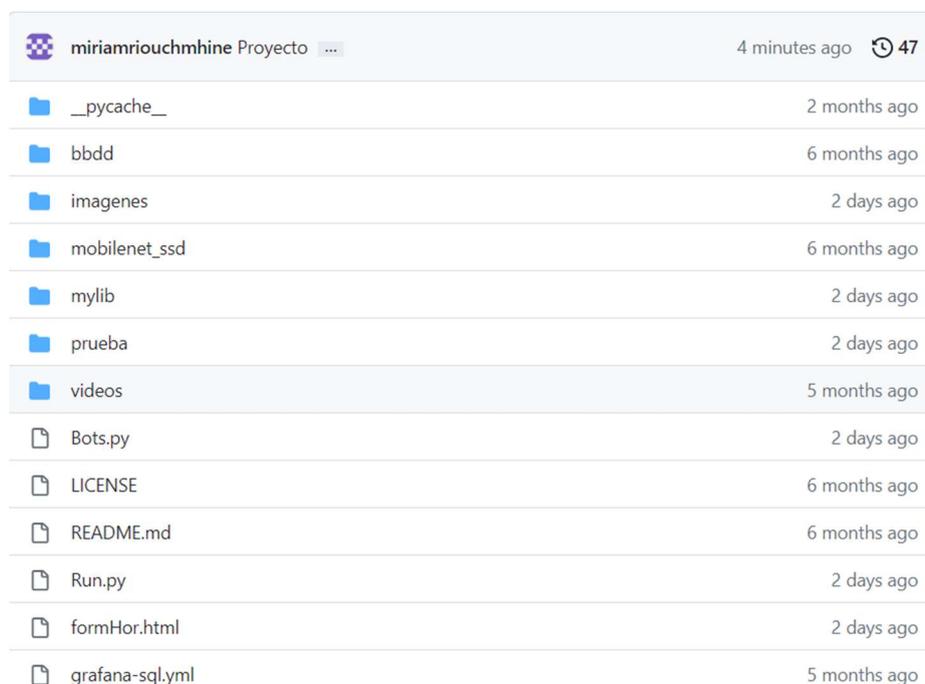
- [20] «Sistemas de conteo de personas: Software y tecnologías existentes en 2023». <https://www.pro.affluences.com/post/sistemas-conteo-personas-tecnolog%C3%ADas?lang=es> (accedido 29 de agosto de 2023).
- [21] «CNN vs.RNN: ¿En qué se diferencian?» <https://kryptonsolid.com/cnn-vs-rnn-en-que-se-diferencian/> (accedido 29 de agosto de 2023).
- [22] «TC70 | Cámara de Vigilancia Wi-Fi inteligente | Tapo». <https://www.tapo.com/es/product/smart-camera/tc70/> (accedido 24 de agosto de 2023).
- [23] «¿Cómo puedo reproducir una transmisión en vivo RTSP de mi cámara Tapo? | Tapo». <https://www.tapo.com/es/faq/34/> (accedido 25 de agosto de 2023).
- [24] «COCO - Common Objects in Context». <https://cocodataset.org/#home> (accedido 1 de septiembre de 2023).
- [25] «YOLO Algorithm for Object Detection Explained [+Examples]». <https://www.v7labs.com/blog/yolo-object-detection> (accedido 1 de septiembre de 2023).
- [26] J. Redmon, S. Divvala, R. Girshick, y A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection», jun. 2015, [En línea]. Disponible en: <http://arxiv.org/abs/1506.02640>
- [27] «Detección de objetos usando YoloV3 y OpenCV». <https://ichi.pro/es/deteccion-de-objetos-usando-yolov3-y-opencv-28510119961632> (accedido 1 de septiembre de 2023).
- [28] «Principales proyectos de detección de objetos YOLO de código abierto». <https://ichi.pro/es/principales-proyectos-de-deteccion-de-objetos-yolo-de-codigo-abierto-227138885883936> (accedido 1 de septiembre de 2023).
- [29] A. G. Howard *et al.*, «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications», abr. 2017, [En línea]. Disponible en: <http://arxiv.org/abs/1704.04861>
- [30] «Comprender las circunvoluciones separables en profundidad y la eficiencia de MobileNets». <https://ichi.pro/es/comprender-las-circunvoluciones-separables-en-profundidad-y-la-eficiencia-de-mobilenets-120825327265027> (accedido 1 de septiembre de 2023).
- [31] W. Liu *et al.*, «SSD: Single Shot MultiBox Detector», dic. 2015, doi: 10.1007/978-3-319-46448-0_2.
- [32] L. Moreno Villalba, A. Meléndez Ramírez, J. Avila, J. Manuel, y S. Carrillo, «CLASIFICACIÓN DE OBSTACULOS A TRAVÉS DEL ALGORITMO SDD», 2017. [En línea]. Disponible en: <https://www.researchgate.net/publication/336038270>
- [33] «Object Detection : A Comparison of performance of Deep learning Models on Edge Using Intel Movidius Neural Compute Stick and Raspberry PI3 | by Tejas Narayan | Intel Student Ambassadors | Medium». <https://medium.com/intel-student-ambassadors/object-detection-a-comparison-of-performance-of-deep-learning-models-on-edge-using-intel-f66eb7f45b17> (accedido 1 de septiembre de 2023).

- [34] «Simple object tracking with OpenCV - PyImageSearch». <https://pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/> (accedido 4 de septiembre de 2023).
- [35] F. Wang, «I-Introduction RTSP Live Streaming».
- [36] «GitHub - saimj7/People-Counting-in-Real-Time: People Counting in Real-Time with an IP camera.» <https://github.com/saimj7/People-Counting-in-Real-Time> (accedido 1 de septiembre de 2023).
- [37] «Simple object tracking with OpenCV - PyImageSearch». <https://pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/> (accedido 1 de septiembre de 2023).
- [38] «¿Qué es Grafana?» <https://www.redhat.com/es/topics/data-services/what-is-grafana> (accedido 29 de agosto de 2023).

Anexos

Anexo 1. Estructura y organización del código

El código del servidor está subido en GitHub: https://github.com/miriamriouchmhine/People-counting-server_bib



Nombre del archivo/directorio	Fecha de modificación
miriamriouchmhine Proyecto ...	4 minutes ago 47
__pycache__	2 months ago
bbdd	6 months ago
imagenes	2 days ago
mobilenet_ssd	6 months ago
mylib	2 days ago
prueba	2 days ago
videos	5 months ago
Bots.py	2 days ago
LICENSE	6 months ago
README.md	6 months ago
Run.py	2 days ago
formHor.html	2 days ago
grafana-sql.yml	5 months ago

Figura 75. Proyecto en GitHub

Anexo 2. Librerías utilizadas

OpenCV : es una librería de código abierto utilizada para el procesamiento de imágenes y videos. Proporciona herramientas para realizar operaciones como la detección de bordes, la eliminación de ruido, la detección de objetos, entre otras.

“opencv-python”: se utiliza para procesamiento de imágenes y visión por computadora.

“mylib.centroidtracker” y **“mylib.trackableobject”**: son dos clases personalizadas que son utilizadas para rastrear los objetos de un video.

“imutils”: es una colección de funciones y utilidades de procesamiento de imágenes que simplifica muchas de las tareas comunes realizadas al trabajar con imágenes en Python, en particular, cuando se utiliza la biblioteca OpenCV

“imutils.video.VideoStream” y **“imutils.video.FPS”**: estas clases se utilizan para leer el video en tiempo real y calcular la velocidad de frames por segundo del video.

“mylib”, **“config”**: son módulos personalizados que se utilizan para almacenar la configuración.

“numpy”: se emplea en el código para llevar a cabo cálculos numéricos, específicamente para el registro de los recuentos de personas que atraviesan el sistema.

“dlib”: Su principal enfoque es proporcionar herramientas para la detección y reconocimiento de objetos en imágenes y videos.

“schedule”: se utiliza para programar eventos que se ejecuten en un momento determinado.

“time”: es una biblioteca que ofrece utilidades para el manejo de tiempo y fechas. Facilita la medición del tiempo de ejecución de un programa y proporciona funciones para gestionar marcas de tiempo y fechas.

“datetime”: es una librería que proporciona herramientas para trabajar con fechas y horas. Permite manipular, comparar y formatear fechas y horas de una manera sencilla.