



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS
DE LA COMPUTACIÓN E INNOVACION
TECNOLÓGICA**

CARRERA DE SOFTWARE

TITULO

**SOFTWARE DE MONITOREO Y CONTROL DE FLUJO DE
PERSONAS EN TIEMPO REAL CON IA Y VISIÓN
COMPUTACIONAL.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SOFTWARE**

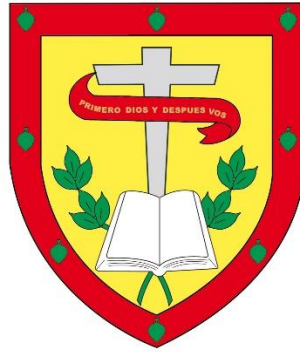
AUTOR: ALFREDO GERARDO LOAIZA BARRIOS

DIRECTOR: Ing. SEGUNDO ISABEL SAÑAY SAÑAY

CUENCA - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACION TECNOLÓGICA**

CARRERA DE SOFTWARE

TITULO

SOFTWARE DE MONITOREO Y CONTROL DE FLUJO DE PERSONAS EN
TIEMPO REAL CON IA Y VISIÓN COMPUTACIONAL.

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SOFTWARE**

AUTOR: ALFREDO GERARDO LOAIZA BARRIOS

DIRECTOR: Ing. SEGUNDO ISRAEL SAÑAY SAÑAY

CUENCA - ECUADOR

2024

DIOS, PATRIA, CULTURA Y DESARROLLO



Declaratoria de Autoría y Responsabilidad

Alfredo Gerardo Loaiza Barrios portador(a) de la cédula de ciudadanía N° 1104475346. Declaro ser el autor de la obra: **"Software de monitoreo y control de flujo de personas en tiempo real con IA y Visión computacional."**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 21 de octubre de 2024

F: 

Alfredo Gerardo Loaiza Barrios

C.I. 1104475346

CERTIFICADO

Certifico que el presente Trabajo de investigación fue desarrollado por **Alfredo Gerardo Loaiza Barrios** con el tema **“Software de monitoreo y control de flujo de personas en tiempo real con IA y Visión computacional.”**, bajo mi supervisión.

Atentamente,

 Firmado digitalmente por:
SEGUNDO ISRAEL SAÑAY
SAÑAY

Ing. Segundo Isael Sañay Sañay

Dedicatoria

Este trabajo de investigación está dedicado a la memoria de mis amados abuelitos, cuyas enseñanzas y amor incondicional han sido una fuente constante de inspiración en mi vida. Su ejemplo de resiliencia me ha orientado en cada uno de mis pasos.

A mis padres, quienes han sido mi pilar fundamental. Su amor, paciencia y confianza en mí me han brindado la fortaleza necesaria para enfrentar cualquier desafío. Estoy profundamente agradecido por sus sacrificios y por enseñarme el valor del esfuerzo y la perseverancia.

A mi hermana, por su apoyo incondicional y su amistad. Gracias por estar siempre a mi lado, por tus palabras motivadoras y por compartir conmigo tantos momentos.

Agradecimiento

En primer lugar, quiero expresar mi sincero agradecimiento a mis padres y hermana, por su dedicación, amor y apoyo incondicional en todo momento. Su fe en mí y sus sacrificios constantes me han impulsado a esforzarme y perseverar en la consecución de mis metas. Gracias por ser mi fuente de inspiración y mi refugio.

A mi novia, por su amor y apoyo incondicional. Su aliento constante y su comprensión me han dado la fuerza para seguir adelante. Gracias por estar a mi lado en cada paso de este viaje.

A mi tutor, cuya guía y apoyo han sido fundamentales en cada etapa de este trabajo. Su experiencia y consejos me han orientado y motivado a alcanzar mis objetivos.

Finalmente, quiero agradecer a la empresa que me brindó la oportunidad de desarrollar este proyecto. Su acogida y apoyo han sido clave para la realización de este trabajo, y estoy profundamente agradecido por la confianza depositada en mí y por el ambiente de colaboración que me ofrecieron.

RESUMEN

Se presenta la estructura de desarrollo de software de “monitoreo y control en tiempo real del flujo de personas, utilizando inteligencia artificial y visión computacional”. Diseñado para una empresa de muebles, este sistema mejora la eficiencia, seguridad y optimización de los espacios comerciales, superando el conteo manual. El proyecto emplea el modelo FURPS+ para evaluar exhaustivamente los requisitos (Araujo Sandoval, 2020; Yungan Gualli et al., 2019) y sigue el Proceso Unificado para un desarrollo iterativo (Sumano, 2012; Hernández González, 2017). Tecnologías como Python, OpenCV, PyTorch y Laravel son esenciales en su implementación (Itseez, 2018; Paszke et al., 2019; Otwell, 2020). Los Resultados del Proyecto muestra una mínima diferencia con el conteo manual, con un margen de error de entre 0.97% y 22.39%. Respalda cada detección con imágenes, y segmenta por edad y género, proporcionando datos valiosos para la toma de decisiones estratégicas.

Palabras clave: Inteligencia Artificial, Visión Computacional, FURPS+, Proceso Unificado, Monitoreo y Control de Flujo de Personas en Tiempo Real.

ABSTRACT

This document presents the development structure of a "real-time monitoring and control system for people flow using artificial intelligence and computer vision." Designed for a furniture company, this system enhances the efficiency, safety, and optimization of commercial spaces, surpassing manual counting methods. The project employs the FURPS+ model to thoroughly evaluate the requirements (Araujo Sandoval, 2020; Yungan Gualli et al., 2019) and follows the Unified Process for iterative development (Sumano, 2012; Hernández González, 2017). Technologies such as Python, OpenCV, PyTorch, and Laravel are essential for its implementation (Itseez, 2018; Paszke et al., 2019; Otwell, 2020). The project results show a minimal difference compared to manual counting, with an error margin between 0.97% and 22.39%. It supports each detection with images and segments by age and gender, providing valuable data for strategic decision-making.

Keywords: Artificial Intelligence, Computer Vision, FURPS+, Unified Process, Real-Time People Flow Monitoring and Control.

**Software de monitoreo y control de
flujo de personas en tiempo real con IA
y Visión computacional.**

**Real-Time People Flow Monitoring And Control
Software With AI And Computer Vision**

INTRODUCCIÓN

El monitoreo en tiempo real del flujo de personas es crucial para organizaciones de distintos niveles, como las dedicadas a la fabricación y comercialización de muebles. Los métodos tradicionales de conteo manual son ineficientes debido a su alto costo, baja precisión y lentitud (Araujo Sandoval, 2020). Ante estos retos, surge la implementación de un sistema basado en inteligencia artificial (IA) y visión computacional para mejorar la eficiencia y seguridad de los espacios comerciales (Gómez Flores, 2020; Redmon y Farhadi, 2018).

El proyecto busca desarrollar un sistema automatizado que permita detectar, rastrear y contar personas en tiempo real mediante video, superando las limitaciones de los métodos manuales y mejorando la precisión y escalabilidad. Los objetivos específicos incluyen analizar el estado actual de los sistemas de video, desarrollar un software basado en redes neuronales profundas, implementarlo en un entorno real y validar su efectividad.

El método de conteo manual presenta problemas como falta de precisión, altos costos operativos y tiempos de respuesta lentos, lo que lo hace inviable para múltiples ubicaciones. En contraste, el sistema automatizado propuesto utiliza visión por computadora y redes neuronales para mejorar la precisión del conteo de personas, segmentándolas por género y edad. Este sistema reduce costos, proporciona datos en tiempo real y es escalable, permitiendo su implementación en múltiples sitios sin complicaciones adicionales.

La automatización del proceso mediante IA y visión computacional optimiza la recolección de datos, mejora la gestión de multitudes y contribuye a la seguridad y eficiencia en espacios comerciales. Esta tecnología promete transformar los procesos operativos, brindando soluciones más precisas, eficientes y escalables para la toma de decisiones estratégicas en la empresa.

MARCO TEÓRICO

PROCESO UNIFICADO (PU) de desarrollo

El Proceso Unificado (PU) es un marco de trabajo para el desarrollo de software basado en iteraciones, lo que permite ajustes continuos y mejora la calidad del producto final. Según Sumano (2012), este enfoque reduce los riesgos asociados con cambios en los requisitos, facilita la identificación temprana de problemas y asegura que el desarrollo se mantenga alineado con las expectativas del cliente.

Una de las principales ventajas del PU, como indica Hernández González (2017), es su capacidad para gestionar la complejidad y el cambio mediante iteraciones cortas y manejables. Esto permite realizar evaluaciones periódicas y ajustes necesarios,

mejorando la calidad del producto final y aumentando la satisfacción tanto del cliente como del equipo de desarrollo.

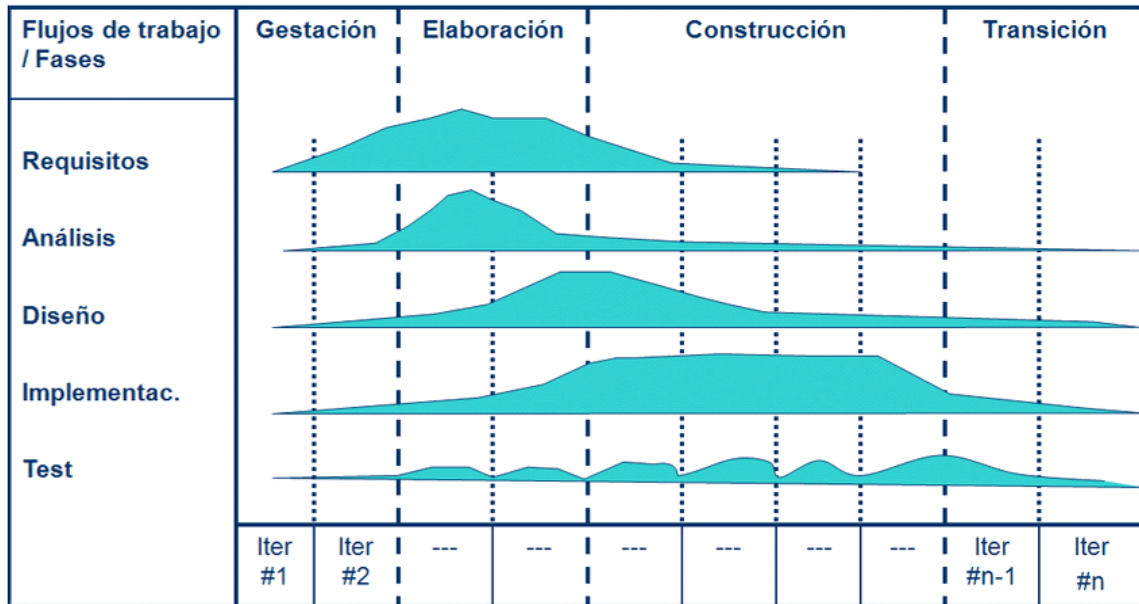


ILUSTRACIÓN 1. Santana Núñez, J. M. (2015). Flujos de trabajo del Proceso Unificado: La gráfica proporciona una representación visual detallada de cómo las actividades clave del Proceso Unificado (PU)—incluyendo Requisitos, Análisis, Diseño, Implementación y Pruebas—se desarrollan a lo largo de sus distintas fases.

Como se observa en la Ilustración 1, el PU se estructura en cuatro fases: Iniciación, Elaboración, Construcción y Transición. En la fase de Gestación, se define el alcance del proyecto, se identifican los principales riesgos y se elabora una visión general del sistema a desarrollar. Sumano (2012) menciona que esta fase es fundamental para establecer una base sólida, asegurando que todos los involucrados tengan una comprensión clara de los objetivos y riesgos del proyecto.

La fase de Elaboración implica desarrollar una arquitectura base y refinar el plan del proyecto, enfocándose en mitigar los riesgos más críticos identificados previamente (Hernández González, 2017). Durante la fase de Construcción, se desarrolla el sistema completo a través de múltiples iteraciones, añadiendo funcionalidad de manera incremental y realizando pruebas para garantizar que cada componente funcione correctamente.

Finalmente, la fase de Transición se centra en la entrega del sistema al usuario final, incluyendo actividades como la instalación, el soporte inicial y la capacitación de los usuarios (Sumano, 2012; Hernández González, 2017).

FURPS+

El modelo FURPS+ es una extensión del modelo original FURPS, utilizado para clasificar y evaluar los requisitos de software. Como se muestra en la Ilustración 2, cada letra del acrónimo representa una dimensión específica de la calidad: Funcionalidad, Usabilidad, Fiabilidad, Rendimiento y Soporte. La extensión "+" agrega consideraciones adicionales sobre mantenibilidad, portabilidad, reusabilidad y otros factores. Este modelo es

ampliamente utilizado debido a su capacidad para proporcionar una evaluación detallada y estructurada de los distintos aspectos de un software, permitiendo a los desarrolladores y gestores de proyectos identificar áreas críticas que necesitan atención (Yungan Gualli et al., 2019).

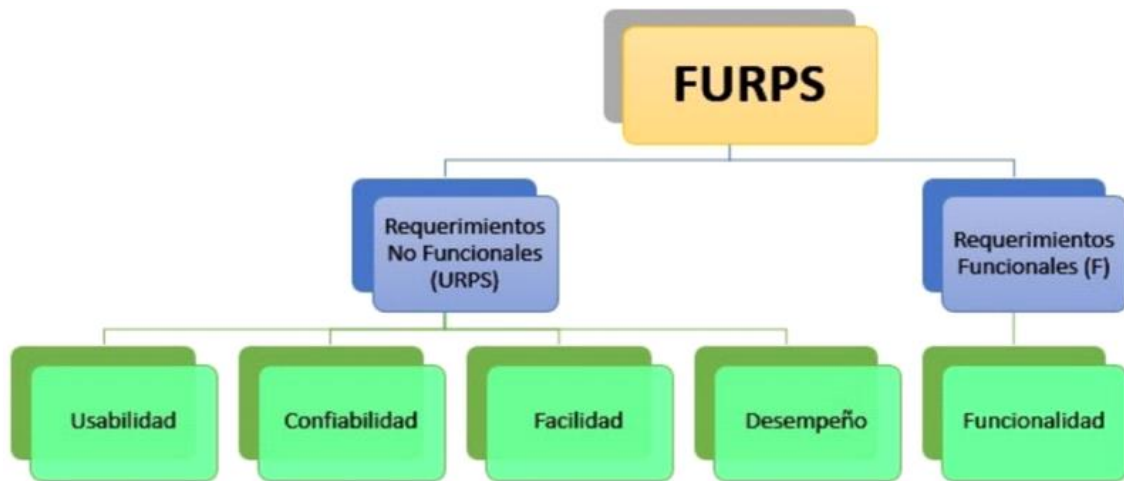


ILUSTRACIÓN 2. López Meneses, D. A. Modelos FURPS: El diagrama ilustra el modelo FURPS, el cual se utiliza para clasificar y organizar los requisitos del software en dos categorías principales: Funcionales (F) y No Funcionales (URPS).

La implementación del modelo FURPS+ en la gestión de proyectos de software proporciona una guía clara para asegurar la calidad del producto. Permite identificar áreas de mejora y garantiza que el software cumpla con los estándares necesarios para satisfacer las expectativas del cliente (Araujo Sandoval, 2020).

- **Funcionalidad** se refiere a la capacidad del software para cumplir con las tareas y requisitos para los cuales fue diseñado, incluyendo aspectos como la exactitud, la interoperabilidad y la seguridad.
- **Usabilidad** se centra en la facilidad con la que los usuarios pueden aprender a usar el software y alcanzar sus objetivos, evaluando la simplicidad de la interfaz, la documentación y el soporte técnico.
- **Fiabilidad** mide la capacidad del software para funcionar de manera consistente y sin fallos, considerando la tasa de fallos, la capacidad de recuperación y la disponibilidad.
- **Rendimiento** evalúa la eficiencia con la que el software utiliza los recursos del sistema y su capacidad para manejar la carga de trabajo, midiendo el tiempo de respuesta, el uso de la memoria y la capacidad de procesamiento (Yungan Gualli et al., 2019).
- **Soporte** se refiere a la facilidad con la que el software puede ser mantenido, actualizado y mejorado, incluyendo la documentación, la asistencia técnica y la disponibilidad de actualizaciones (Araujo Sandoval, 2020).

Además de estas cinco dimensiones principales, la extensión "+" del modelo FURPS incluye otros factores críticos para la calidad del software, como la mantenibilidad, la

portabilidad y la seguridad, lo que permite una evaluación más completa y detallada del software (Yungan Gualli et al., 2019).

INTELIGENCIA ARTIFICIAL (IA)

La Inteligencia Artificial (IA) se refiere a la capacidad de las máquinas para imitar funciones cognitivas humanas, como el aprendizaje y la toma de decisiones. Según McCarthy (2021), la IA es crucial en el desarrollo de aplicaciones para analizar datos en tiempo real y tomar decisiones basadas en patrones detectados. Los algoritmos de aprendizaje automático, especialmente el aprendizaje profundo (deep learning), permiten procesar grandes volúmenes de datos y extraer información relevante. La IA se aplica en diversos campos, como la salud para diagnósticos médicos, en la industria para optimización de procesos, y en seguridad para vigilancia y detección de fraudes. Además, la IA mejora la interacción usuario-software a través de chatbots y asistentes virtuales, y optimiza la gestión de procesos de negocio (Cordero Guzmán et al., 2020).

REDES NEURONALES CONVOLUCIONALES (CNN)

Las Redes Neuronales Convolucionales (CNN) son redes profundas eficientes en el procesamiento de datos con estructura en cuadrícula, como imágenes. Gómez Flores (2020) destaca que las CNN aprenden características jerárquicas de las imágenes, facilitando la detección de patrones y mejoras en áreas como el reconocimiento facial y la segmentación de imágenes. La Ilustración 3 muestra cómo las CNN utilizan capas de convolución, pooling y totalmente conectadas para procesar y clasificar imágenes, permitiendo un aprendizaje automático sin necesidad de extracción manual de características.

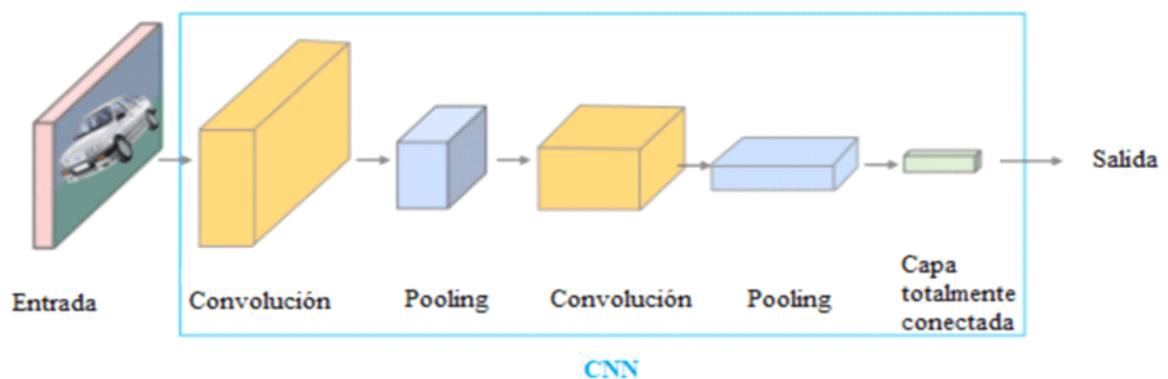


ILUSTRACIÓN 3. Diagrama del funcionamiento de una red neuronal convolucional (CNN) Andrade, H. (2021). Esta ilustración proporciona una visión detallada del flujo de datos a través de una red neuronal convolucional, destacando las distintas etapas [entrada, convolución, pooling y capas completamente conectadas hasta la salida] que conforman el proceso de análisis de imágenes.

YOLO (You Only Look Once)

Dentro de las aplicaciones de Redes Neuronales Convolucionales (CNN), YOLO (You Only Look Once) se destaca por su capacidad para la detección de objetos en tiempo real.

Según Redmon y Farhadi (2018), YOLO logra un equilibrio entre precisión y velocidad, lo que lo hace ideal para aplicaciones que requieren procesamiento rápido sin sacrificar la exactitud.

YOLO divide una imagen en una cuadrícula y predice simultáneamente las cajas delimitadoras y las probabilidades de clase para cada celda de la cuadrícula. Esta metodología permite procesar imágenes más rápidamente que otros enfoques que analizan regiones de interés por separado. Además, YOLO es eficaz para detectar objetos a diferentes escalas y es robusto en la detección de objetos pequeños y grandes en la misma imagen.

Gracias a su rendimiento en tiempo real, YOLO es ideal para diversas aplicaciones, como sistemas de vigilancia, conducción autónoma y aplicaciones móviles.

VISIÓN COMPUTACIONAL

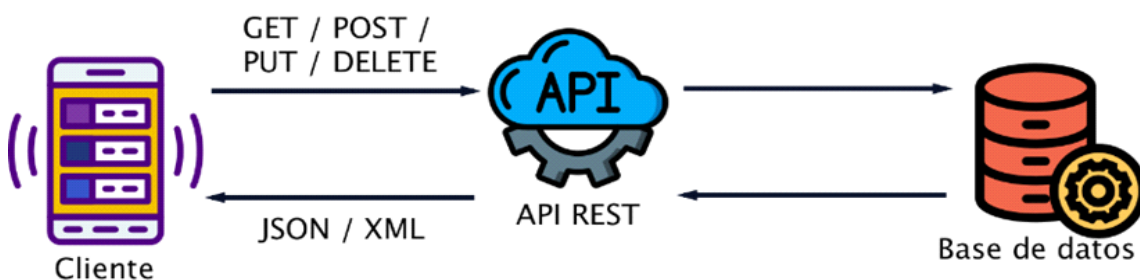
La visión computacional es un área de la inteligencia artificial que permite a las computadoras La visión computacional es una rama de la inteligencia artificial que permite a las computadoras interpretar y analizar el mundo visual. De La Escalera Hueso (2020) destaca su uso en el análisis de imágenes en tiempo real para reconocer objetos, detectar movimientos y extraer información de escenas completas.

Técnicas como la segmentación de imágenes, la detección de bordes y la extracción de características son esenciales para aumentar la precisión de estos sistemas, permitiendo la identificación y clasificación de objetos, el reconocimiento de patrones y el seguimiento de cambios en el entorno.

OpenCV, una biblioteca de código abierto mencionada por Itseez (2018), facilita el procesamiento y análisis de imágenes y videos en tiempo real, con aplicaciones comunes como el reconocimiento facial, la detección de movimiento y la realidad aumentada. La integración de visión computacional e inteligencia artificial optimiza la toma de decisiones basadas en datos visuales, creando sistemas más robustos y eficientes (De La Escalera Hueso, 2020).

API REST

Las API RESTful (Representational State Transfer) son arquitecturas de software diseñadas para crear servicios web escalables y eficientes. Como se muestra en la Ilustración 4, estas APIs facilitan la comunicación entre sistemas utilizando el protocolo.



ILUSTRACION 4. Dossetenta (2024). Descripción de un modelo de análisis de datos: La Ilustración muestra un modelo de análisis de datos donde una aplicación móvil se comunica con una API, que a su vez interactúa con una base de datos.

Según Fielding (2000), las API RESTful se basan en operaciones CRUD (Create, Read, Update, Delete) ejecutadas a través de verbos HTTP (POST, GET, PUT, DELETE), lo que permite una interacción estandarizada entre aplicaciones y facilita la integración de datos.

En el desarrollo de aplicaciones modernas, las API RESTful son cruciales para la interoperabilidad entre sistemas y plataformas. Frameworks como Laravel son frecuentemente utilizados para implementar estas API debido a su robustez y capacidades avanzadas para manejar solicitudes HTTP, autenticación y gestión de bases de datos (Otwell, 2020).

Laravel ofrece una estructura clara y herramientas integradas que simplifican el desarrollo y mantenimiento de aplicaciones web, mejorando la eficiencia en el despliegue de aplicaciones (Campoverde et al., 2024).

PROTOCOLO RTSP

El Protocolo de Transmisión en Tiempo Real (RTSP) es un protocolo de red que controla la transmisión de datos multimedia en tiempo real. Según Schulzrinne et al. (1998), RTSP permite a los clientes emitir comandos para iniciar, detener y pausar la transmisión de medios desde un servidor. Es fundamental para aplicaciones que requieren la transmisión en directo de audio y video, como videoconferencias y cámaras de seguridad.

RTSP facilita el manejo de sesiones interactivas de medios mediante comandos similares a HTTP, como DESCRIBE, SETUP, PLAY, PAUSE y TEARDOWN, permitiendo un control preciso de la transmisión. La similitud con HTTP facilita su implementación y uso para desarrolladores familiarizados con este protocolo (Fielding, 2000).

RTSP se utiliza en una variedad de dispositivos y aplicaciones, desde sistemas de videovigilancia hasta plataformas de streaming como Netflix y YouTube, siendo especialmente útil en entornos donde la latencia y la calidad de transmisión son críticas (Wright y Stevens, 2011).

Herramientas empleadas en la construcción del software

Las herramientas clave empleadas en la construcción del software incluyen:

Laravel: Un framework PHP que facilita el desarrollo de aplicaciones web robustas con una sintaxis elegante. Laravel gestiona tareas como enrutamiento, autenticación y bases de datos, y es ideal para crear API RESTful. Además, incluye Blade, un sistema de plantillas, y Eloquent ORM, que simplifica las interacciones con bases de datos (Otwell, 2020).

Python: Un lenguaje versátil utilizado en desarrollo de software, ciencia de datos y aprendizaje automático. Bibliotecas como NumPy, pandas y scikit-learn permiten el desarrollo rápido de aplicaciones complejas.

OpenCV: Una biblioteca para visión por computadora utilizada en procesamiento de imágenes y videos. Facilita aplicaciones como el reconocimiento facial y la detección de movimiento, siendo clave en sistemas de análisis visual en tiempo real (Itseez, 2018).

PyTorch: Una biblioteca de aprendizaje profundo que permite el desarrollo ágil de modelos de redes neuronales, utilizada en visión por computadora y procesamiento de lenguaje natural (Paszke et al., 2019).

Arquitectura de software

La arquitectura de software define la estructura de un sistema, incluyendo sus componentes y sus relaciones. Una buena arquitectura proporciona una base sólida para el desarrollo, permitiendo cambios eficientes y controlados (Bass et al., 2003). También es esencial para cumplir requisitos no funcionales como escalabilidad, seguridad y rendimiento. La elección del estilo arquitectónico adecuado, como REST para servicios web, puede mejorar la flexibilidad y escalabilidad del sistema (Fielding, 2000).

Documentar la arquitectura es crucial para comunicar las decisiones de diseño a todo el equipo de desarrollo (Bass et al., 2003). Esto incluye diagramas, patrones arquitectónicos y decisiones de diseño que clarifican cómo deben interactuar los componentes del sistema. Una documentación efectiva asegura una comprensión común del sistema y facilita su evolución y mantenimiento (Sañay et al., 2019).

Metodología o Proceso de Desarrollo de Software

En el desarrollo de software, se utilizó el Proceso Unificado (PU), un enfoque iterativo que guía al equipo a través de diferentes fases, permitiendo ajustes y mejoras continuas. Este proceso se complementó con el modelo FURPS+, que clasifica y evalúa los requisitos del software en términos de funcionalidad, usabilidad, fiabilidad, rendimiento y soporte, asegurando el cumplimiento de los estándares de calidad y las expectativas del cliente.

El PU se divide en cuatro fases principales: Iniciación, Elaboración, Construcción y Transición. En la fase de Iniciación, se define el alcance del proyecto, se identifican los riesgos y se elabora una visión general del sistema. Los entregables claves de esta fase incluyen un Documento de Visión y las Especificaciones de Requisitos, basadas en el modelo FURPS+.

En la fase de Elaboración, se desarrolla la arquitectura base del sistema y se refina el plan del proyecto, centrándose en resolver los riesgos más críticos. Los entregables incluyen la definición de la Arquitectura del Sistema, Prototipos de Interfaces y un Plan de Iteraciones detallado. Durante la fase de Construcción, se desarrolla el sistema completo mediante iteraciones, con entregables como el Código Fuente, Documentación Técnica y un Manual de Usuario.

Finalmente, en la fase de Transición, se entrega el sistema al usuario final, asegurando su instalación, soporte inicial y capacitación. Los entregables principales de esta fase son el Sistema Implementado, las Pruebas de Aceptación y la Capacitación del usuario final para garantizar un uso eficiente del software.

ENTREGABLES DEL PROCESO

Fase de Inicio

Análisis de Requisitos aplicando el modelo FURPS+

- **Funcionalidad (F):**
 - **CU 001:** El sistema debe contar personas en tiempo real usando cámaras IP, con captura y procesamiento de video mediante OpenCV y modelos de aprendizaje profundo en PyTorch. Los datos se almacenan en una base de datos PostgreSQL.
 - **CU 002:** Segmentación por género y edad utilizando modelos de PyTorch (género) y Keras (edad).
 - **CU 003:** Identificación de empleados a través de una interfaz gráfica, con almacenamiento de información en la base de datos.
 - **CU 004:** Almacenamiento seguro de todas las detecciones, con integración del Backend en Laravel y APIs REST.
 - **CU 005:** Configuración de cámaras IP mediante una interfaz gráfica para definir líneas de entrada y salida.
 - **CU 006:** Visualización en tiempo real de las detecciones a través de una interfaz gráfica.
 - **CU 007:** Interfaz gráfica intuitiva desarrollada en Tkinter para configuración y visualización.
- **Usabilidad (U):** Interfaz intuitiva con elementos gráficos claros, retroalimentación inmediata, y configuración guiada de cámaras y líneas de detección.
- **Confiabilidad (R):** Almacenamiento seguro en PostgreSQL, mecanismos de recuperación automática de errores en las cámaras IP.
- **Rendimiento (P):** Procesamiento en tiempo real con modelos optimizados en PyTorch y OpenCV, escalabilidad para múltiples cámaras IP.
- **Soporte (S):** Documentación completa, soporte técnico, y mantenimiento fácil del sistema.
- **Requerimientos Adicionales (+):** Integración con otros sistemas a través de APIs RESTful.

Casos de Uso y Diagrama de Secuencia

El sistema "Monitoreo y control de flujo de personas en tiempo real" (MCFTR) utiliza cámaras IP, inteligencia artificial y visión computacional para monitorear personas como se muestra en la ilustración 5. Los actores incluyen cámaras IP, el encargado de cámaras, seguridad física, backend y base de datos.

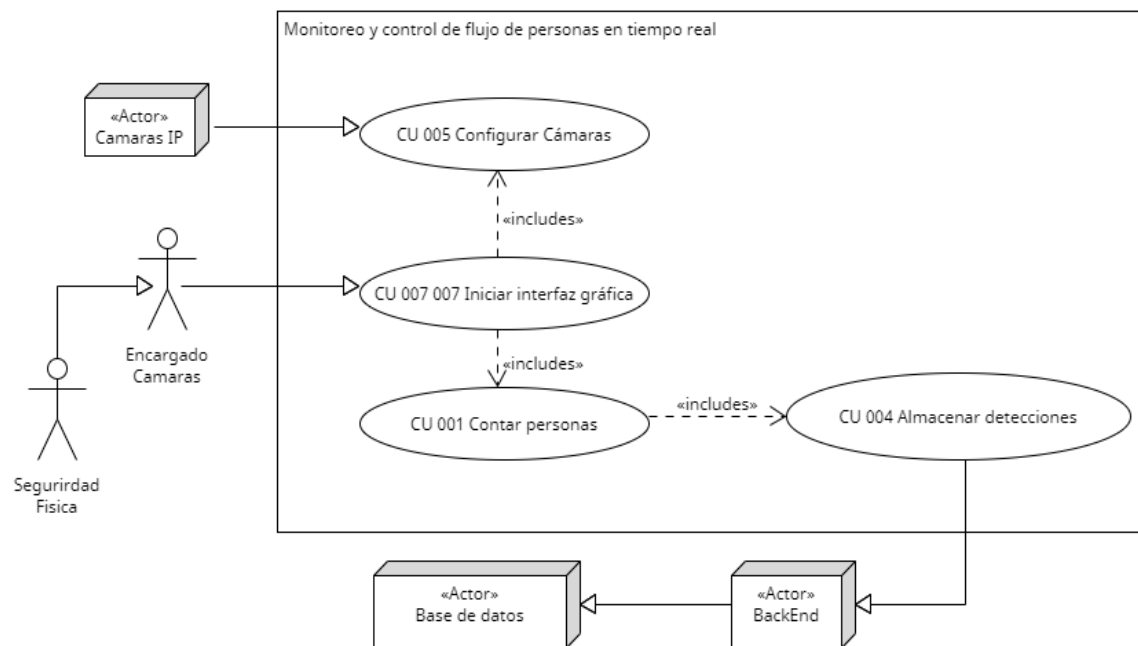


Ilustración 5. Diagrama de Caso de Uso: El diagrama de caso de uso proporciona una vista general de cómo los diferentes actores interactúan con las funcionalidades del sistema de monitoreo y control de flujo de personas en tiempo real.

- **Configuración de Cámaras (CU 005).** - Añadir, eliminar y ajustar cámaras y líneas de detección.
- **Conteo de Personas (CU 001).** - Captura y procesamiento de video para detectar y contar personas.
- **Almacenamiento de Detecciones (CU 004).** - Backend centraliza almacenamiento y gestión de datos en PostgreSQL.
- **Detección de Género y Edad (CU 002).** - Segmentación de personas detectadas.
- **Identificación de Empleados (CU 003).** - Marcado manual y almacenamiento de información.
- **Visualización de Detecciones (CU 006).** - Interfaz gráfica muestra detecciones y permite análisis en tiempo real.

Diagrama de Secuencias

El diagrama de secuencias que se observa en la ilustración 6, describe el flujo de interacción entre el encargado de cámaras, las cámaras IP, el PC principal, el contador de personas, el backend, y la base de datos. Destaca las etapas de configuración, captura de video, procesamiento, almacenamiento, y visualización, mostrando cómo cada componente interactúa para asegurar un conteo preciso y en tiempo real de personas.

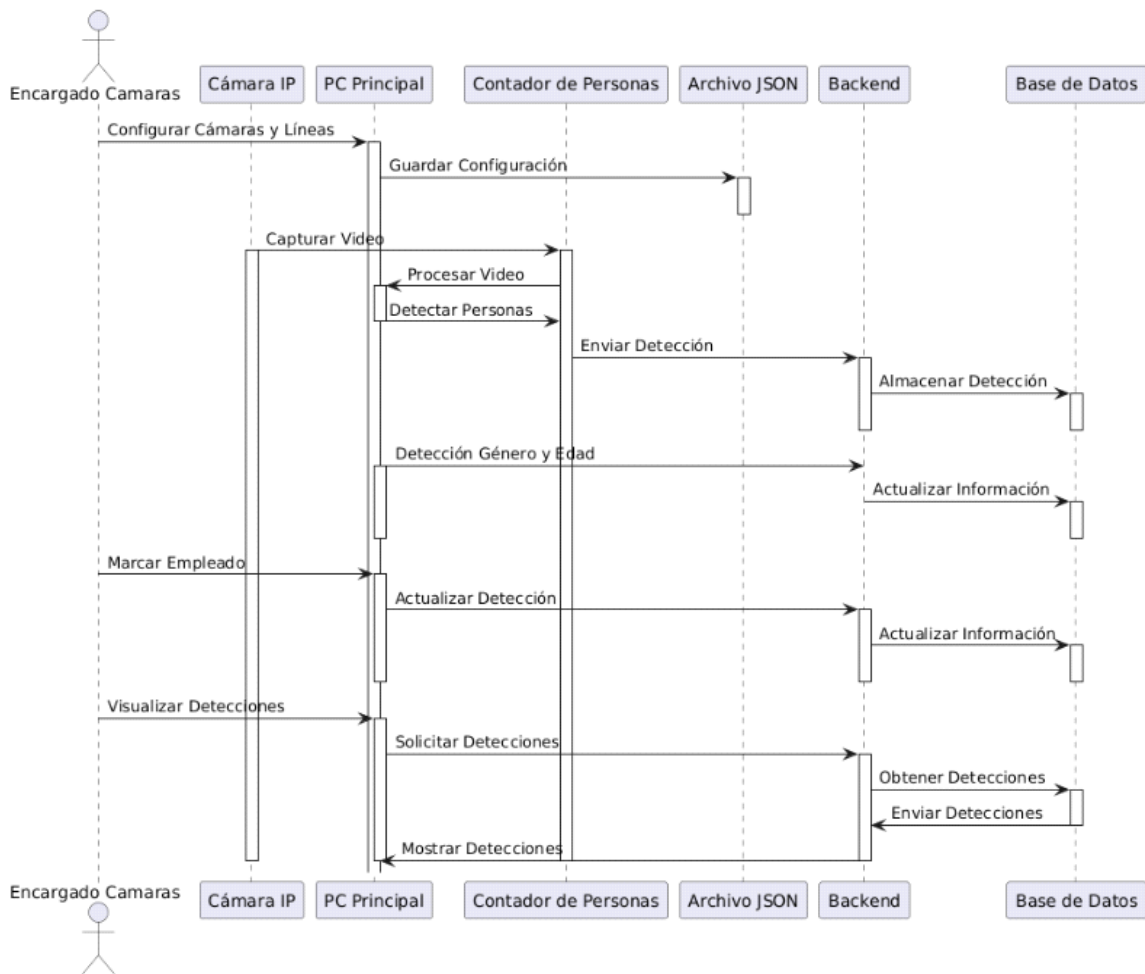


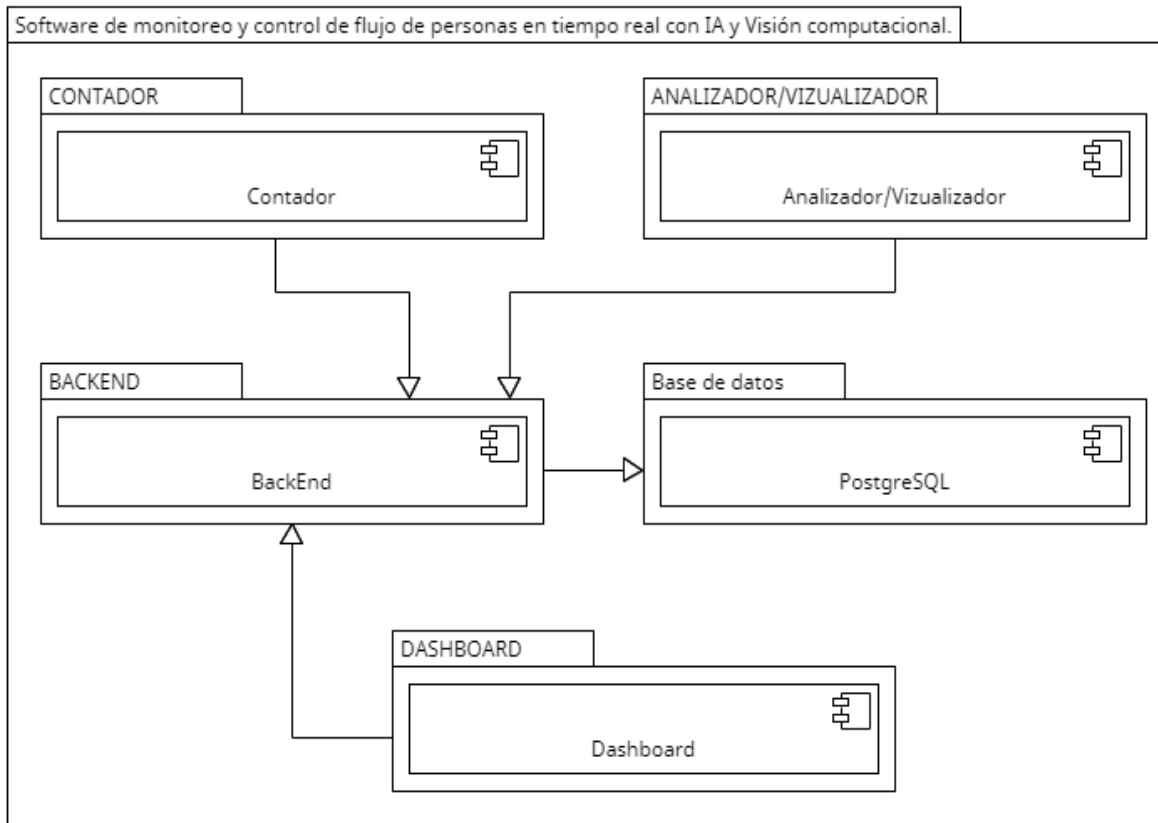
Ilustración 6. Diagrama de secuencias: El diagrama de secuencia muestra la interacción entre el Encargado de cámaras, la Cámara IP, el PC Principal, el Contador de Personas, el Archivo JSON, el Backend y la Base de Datos durante la configuración de cámaras, captura de video, procesamiento y detección de personas, almacenamiento y actualización de detecciones, y visualización de detecciones.

FASE DE ELABORACIÓN

El MCFTR utiliza cámaras IP distribuidas en varias ubicaciones que se conectan a una red mediante el protocolo RTSP, enviando video en tiempo real a un servidor central. Este servidor aloja las aplicaciones de conteo y análisis, procesando las solicitudes de los usuarios y gestionando la base de datos PostgreSQL, que almacena las detecciones y configuraciones. Los usuarios interactúan con el sistema a través de una interfaz gráfica, donde configuran cámaras, visualizan estadísticas en tiempo real y analizan las detecciones almacenadas.

Arquitectura del Sistema

Durante esta fase, se desarrolló una arquitectura base y se refinaron los planes del proyecto, enfocándose en mitigar los riesgos más críticos identificados anteriormente. Se diseñó la arquitectura del sistema tanto a nivel lógico como físico para asegurar la comprensión de cómo los componentes interactúan y cómo se implementan en la infraestructura.



ILUSTRACION 7. Software de monitoreo y control de flujo de personas en tiempo real con IA y visión computacional. El diagrama de integración muestra las interacciones entre los componentes del sistema: Contador, Analizador/Visualizador, Backend, Dashboard y la Base de datos PostgreSQL. El Contador, Analizador/Visualizador y Dashboard envían y reciben datos a través del Backend, que a su vez se comunica con la Base de datos para almacenar y recuperar información.

La arquitectura lógica (Ilustración 7) detalla los módulos principales: el Backend, el Contador de Personas, el Analizador y Visualizador de Detecciones, y el Dashboard. Cada módulo tiene roles específicos y claramente definidos para asegurar un flujo de datos eficiente. El Backend actúa como el núcleo del sistema, gestionando solicitudes HTTP, proporcionando endpoints RESTful, manejando la lógica de negocio, y facilitando la interacción con la base de datos PostgreSQL. El Contador captura y procesa video en tiempo real, utilizando tecnologías de visión computacional y aprendizaje profundo (OpenCV y PyTorch) para detectar y contar personas. El Analizador y Visualizador predice el género y la edad de las personas detectadas mediante modelos de PyTorch y Keras, y permite la visualización y ajuste de las detecciones. El Dashboard permite a los usuarios visualizar estadísticas y realizar consultas detalladas sobre el flujo de personas, utilizando datos proporcionados por el backend.

La arquitectura física (Ilustración 11) muestra cómo el sistema se implementa en términos de infraestructura: servidores, bases de datos y conexiones de red. Esta arquitectura es fundamental para asegurar que el sistema sea escalable y capaz de manejar la carga esperada de datos y solicitudes.

Funcionalidades Clave

- **Backend.** - Gestiona las solicitudes HTTP y proporciona endpoints RESTful para operaciones CRUD, implementando la lógica de negocio y facilitando la interacción con la base de datos.
- **Contador de Personas.** - Captura y procesa video para detectar personas en tiempo real, utilizando OpenCV y PyTorch.
- **Analizador y Visualizador.** - Analiza las detecciones para predecir género y edad, proporcionando una interfaz gráfica para la visualización y ajuste de datos.
- **Dashboard.** - Visualiza estadísticas agregadas y genera reportes basados en los datos almacenados, interactuando con el backend para obtener información actualizada.

Esta integración asegura que el sistema funcione de manera eficiente y coherente, proporcionando monitoreo y análisis en tiempo real del flujo de personas en las instalaciones.

FASE DE CONSTRUCCIÓN

Desarrollo de Módulos del Sistema

Contador de Personas. - El módulo Contador de Personas es responsable de capturar y procesar video en tiempo real para detectar y contar personas utilizando técnicas avanzadas de visión computacional y aprendizaje profundo. En la Ilustración 8 se presenta su diagrama de clases, que incluye las clases principales: MultiVideoPersonCounter, PersonCounter, VideoStreamProcessor, y GUIUpdateThread.

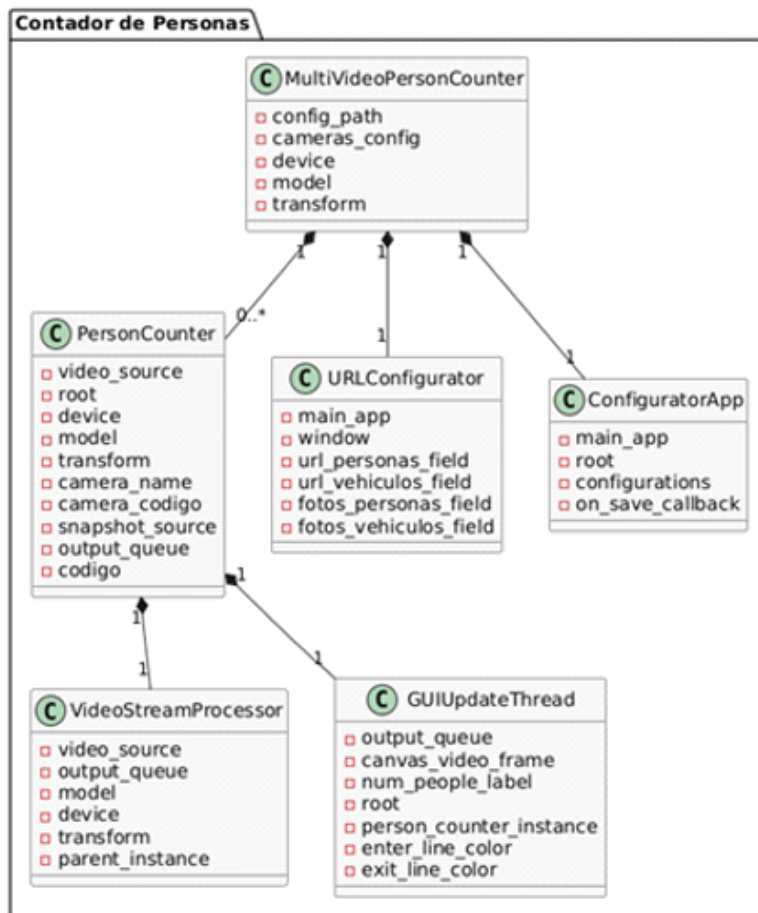


Ilustración 8. Diagrama de clases del contador: El diagrama de clases muestra las relaciones entre las clases principales del sistema de conteo de personas, incluyendo *MultiVideoPersonCounter*, *PersonCounter*, *URLConfigurator*, *ConfiguratorApp*, *VideoStreamProcessor*, y *GUIUpdateThread*, junto con sus atributos clave.

Los módulos del subsistema contador de persona se describen a continuación:

- **MultiVideoPersonCounter.** - Gestiona múltiples cámaras IP, inicializando el modelo YOLO para detección de personas. El modelo se carga con su configuración y pesos específicos para garantizar detecciones rápidas y precisas.
- **PersonCounter.** - Captura video en tiempo real y procesa cada fotograma usando YOLO para detectar personas, aplicando supresión de no máximos para eliminar detecciones redundantes.
- **VideoStreamProcessor.** - Procesa el flujo de video en tiempo real en un hilo separado, utiliza YOLO para la detección de objetos y envía las detecciones a una cola de salida para su procesamiento posterior.
- **GUIUpdateThread.** - Actualiza la interfaz gráfica con las detecciones en tiempo real, mostrando los resultados en la GUI.

Analizador y Visualizador de Detecciones

El módulo Analizador y Visualizador de Detecciones analiza las detecciones para predecir género y edad, utilizando modelos de aprendizaje profundo. En la Ilustración 9 se muestra su diagrama de clases, destacando las clases *MainApp*, *VisualizerApp*, y *AnalyzerApp*.

- **MainApp:** Administra la interfaz gráfica, permitiendo a los usuarios alternar entre los modos de visualización y análisis.
- **AnalyzerApp:** Gestiona la lógica de análisis, prediciendo género y edad con un pipeline de clasificación de imágenes preentrenado.
- **VisualizerApp:** Gestiona la visualización de detecciones, permitiendo filtros basados en criterios como fecha, categoría, edad, y ubicación.

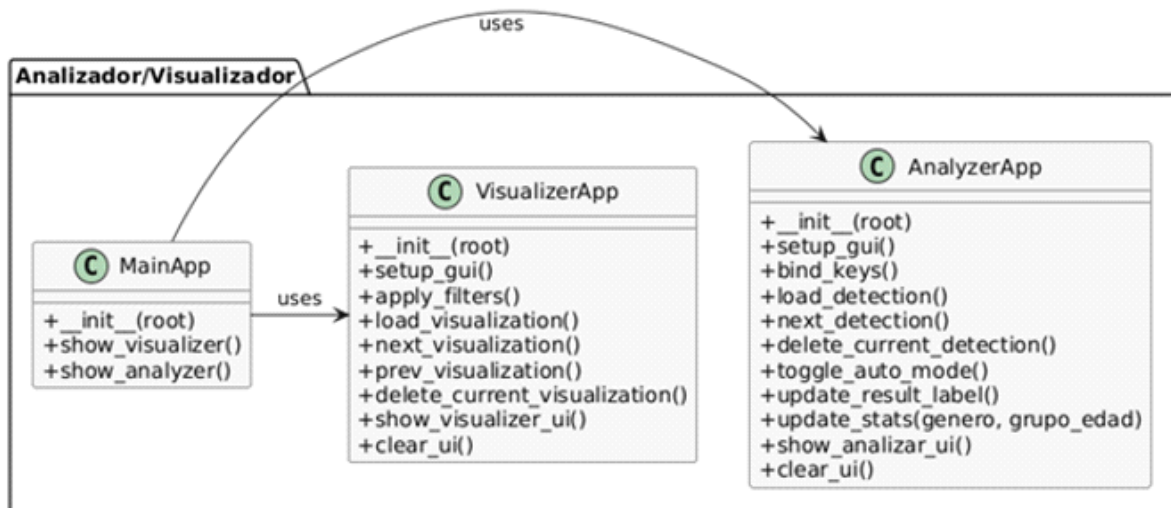


Ilustración 9. Diagrama de clases Analizador/Visualizador: El diagrama de clases muestra las relaciones entre MainApp, VisualizerApp y AnalyzerApp, incluyendo los métodos utilizados para la visualización y análisis de datos.

Backend y Dashboard

El Backend gestiona solicitudes HTTP, proporciona endpoints RESTful y maneja la lógica de negocio. En la Ilustración 10 se detalla el diagrama de clases, que incluye WebRoutes, DetectionController, Detection, y DashController.

- **Detection.** - Modelo de datos para las detecciones, incluyendo atributos como ID, nombre, fecha, género, y edad.
- **DetectionController.** - Maneja solicitudes HTTP relacionadas con las detecciones, permitiendo operaciones CRUD.
- **WebRoutes.** - Define las rutas web para manejar operaciones CRUD de detecciones.
- **DashController.** - Gestiona la lógica del dashboard, proporcionando métodos para visualizar estadísticas y cargar datos específicos.

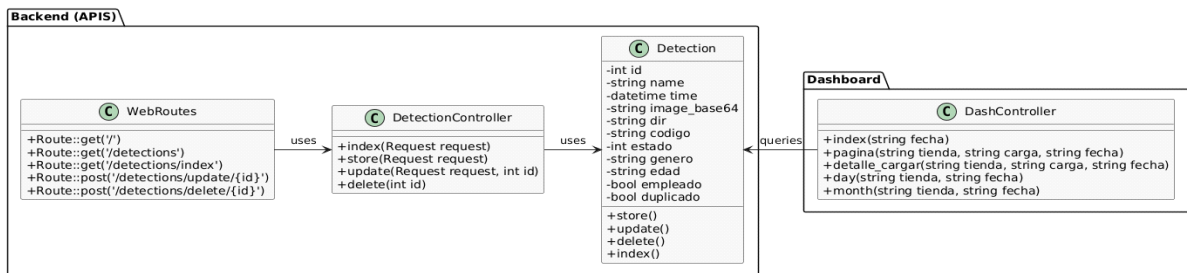


Ilustración 10. Diagrama de clases Backend (APIs): El diagrama de clases muestra la estructura del backend con las clases *WebRoutes*, *DetectionController*, *Detection* y *DashController*, destacando las relaciones y métodos utilizados para gestionar rutas, detecciones y consultas al dashboard.

FASE DE TRANSICIÓN

Implementación de la Arquitectura del Sistema

La arquitectura del sistema se ha implementado con varios componentes clave que interactúan para ofrecer funcionalidades de conteo y análisis de personas:

- **Cámaras IP:** Conectadas a la red local mediante el protocolo SRTP, capturan video en tiempo real en distintos locales y envían los datos a la PC Principal.
- **PC Principal:** Aloja los programas de conteo, análisis y visualización. Procesa el video recibido de las cámaras, realizando detecciones y análisis en tiempo real, y generando los datos necesarios para su visualización.
- **Servidor Central:** Alojado en una infraestructura dedicada, contiene el backend desarrollado en Laravel y la base de datos PostgreSQL. Gestiona solicitudes HTTP, almacena detecciones y ofrece endpoints RESTful para la interacción con otros componentes.
- **PC Usuario:** Permite a los usuarios configurar cámaras, utilizar las aplicaciones de conteo y análisis, y conectarse al servidor central para visualizar y analizar las detecciones almacenadas.
- **Red:** Los usuarios autorizados pueden acceder al dashboard desde cualquier red permitida para ver estadísticas y realizar análisis adicionales.

Esta infraestructura garantiza la coordinación eficiente entre los componentes para procesar, almacenar y analizar datos en tiempo real.

Funcionamiento del Sistema

El sistema consta de dos módulos principales: **Contador** y **Analizador/Visualizador**.

- **Contador.** - Captura video en tiempo real, detecta personas y actualiza los datos en el servidor. Se configuran las URLs del servidor y las cámaras que capturarán el flujo de personas, definiendo las líneas de entrada y salida.

- **Analizador/Visualizador.** - Carga imágenes no analizadas, predice género y edad, y permite ajustes manuales. Proporciona una interfaz para revisar y modificar datos de detecciones.

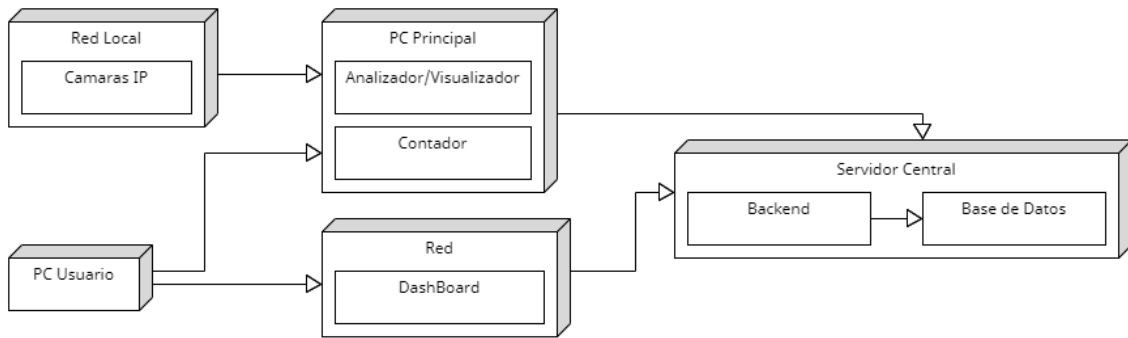


Ilustración 11. Arquitectura de nivel físico: El diagrama físico muestra la estructura del sistema, destacando la interconexión entre las Cámaras IP, el PC Principal (con programas de análisis y conteo), el Servidor Central (con servidor backend y base de datos), el PC del Usuario y el Dashboard en la red.

En la Ilustración 11 se muestra la estructura del sistema, desde la captura de video hasta la visualización y análisis de datos. La Ilustración 12 detalla la integración de los módulos, mostrando cómo se relacionan entre sí para mantener un flujo eficiente de información y procesamiento de datos en tiempo real.

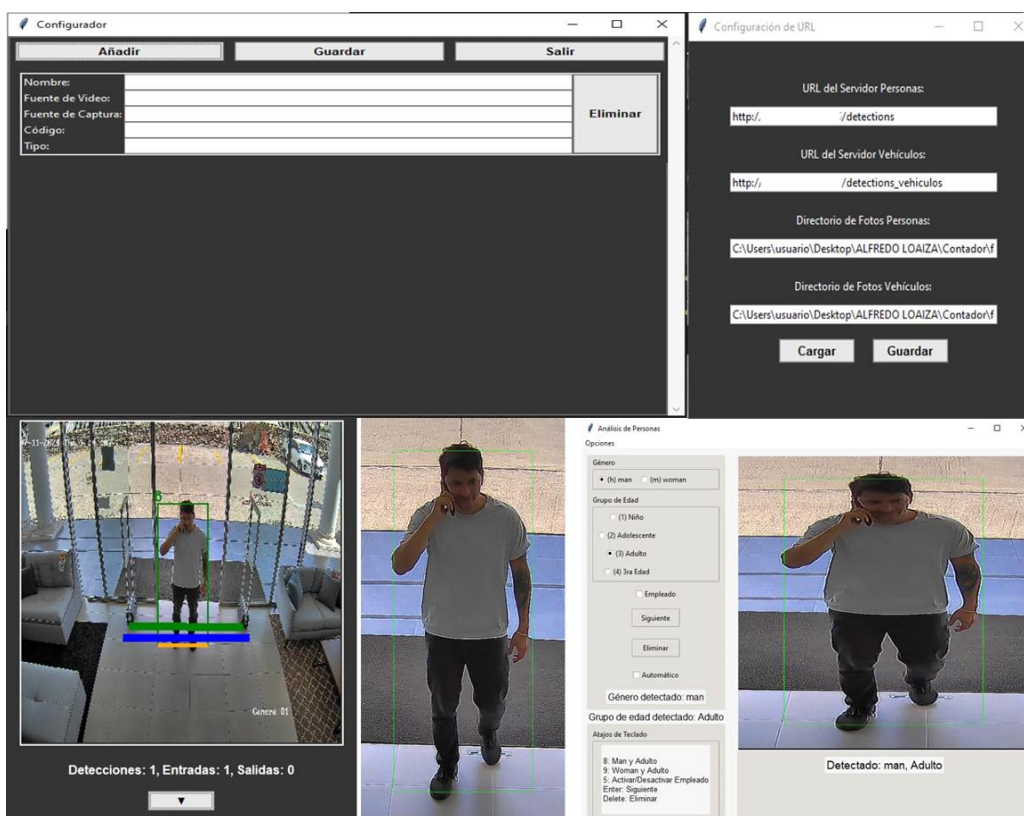


Ilustración 12. Funcionamiento del sistema: Muestra la integración y relación entre los módulos principales del sistema de conteo de personas.

El Servidor Central almacena las detecciones en la base de datos y proporciona APIs para la interacción con otros módulos. El Analizador/Visualizador utiliza esta información para

predecir datos demográficos, asegurando una monitorización precisa del flujo de personas.

Validación y Pruebas

Para validar el funcionamiento del MCFTR, se realizó una prueba final en un entorno de producción real de 5 días, durante la liquidación de productos de la empresa. Se instalaron cámaras en ubicaciones estratégicas para evaluar la precisión y eficacia del sistema:

- **Puerta Principal.** - Capturó el flujo de entrada y salida de personas.
- **Puerta Trasera.** - Monitoreó el flujo de personas por la entrada secundaria.
- **Parqueadero.** - Contó el número de vehículos que ingresaban al estacionamiento.

Métodos de Comparación

Se emplearon varios métodos para validar los datos del sistema automatizado, entre los que se menciona:

- **Conteo Manual en la Puerta Trasera.** - Realizado por empleados para obtener una referencia directa, aunque solo proporcionaba un total general sin segmentar por género, edad o incluir a los empleados.
- **Sistema Anterior en las Puertas Principal y Trasera.** - Comparación con el sistema anterior, que diferenciaba entre hombres y mujeres, y empleados, pero no proporcionaba rangos de edad y solo contabilizaba adultos.
- **Contador de Infrarrojo en la Puerta Principal.** - Utilizado como medida adicional, sin diferenciación entre entradas y salidas, género, edad, o identificación de empleados.

Resultados de las Comparaciones

Los resultados mostraron que el sistema automatizado del Proyecto Contador ofreció una mayor precisión en el conteo de personas y vehículos, gracias a tecnologías avanzadas de inteligencia artificial y visión computacional.

Comparaciones Específicas

- **Proyecto Contador vs. Sistema Anterior**
 - El sistema anterior presentó un margen de error significativo, con hasta 100 personas no contabilizadas en comparación con el Proyecto Contador.
 - Las diferencias se debieron a la segmentación incompleta y la falta de respaldo de imágenes en el sistema anterior, mientras que el Proyecto Contador utiliza algoritmos avanzados y cada detección está respaldada por una imagen.
- **Proyecto Contador vs. Contador de Infrarrojo:**

- El contador de infrarrojos mostró un margen de error significativo debido a la falta de diferenciación entre entradas, salidas, género y edad.
- Las diferencias se atribuyeron a errores en la detección y la falta de especificidad en el conteo total.
- **Proyecto Contador vs. Conteo Manual:**
 - El conteo manual proporcionó una referencia directa para evaluar la precisión del sistema.
 - Al excluir empleados y segmentar por edad, se logró una comparación más precisa, destacando la fiabilidad del Proyecto Contador.

Análisis de Resultados

El análisis detallado se centró en la comparación con los datos del conteo manual por ser considerado el método más fiable:

- **Fiabilidad.** - El conteo manual se considera un método estándar para validar nuevos sistemas de conteo automatizado.
- **Limitaciones de Otros Métodos.** -
 - El sistema anterior no segmenta por edad y no es tan preciso como el Proyecto Contador.
 - El contador de infrarrojos no diferencia entradas y salidas, género o edad.

TITULO DIA	PROYECTO CONTADOR			CONTEO MANUAL
	Hombres	Mujeres	Total	Total
MIERCOLES	145	176	321	294
JUEVES	163	154	317	259
VIERNES	132	140	272	299
SABADO	236	231	467	502
DOMINGO	259	260	519	514

Tabla 1. Resultados del Proyecto Contador y Conteo Personal

Día	Total, Proyecto Contador	Total, Manual	Porcentaje de Error (%)
Miércoles	321	294	9.18
Jueves	317	259	22.39
Viernes	272	299	9.03
Sábado	467	502	6.98
Domingo	519	514	0.97

Tabla 2. Porcentaje de Error entre el Proyecto Contador y el Conteo Personal

- **Margen de diferencia con el Conteo Manual.** - El margen de diferencia con el conteo manual es casi mínimo, lo que demuestra el alto grado de certeza del Proyecto Contador. Al excluir empleados y segmentar por edad, los datos obtenidos son comparables con el conteo manual, validando así la exactitud del sistema.

Día	Diferencia Absoluta (Total)	Porcentaje de Error (%)
Miércoles	27	9.18
Jueves	58	22.39
Viernes	27	9.03
Sábado	35	6.98
Domingo	5	0.97

Tabla 3. Diferencia entre el Proyecto Contador y el Conteo Manual

- Respaldo de Datos con Imágenes:** Una de las mayores ventajas del Proyecto Contador es que cada detección está respaldada con una imagen. Esto permite verificar y corroborar que no se han cometido errores en las detecciones, ofreciendo una mayor confianza en los datos obtenidos. No se trata solo de números; cada dato está documentado visualmente, asegurando que ninguna detección sea un dato falso.

Día	Total, Detecciones	Detecciones con Imágenes (%)
Miércoles	321	100
Jueves	317	100
Viernes	272	100
Sábado	467	100
Domingo	519	100

Tabla 4. Respaldo de Datos con Imágenes

- Segmentación y Diferenciación:** A diferencia del sistema anterior y del contador de infrarrojos, el Proyecto Contador proporciona una segmentación detallada por género y edad. Esto no solo mejora la precisión del conteo, sino que también ofrece datos más valiosos para la toma de decisiones estratégicas.

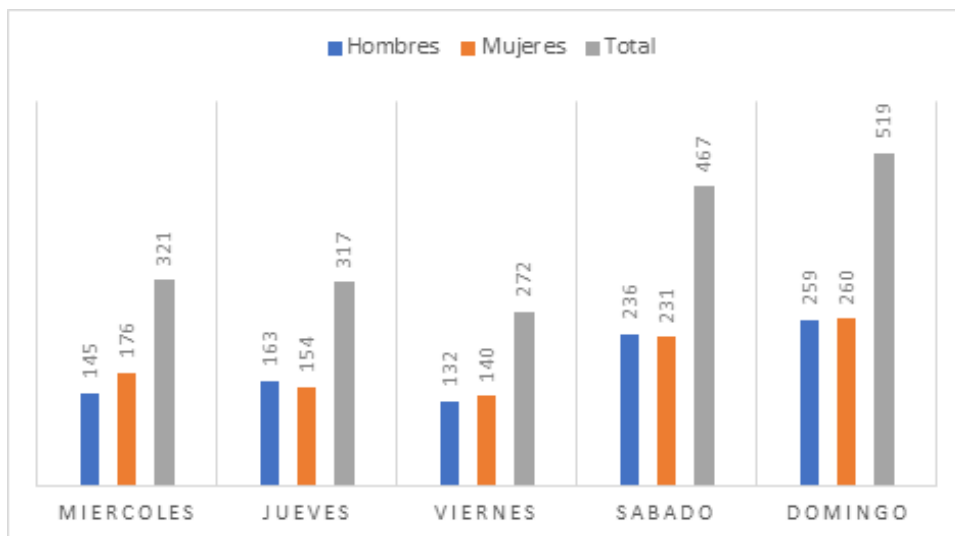


Ilustración 14. Resultados del Sistema: Gráfica de los datos de detección segmentados por género.

El MCFTR se establece como la mejor opción para el monitoreo y control del flujo de personas en tiempo real. Su alta precisión, respaldada por imágenes, y su capacidad para segmentar y diferenciar entre diferentes grupos demográficos lo hacen superior a los métodos anteriores y a otros sistemas automatizados. Esto asegura que las decisiones basadas en los datos del Proyecto Contador sean informadas y fiables, mejorando la eficiencia y seguridad en los espacios comerciales.

CONCLUSIONES

Evaluación Inicial: El análisis de los sistemas de video en tiempo real basados en redes neuronales reveló deficiencias en los métodos tradicionales de conteo de personas, identificando áreas clave para su mejora. La implementación de modelos avanzados de aprendizaje profundo, como YOLO, resultó fundamental para aumentar la precisión y eficiencia en la detección y seguimiento de personas en tiempo real.

Desarrollo del Software: Se logró desarrollar un sistema robusto que integra tecnologías de visión por computadora y aprendizaje profundo para la detección y conteo de personas. El software consta de módulos especializados como el Contador, Backend, Dashboard y Analizador/Visualizador, que operan de manera coordinada para ofrecer una solución integral y eficiente. Cada módulo fue diseñado y optimizado para su función específica, garantizando un flujo continuo de información y una operatividad constante del sistema.

Implementación en Entornos Reales: La implementación en un entorno operativo real permitió validar la funcionalidad y efectividad del sistema. Las pruebas realizadas en tiendas de muebles demostraron que el sistema puede operar en condiciones reales, capturando y procesando datos en tiempo real con un alto nivel de precisión. La capacidad del sistema para manejar múltiples fuentes de video y adaptarse a diversas configuraciones destacó su escalabilidad y flexibilidad.

Validación y Resultados: Las pruebas exhaustivas confirmaron que el software desarrollado cumple con los objetivos establecidos. La precisión en la detección y conteo de personas y en el análisis y visualización de datos se evaluaron positivamente. La retroalimentación obtenida permitió realizar ajustes y optimizaciones, garantizando un rendimiento óptimo y una alta confiabilidad del sistema.

AGRADECIMIENTO

Se expresa un sincero agradecimiento a la empresa por brindar la oportunidad de desarrollar y probar el sistema de conteo de personas en tiempo real en un entorno operativo real. Su constante apoyo y colaboración fueron esenciales para el éxito de este proyecto.

Un agradecimiento especial al equipo del departamento de Sistemas, cuyo compromiso, experiencia y disposición durante todas las fases de implementación y pruebas resultaron invaluable. Su dedicación y esfuerzo contribuyeron significativamente a alcanzar los objetivos planteados, asegurando el desarrollo de una solución robusta y eficaz.

Gracias por su confianza en este proyecto y por su valiosa contribución al avance de la tecnología en la empresa.

REFERENCIAS

Andrade, H. (2021). Descripción del funcionamiento de una red neuronal convolucional (CNN) [Diagrama]. ResearchGate. <https://www.researchgate.net/profile/Hugo-Andrade-14/publication/348825166/figure/fig1/AS:984814450667520@1611809470042/Figura-1-Descripcion-del-funcionamiento-de-una-red-neuronal-convolucional-CNN-10.png>

Araujo Sandoval, O. I. (2020). Modelo FURPS aplicado al análisis de calidad de un software desarrollado con Sencha Ext JS. REDDI Unlam. Recuperado de <http://reddi.unlam.edu.ar>

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley.

Baugher, M., McGrew, D., Naslund, M., Carrara, E., & Norrman, K. (2004). The Secure Real-time Transport Protocol (SRTP). IETF. Recuperado de <https://datatracker.ietf.org/doc/html/rfc3711>

Campoverde-Calle, M. S., Sañay-Sañay, S. I., & Cabrera-Duffaut, A. E. (2024). Desarrollo de metodología de validación de APPS basada en el modelo de aceptación tecnológica: Caso de estudio “BodyUC”. *MQR Investigar*, 8(1), 3742–3770. <https://doi.org/10.56048/MQR20225.8.1.2024.3742-3770>

Cordero Guzmán, D. M., & Sañay, I. S. (2020). Marco de trabajo para gestión de procesos de negocio (BPM): Caso de una empresa de servicios. *Revista Científica y Tecnológica UPSE (RCTU)*, 7(1), 43-53. <https://doi.org/10.26423/rctu.v7i1.509>

De La Escalera Hueso, A. (2020). *Visión por Computador*. Universidad de Málaga.

Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation, University of California, Irvine). Recuperado de https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Gómez Flores, W. (2020). *Red Neuronal Convolucional*. Instituto Politécnico Nacional.

Grinberg, M. (2018). *Flask web development: Developing web applications with Python*. O'Reilly Media.

Hernández González, A. (2017). Aplicación del proceso unificado de desarrollo a proyectos de software. ResearchGate. Recuperado de https://www.researchgate.net/profile/AnaisaHernandezGonzalez/publication/312656269_Aplicacion_del_Proceso_Unificado_de_Desarrollo_a_proyectos_de_softw

[are/links/58878a87a6fdcc6b791ec281/Aplicacion del Proceso Unificado de Desarrollo a proyectos de software.pdf](https://repository.unikom.ac.id/67052/1/Learning%20OpenCV%203%20Computer%20vision%20with%20Python%20%28%20PDFDrive.com%20%29.pdf)

Itseez. (2018). *Learning OpenCV 3: Computer vision with Python*. Recuperado de <https://repository.unikom.ac.id/67052/1/Learning%20OpenCV%203%20Computer%20Vision%20with%20Python%20%28%20PDFDrive.com%20%29.pdf>

McCarthy, J. (2021). What is artificial intelligence? Open University of Catalonia. Recuperado de <https://openaccess.uoc.edu/bitstream/10609/148039/3/QueEsLaInteligenciaArtificial.pdf>

Otwell, T. (2020). *Laravel: Up & Running*. Recuperado de <https://profmatiasgarcia.com.ar/uploads/tutoriales/Laravel-Clase1.pdf>

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. Recuperado de <https://www.cs.princeton.edu/courses/archive/fall19/cos484/lectures/pytorch.pdf>

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. University of Washington. Recuperado de <https://arxiv.org/abs/1804.02767>

Santana Núñez, J. M. (2015). Flujos de trabajo del proceso unificado [Diagrama]. ResearchGate. <https://www.researchgate.net/profile/Jose-Miguel-Santana-Nunez/publication/279751761/figure/fig1/AS:648590296903681@1531647385501/Flujos-de-trabajo-del-proceso-unificado.png>

Sañay, I., Becerra Molina, E., & Calle Masache, O. (2019). La ingeniería del software para el desarrollo industrial en la zona 6 del Austro. *RECIMUNDO*, 3(1), 1625-1643. [https://doi.org/10.26820/recimundo/3.\(1\).enero.2019.1625-1643](https://doi.org/10.26820/recimundo/3.(1).enero.2019.1625-1643)

Schulzrinne, H., Rao, A., & Lanphier, R. (1998). Real-Time Streaming Protocol (RTSP). RFC 2326. Internet Engineering Task Force (IETF).

Sumano, A. (2012). *Proceso Unificado*. Universidad Veracruzana.

Wright, G. R., & Stevens, W. R. (2011). *TCP/IP illustrated, Volume 2: The implementation*. Addison-Wesley.

Yungan Gualli, A. F., Morales Alarcón, C. H., Delgado Altamirano, J. E., & Espinoza Tinoco, L. M. (2019). Modelo FURPS para el análisis del rendimiento de frameworks JSF. *3C TIC. Cuadernos de desarrollo aplicados a las TIC*, 8(4), 6583. <https://doi.org/10.17993/3ctic.2019.84.6583>