

MANUAL TECNICO

DESARROLLO DE UN SISTEMA AUTOMATIZADO PARA EL CONTEO Y
|CLASIFICACIÓN DE LA AFLUENCIA VEHICULAR EN GIRARDOT UTILIZANDO
VISIÓN ARTIFICIAL Y SEGUIMIENTO DE OBJETOS

JUAN CAMILO BULA OYUELA

UNIVERSIDAD PILOTO DE COLOMBIA
SECCIONAL DEL ALTO MAGDALENA
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
GIRARDOT, CUNDINAMARCA
2025

TABLA DE CONTENIDO

1. INTRODUCCION	3
1.1. Objetivo del Manual	4
1.2. Alcance del Sistema.....	4
2. ARQUITECTURA DEL SISTEMA.....	5
2.1. Diagrama General del Sistema	5
2.2. Descripción del sistema	6
3. ANÁLISIS DEL SISTEMA	8
3.1. Diagrama de Casos de Uso	8
3.2. Descripción del diagrama de caso de uso	9
4. DISEÑO DEL SISTEMA.....	10
4.1. Diagrama de Clases.....	10
4.2. Descripción de Clases	11
4.3. Diagrama Entidad-Relación	13
4.4. Diccionario de datos.....	14
5. TECNOLOGÍAS UTILIZADAS.....	16
6. ESTRUCTURA DE ARCHIVOS Y CARPETAS	17
7. CONFIGURACIÓN DEL ENTORNO DE DESARROLLO.....	20
7.1. Instalación de Dependencias	20
7.2. Flujo de Uso.....	20
7.3. Conteo y Seguimiento.....	20
8. REQUERIMIENTOS MÍNIMOS DE HARDWARE PARA EL PROCESAMIENTO DE IMÁGENES CON YOLOV8	21

1. INTRODUCCION

Este manual técnico proporciona una guía detallada sobre el funcionamiento interno y la estructura del sistema de conteo de vehículos mediante visión por computador. Su propósito es facilitar la comprensión del diseño, los componentes y la lógica del sistema para los desarrolladores, técnicos o cualquier persona encargada de su mantenimiento y mejora. Este documento incluye descripciones técnicas de las funciones, estructuras de datos, interfaces del sistema, así como diagramas de modelado. La documentación resulta clave para garantizar la escalabilidad, mantenibilidad y correcta evolución del sistema.

1.1. Objetivo del Manual

El objetivo de este manual es ofrecer una referencia técnica clara y estructurada sobre el sistema de conteo de vehículos. Está dirigido principalmente a desarrolladores, ingenieros de software, técnicos en sistemas y personal de soporte que deseen comprender la arquitectura, los módulos principales y la lógica del software. También puede ser útil como documentación de respaldo para futuras mejoras o integración con otros sistemas.

1.2. Alcance del Sistema

El sistema tiene como finalidad automatizar el conteo de vehículos en videos mediante técnicas de visión artificial y seguimiento de objetos. Las funcionalidades principales incluyen:

- Detección y clasificación de objetos (carros, motos y buses).
- Registro del número de vehículos detectados por minuto.
- Interfaz gráfica amigable para selección de video y visualización de historial.
- Exportación de los resultados en formato Excel.
- Almacenamiento y consulta de datos en base de datos local.
- Herramienta visual para el dibujo de líneas de conteo de forma interactiva.

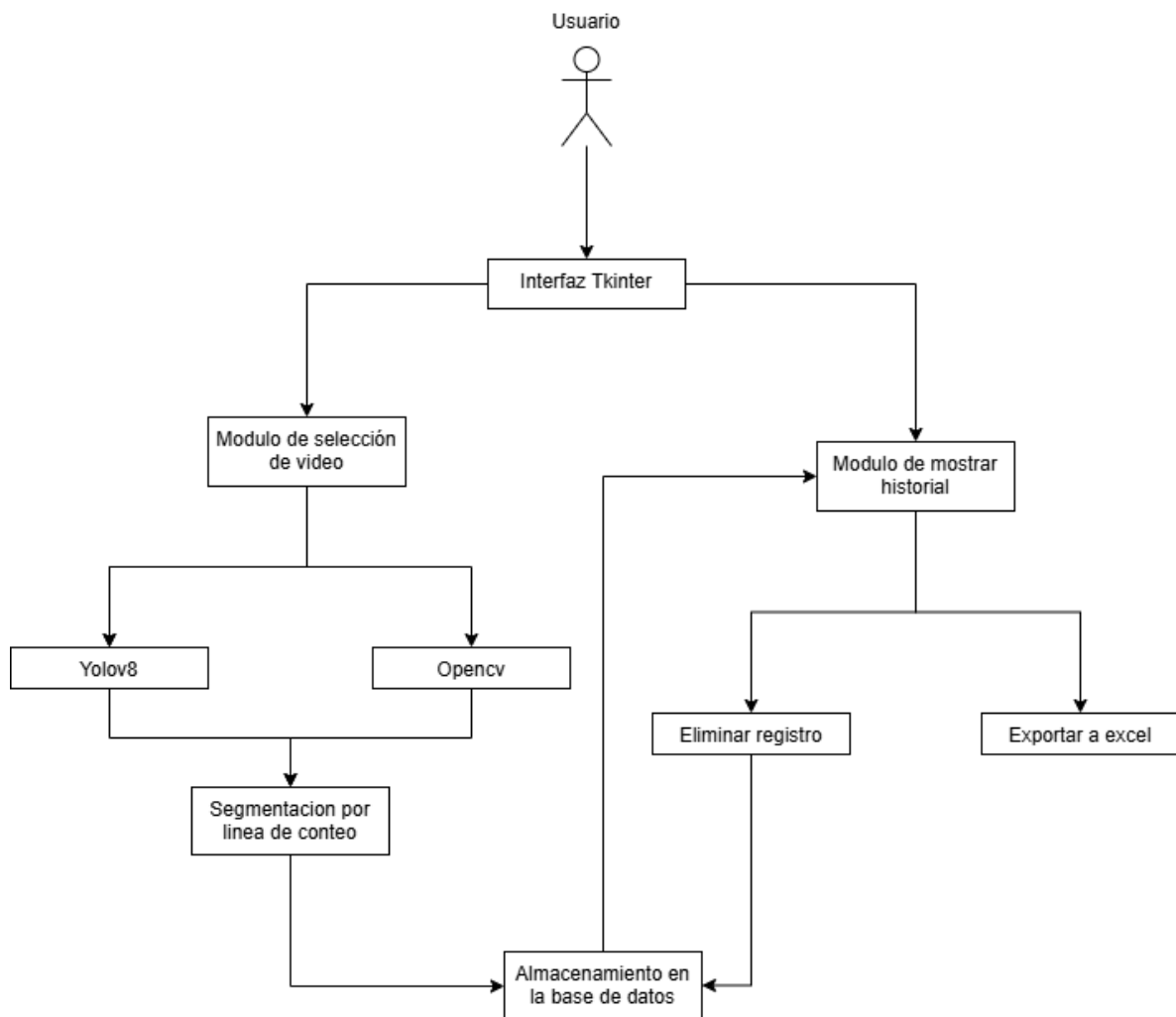
El sistema está diseñado para operar de manera local, con procesamiento por lotes de videos previamente grabados.

2. ARQUITECTURA DEL SISTEMA

2.1. Diagrama General del Sistema

En la Figura 1 se presenta el diagrama general del sistema de conteo de vehículos, mostrando los módulos principales y su interacción.

Figura 1. DIAGRAMA GENERAL DEL SISTEMA



Fuente: Elaboración propia.

2.2. Descripción del sistema

Este sistema está diseñado para interactuar con el usuario a través de una interfaz gráfica creada con Tkinter. A continuación, se detallan sus componentes y funcionalidades principales:

- Usuario:
 - El punto de entrada del sistema, el usuario puede comenzar a interactuar con la aplicación.
- Interfaz Tkinter:
 - Proporciona la plataforma visual para que el usuario seleccione opciones y visualice información.
- Módulo de Selección de Video:
 - Permite al usuario seleccionar un video para su procesamiento.
 - YOLOv8: Un marco de detección de objetos utilizado para analizar el contenido del video.
 - OpenCV: Otra biblioteca utilizada para realizar operaciones de procesamiento de imágenes.
 - Segmentación por Línea de Conteo: Un proceso que separa elementos o datos relevantes en el video.
- Almacenamiento en la Base de Datos:
 - Los datos procesados se almacenan en una base de datos para su posterior consulta y análisis.
- Módulo de Mostrar Historial:
 - Permite al usuario visualizar el historial de los videos procesados.
 - Eliminar Registro: Opción para borrar entradas específicas del historial.

- Exportar a Excel: Funcionalidad que permite exportar los datos del historial a un archivo Excel.

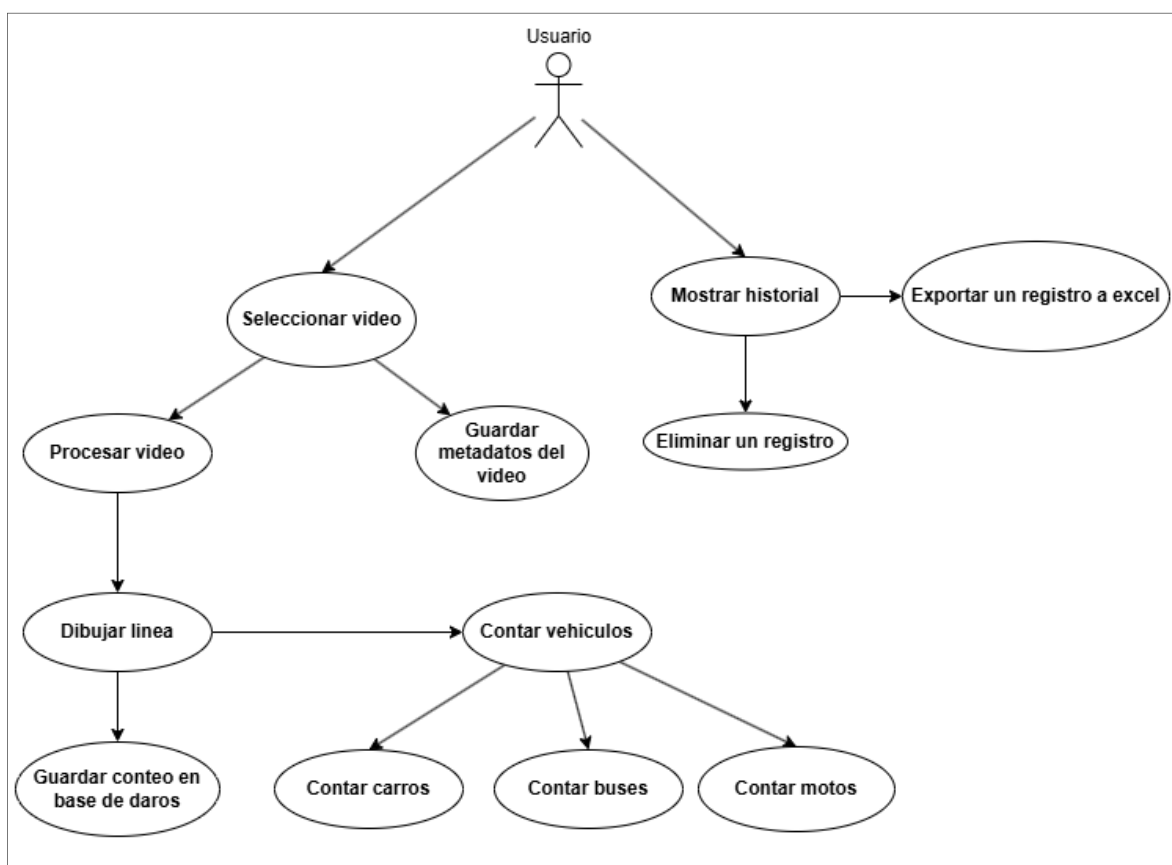
Este sistema está estructurado para facilitar el manejo y análisis de videos de manera amigable para el usuario, integrando tecnologías avanzadas de detección y procesamiento de imágenes.

3. ANÁLISIS DEL SISTEMA

3.1. Diagrama de Casos de Uso

La Figura 2 muestra el diagrama de casos de uso, el cual describe las principales interacciones entre el usuario y el sistema.

Figura 2. DIAGRAMA DE CASOS DE USO DEL SISTEMA



Fuente: Elaboración propia.

3.2.Descripción del diagrama de caso de uso

La Tabla 5 resume los principales elementos que componen el diagrama de casos de uso del sistema, detallando las acciones, actores y flujos asociados.

Tabla 1. LOS ELEMENTOS DEL DIAGRAMA DE CASO DE USO

CATEGORÍA	ELEMENTO	DESCRIPCIÓN
Actores	Usuario	Interactúa con el sistema mediante la interfaz.
Flujos Principales	Seleccionar Video	Selecciona un video que desea analizar.
	Procesar Video	Procesa el video para su análisis.
	Guardar Metadatos del Video	Se almacenan datos relevantes del video (nombre, fecha, duración.).
	Dibujar Línea	El usuario define una línea en el video para detectar el cruce de vehículos.
	Contar Vehículos	El sistema detecta y cuenta vehículos que cruzan la línea definida.
	└─ Contar Carros	Subproceso específico para contar automóviles.
	└─ Contar Buses	Subproceso específico para contar buses.
	└─ Contar Motos	Subproceso específico para contar motocicletas.
Acciones Adicionales	Mostrar Historial	consultar el historial de conteos realizados.
	Eliminar un Registro	Posibilidad de borrar un registro del historial.
	Exportar Registro a Excel	Opción para exportar los datos a un archivo Excel para análisis posterior.
Almacenamiento	Guardar en Base de Datos	Se almacenan los conteos y metadatos para futuras consultas o análisis.
Objetivo del Sistema		Facilitar el conteo y análisis de vehículos mediante una interfaz amigable y eficiente basada en datos.

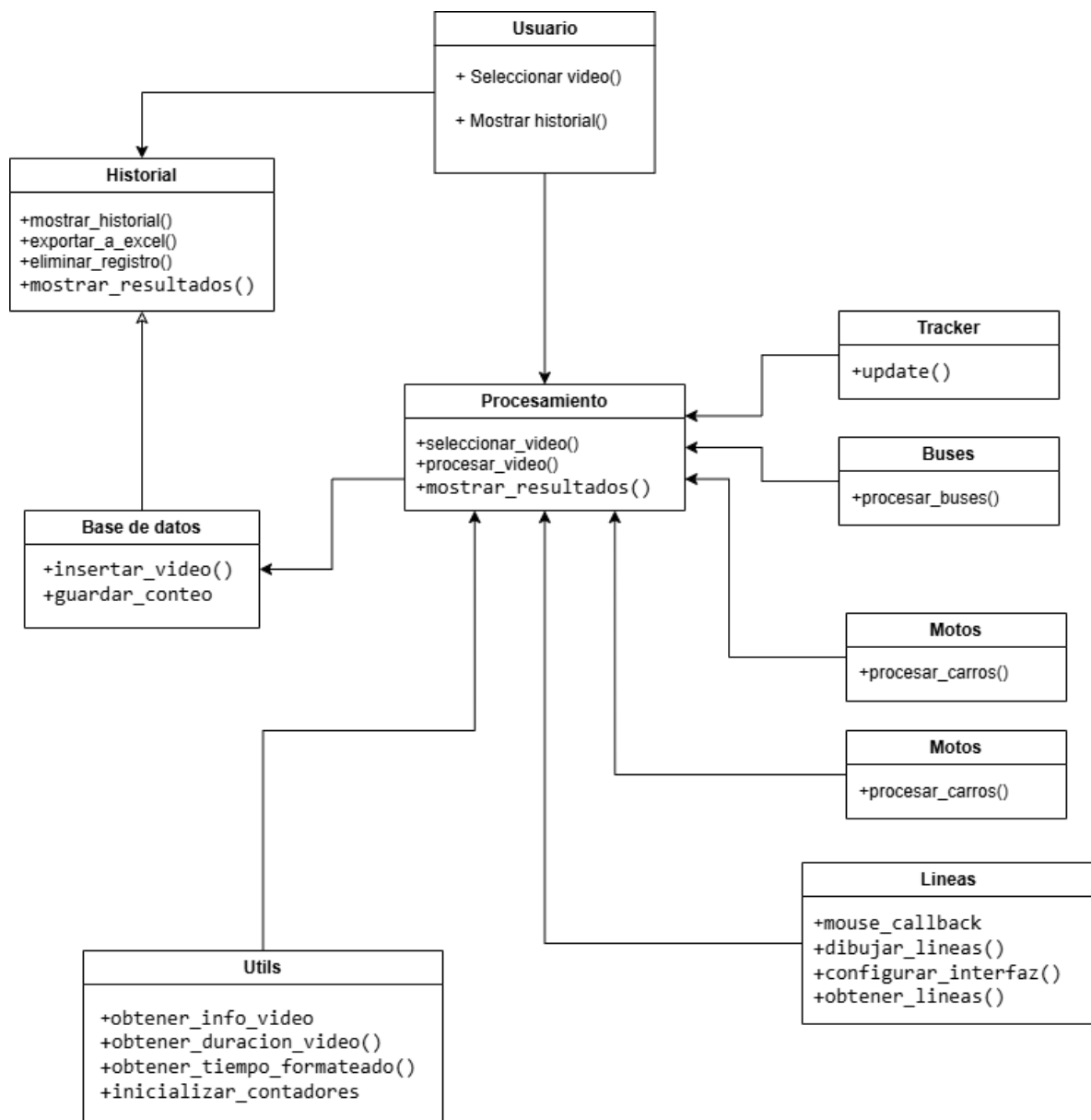
Fuente: Elaboración propia

4. DISEÑO DEL SISTEMA

4.1. Diagrama de Clases

En la Figura 3 se representa el diagrama de clases que define las entidades y relaciones principales del sistema de conteo de vehículos.

Figura 3. DIAGRAMA DE CLASES DEL SISTEMA



Fuente: Elaboración propia.

4.2. Descripción de Clases

La Tabla 2 presenta las clases principales del sistema de conteo vehicular, agrupando aquellas que gestionan el procesamiento de videos, la visualización y el almacenamiento de registros.

Tabla 2. DESCRIPCIÓN DE LAS CLASES PRINCIPALES DEL SISTEMA

Clase	Métodos	Descripción breve
Usuario	seleccionar_video(), mostrar_historial()	Permite cargar videos y mostrar registros anteriores.
Historial	mostrar_historial(), exportar_a_excel(), eliminar_registro(), mostrar_resultados()	Administra la visualización, exportación y eliminación de historiales.
Procesamiento	seleccionar_video(), procesar_video(), mostrar_resultados()	Gestiona el procesamiento de videos y el conteo de vehículos.
Base de datos	insertar_video(), guardar_conteo()	Inserta videos y guarda conteos en la base de datos local.

Fuente: Elaboración propia

La Tabla 3 describe las clases auxiliares que apoyan funciones complementarias en el procesamiento de objetos, la interacción de la interfaz gráfica y el seguimiento de los vehículos detectados.

Tabla 3. DESCRIPCIÓN DE LAS CLASES AUXILIARES DEL SISTEMA

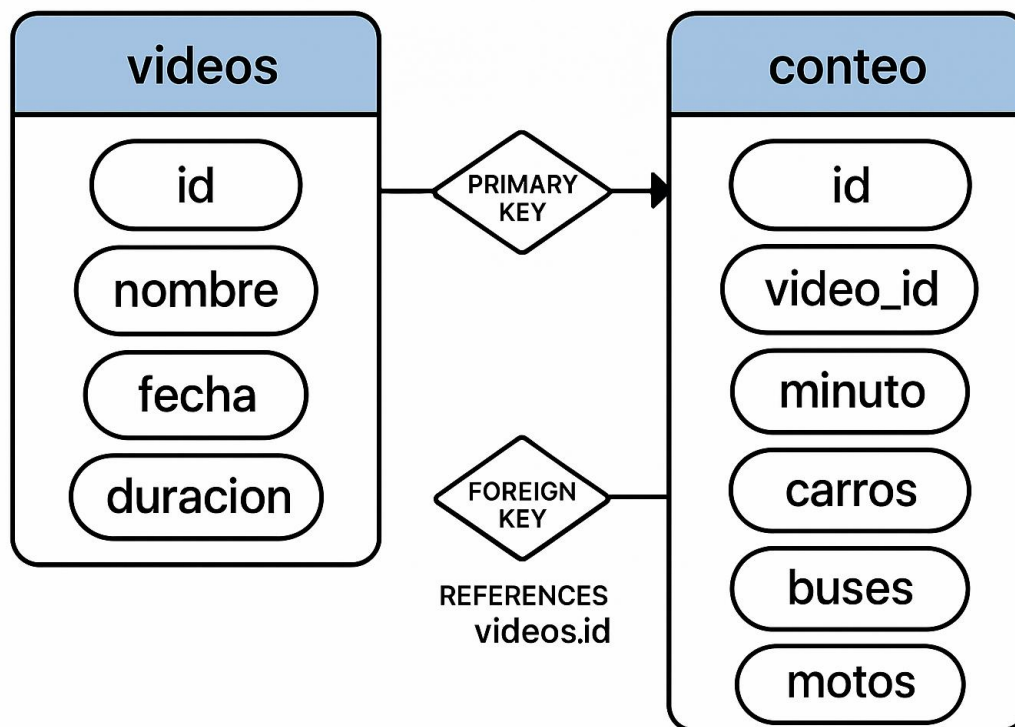
Clase	Métodos	Descripción breve
Utils	obtener_info_video(), obtener_duracion_video(), obtener_tiempo_formateado(), inicializar_contenedores()	Funciones auxiliares para manejo de videos y tiempo.
Tracker	update()	Algoritmo de seguimiento de objetos detectados.
Módulos (buses, carros, motos)	procesar_buses(), procesar_carros(), procesar_motos()	Procesamiento individual por tipo de vehículo.
Líneas	mouse_callback(), dibujar_lineas(), configurar_interfaz(), obtener_lineas()	Manejo de eventos de dibujo e interacción con líneas de conteo.

Fuente: Elaboración propia

4.3. Diagrama Entidad-Relación

La Figura 4 ilustra el diagrama entidad-relación que organiza las tablas de base de datos del sistema y las relaciones entre ellas.

Figura 4. DIAGRAMA ENTIDAD-RELACIÓN DEL SISTEMA



4.4. Diccionario de datos

Las Tablas 4 y 5 describen los campos que componen la base de datos del sistema, tanto para los registros de videos como para los datos de conteo.

Tabla 4. DEFINICIÓN DE CAMPOS DE LA TABLA VIDEOS

Nombre de tabla		VIDEOS	
Nombre del Campo	Tipo de Dato	Clave	Descripción
id	INTEGER	PK	Identificador único del video.
nombre	TEXT	-	Nombre del archivo de video.
fecha	TEXT	-	Fecha en la que se grabó o cargó.

Fuente: Elaboración propia

Tabla 5. DEFINICIÓN DE CAMPOS DE LA TABLA CONTEO

Nombre de tabla	CONTEO		
Nombre del Campo	Tipo de Dato	Clave	Descripción
id	INTEGER	PK	Identificador único del conteo.
video_id	INTEGER	FK → videos.id	Relación con el video al que pertenece el conteo.
minuto	INTEGER	-	Minuto específico del video donde se hace el conteo.
carros	INTEGER	-	Número de carros detectados.
buses	INTEGER	-	Número de buses detectados.
motos	INTEGER	-	Número de motocicletas detectadas.

Fuente: Elaboración propia

5. TECNOLOGÍAS UTILIZADAS

La Tabla 3 enumera las principales tecnologías, librerías y herramientas empleadas para el desarrollo del sistema de conteo de vehículos.

Tabla 6. TECNOLOGÍAS UTILIZADAS EN EL SISTEMA

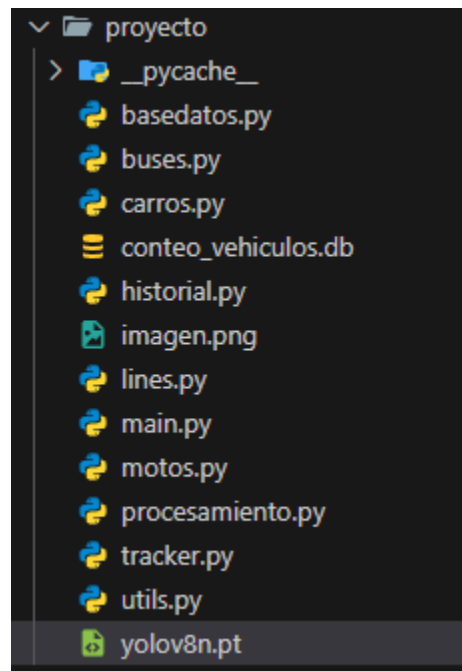
TIPO	NOMBRE	DESCRIPCIÓN
Lenguaje de programación	Python 3.10+	Lenguaje de programación no tipado
Librería	Tkinter	Creación de interfaces
Librería	OpenCb	Procesamientos de video
Librería de Ultralytics	YOLOv8	Librería del modelo de detección de objetos
Librería	Pandas y Openpyxl	Librería para exportar datos y análisis de datos
Motor de base de datos	SQLite3	Base de datos
herramienta de Python	auto-py-to-exe	convierte archivos .py a archivos ejecutable

Fuente: Elaboración propia.

6. ESTRUCTURA DE ARCHIVOS Y CARPETAS

La Figura 5 muestra la estructura de archivos y carpetas del sistema, organizada de manera modular para facilitar el mantenimiento y la gestión eficiente del proyecto.

Figura 5. ESTRUCTURA DE ARCHIVOS Y CARPETAS DEL SISTEMA



Fuente: Elaboración propia.

El sistema de conteo de vehículos está organizado de manera modular, lo cual permite una gestión clara y eficiente del código. A continuación, se describe el propósito de cada archivo y carpeta dentro del proyecto:

- ✓ `__pycache__`/
Carpeta generada automáticamente por Python que contiene versiones compiladas de los archivos .py para acelerar la ejecución del programa.
- ✓ `basedatos.py`
Módulo encargado de gestionar la base de datos SQLite, incluyendo la creación de tablas y el almacenamiento de los registros de conteo.

- ✓ buses.py
Contiene funciones específicas para el conteo y procesamiento de buses detectados en los videos.
- ✓ carros.py
Encargado del procesamiento y conteo de vehículos tipo automóvil (carros).
- ✓ motos.py
Administra el conteo y procesamiento de motocicletas detectadas por el sistema.
- ✓ conteo_vehiculos.db
Archivo de base de datos SQLite que almacena los datos históricos de conteo, clasificados por tipo de vehículo y fecha.
- ✓ historial.py
Módulo que permite visualizar y exportar el historial de conteo por registro guardado en la base de datos.
- ✓ imagen.png
Imagen utilizada posiblemente como logo o fondo dentro de la interfaz gráfica del sistema.
- ✓ lines.py
Gestiona la lógica de las líneas de conteo (por ejemplo, líneas que se dibujan en pantalla para determinar cuándo contar un vehículo).
- ✓ main.py
Archivo principal del sistema. Es el punto de entrada desde donde se inicializa la interfaz gráfica y se ejecutan las funcionalidades del programa.
- ✓ procesamiento.py
Orquesta el procesamiento del video, incluyendo la detección de objetos y la lógica de conteo utilizando el modelo de inteligencia artificial.

- ✓ `tracker.py`

Implementa el seguimiento (tracking) de objetos detectados entre cuadros consecutivos para asegurar que cada vehículo sea contado solo una vez.

- ✓ `utils.py`

Contiene funciones auxiliares y utilitarias que apoyan diferentes partes del sistema, como formatos y conversiones, etc.

- ✓ `yolov8n.pt`

Modelo preentrenado de YOLOv8 (versión nano), utilizado por el sistema para realizar la detección de vehículos a partir del video.

7. CONFIGURACIÓN DEL ENTORNO DE DESARROLLO

7.1. Instalación de Dependencias

La Figura 6 muestra el comando utilizado para instalar las librerías y dependencias necesarias para el correcto funcionamiento del sistema de conteo de vehículos:

Figura 6. COMANDO DE INSTALACIÓN DE DEPENDENCIAS DEL SISTEMA

```
pip install opencv-python pillow pandas openpyxl tk ultralytics
```

Fuente: Elaboración propia

7.2. Flujo de Uso

1. El usuario abre la interfaz.
2. Selecciona un video.
3. Dibuja una línea de conteo con el mouse.
4. El sistema detecta objetos y cuenta los que cruzan la línea.
5. Los datos por minuto se almacenan automáticamente.
6. Desde el historial, se puede exportar o eliminar registros.

7.3. Conteo y Seguimiento

- Se usa YOLOv8 para detectar objetos.
- Se implementa un algoritmo de seguimiento por ID (usando track_id).
- Solo se cuenta un objeto si cruza la línea verde y no ha sido contado previamente.

8. REQUERIMIENTOS MÍNIMOS DE HARDWARE PARA EL PROCESAMIENTO DE IMÁGENES CON YOLOV8

Los mínimos permitirán realizar pruebas y procesamiento no en tiempo real o con resoluciones bajas.

Tabla 7. REQUERIMIENTOS MÍNIMOS DE HARDWARE PARA EL SISTEMA

COMPONENTE	ESPECIFICACIÓN
Procesador (CPU)	- Intel Core i5 de 7ma generación o superior - AMD Ryzen 5 1600 o superior
Memoria RAM	- 8 GB mínimo
Tarjeta Gráfica (GPU)	- NVIDIA GTX 1050 Ti o superior (Soporte CUDA 11+)
Almacenamiento	-Al menos 20 GB libres - Recomendado: SSD para mejor rendimiento
Sistema Operativo	- Windows 10 / 11 -Ubuntu 18.04+ - macOS (solo CPU)

Fuente: Elaboración propia