

# Universidad Tecnológica Nacional

## Proyecto Final

---

# Monitoreo de tránsito vehicular en tiempo real

---

*Autores:*

- MASUTTI, Diego
- PÍGHÍN, Juan Antonio
- SOTO, María Fernanda

*Proyecto final presentado para cumplimentar los requisitos académicos  
para acceder al título de Ingeniero Electrónico*

*en la*

**Facultad Regional Paraná**

Julio de 2017

## **Declaración de autoría:**

Nosotros declaramos que el Proyecto Final “Monitoreo de tránsito vehicular en tiempo real” y el trabajo realizado son propios. Declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero/a Electrónico/a, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

- 
- 
- 

Fecha:

## Agradecimientos:

Agradecemos al actual director de la central de monitoreo de tránsito de la ciudad de Paraná, Entre Ríos, por recibirnos en su oficina y brindarnos información referente al tránsito de la ciudad.

Diego Masutti

Juan Pighín

M. Fernanda Soto

Universidad Tecnológica Nacional

*Abstract*

Facultad Regional Paraná

Ingeniero en Electrónica

**Monitoreo de tránsito vehicular en tiempo real**

MASUTTI, Diego

PIGHÍN, Juan Antonio

SOTO, María Fernanda

**Abstract:**

There was implemented a system capable of identify vehicular flow and possible stops in the transit road in real time. The way of visualize these parameters is through a web application, that also allows to monitoring the transit.

This project was made with two Raspberry Pi 3, one dedicated to install a local server and the other one to make the measurement with a camera and install a remote server.

The device is connected with an Ethernet port and in addition has the possibility to get power through POE. Also, allows WiFi connection.

The result of this project is a product capable of monitoring the transit in real time and count vehicles with a friendly and versatile user interface.

**Keywords:**

Cars counter, vehicles flow.

**Resumen:**

Se implementó un sistema capaz de identificar el flujo vehicular y posibles detenciones ocurridas en la vía de tránsito en tiempo real. La forma de visualizar estos parámetros es mediante una aplicación web, la que además permite monitorear el tránsito.

Todo fue realizado con dos Raspberry Pi 3, una para instalar un servidor local y la otra para realizar la medición con una cámara y montar un servidor remoto. El dispositivo se conecta por un puerto Ethernet y tiene la posibilidad de alimentarse mediante POE. Además, permite la conexión mediante WiFi.

El resultado de este proyecto es un producto capaz de monitorear el tránsito en tiempo real y contabilizar vehículos con una interfaz de usuario amigable y versátil.

**Palabras Clave:**

Contador de autos, flujo de vehículos.

# Índice:

Capítulo 1: Introducción .....	1
Capítulo 2: Desarrollo .....	2
2.1- Desarrollo general .....	2
2.2- Dispositivo de medición .....	3
2.2.1- Estructura .....	3
2.2.1- Investigación de componentes .....	3
2.2.2- Propuestas de circuitos o esquemas.....	5
2.2.3- Desarrollo del software o firmware .....	7
2.2.4- Análisis del funcionamiento .....	23
2.3- Red .....	32
2.4- Unidad de administración central de datos.....	33
2.4.1- Estructura .....	33
2.4.2- Investigación de componentes .....	33
2.4.3- Propuestas de circuitos o esquemas.....	34
2.4.4- Desarrollo del software o firmware .....	34
2.4.5- Análisis del funcionamiento .....	43
2.5- Diseño Completo .....	46
Capítulo 3: Resultados.....	47
Capítulo 4: Análisis de Costos .....	49
Capítulo 5: Discusión y Conclusión.....	50
Capítulo 6: Bibliografía Consultada.....	52

## Lista de Figuras:

Ilustración 1 – Estructura general del sistema .....	2
Ilustración 2 – Estructura del dispositivo de medición .....	3
Ilustración 3 – Raspberry Pi 3.....	4
Ilustración 4 – Módulo de cámara para Raspberry Pi .....	4
Ilustración 5 – Esquema de conexión del dispositivo de medición .....	5
Ilustración 6 – Circuito de conexión del dispositivo de medición .....	6
Ilustración 7 – Fuente Step Down .....	6
Ilustración 8 – Diagrama de bloques del funcionamiento del PDI.....	7
Ilustración 9 – Diagrama de flujo: Inicialización y obtención de parámetros .....	8
Ilustración 10 – Video esperando configuración .....	9
Ilustración 11 – Delimitación de zonas de barrera y de estacionamiento.....	10
Ilustración 12 – Máscara de procesamiento .....	11
Ilustración 13 – Máscara de posicionamiento.....	11
Ilustración 14 – Máscara de estacionamiento.....	11
Ilustración 15 – Diagrama de flujo: Detección de movimiento .....	13
Ilustración 16 – Lectura de un frame .....	14
Ilustración 17 – Conversión a escala de grises.....	14
Ilustración 18 – Suavizado de imagen y disminución de ruido.....	15
Ilustración 19 – Sustracción de fondo.....	15
Ilustración 20 – Umbralización.....	15
Ilustración 21 – Dilatación de bordes.....	16
Ilustración 22 – Aplicación de máscara de procesamiento .....	16
Ilustración 23 – Detección y filtrado de contornos .....	17
Ilustración 24 – Representación de los objetos en movimiento .....	17
Ilustración 25 – Diagrama de flujo: Seguimiento y conteo de vehículos .....	19
Ilustración 26 – Flujo óptico .....	20
Ilustración 27 – Diagrama de flujo: Generación de alarmas por flujo.....	22

Ilustración 28 – Diagrama de flujo: Generación de alarmas por detención.....	23
Ilustración 29 – Aplicación web del dispositivo: página principal .....	25
Ilustración 30 – Aplicación web del dispositivo: sección de Monitoreo – Alarmas sin atender .....	26
Ilustración 31 – Aplicación web del dispositivo: sección de Monitoreo – Capturas .....	26
Ilustración 32 – Aplicación web del dispositivo: sección de Históricos – Vehículos .....	27
Ilustración 33 – Aplicación web del dispositivo: consulta tabla vehículos .....	27
Ilustración 34 – Aplicación web del dispositivo: consulta gráfico 1 vehículos .....	28
Ilustración 35 – Aplicación web del dispositivo: consulta gráfico 2 vehículos .....	28
Ilustración 36 – Aplicación web del dispositivo: sección de Históricos – Alarmas.....	29
Ilustración 37 – Aplicación web del dispositivo: consulta tabla alarmas.....	29
Ilustración 38 – Aplicación web del dispositivo: consulta gráfico 1 alarmas.....	30
Ilustración 39 – Aplicación web del dispositivo: consulta gráfico 2 alarmas.....	30
Ilustración 40 – Aplicación web del dispositivo: sección de Configuración de parámetros	31
Ilustración 41 – Aplicación web del dispositivo: sección de Configuración de red .....	31
Ilustración 42 – Esquema aproximado de conexión del sistema a la red .....	32
Ilustración 43 – Estructura de la unidad de administración central de datos .....	33
Ilustración 44 – Esquema de conexión del servidor local .....	34
Ilustración 45 – Servidor Local: Monitoreo Proactivo .....	35
Ilustración 46 – Aplicación web del servidor local: sección de Monitoreo - Cámaras .....	35
Ilustración 47 – Aplicación web del servidor local: sección de Monitoreo – Históricos .....	36
Ilustración 48 – Aplicación web del servidor local: sección de Monitoreo – Estadísticas, acceso remoto .....	36
Ilustración 49 – Aplicación web del servidor local: sección de Monitoreo – Históricos, acceso remoto, gráfico. ....	37
Ilustración 50 – Aplicación web del servidor local: sección de Monitoreo – Estado .....	37
Ilustración 51 – Aplicación web del servidor local: sección de Monitoreo – Alarmas .....	38
Ilustración 52 – Aplicación web del servidor local: sección de Monitoreo – Alarmas, acceso remoto .....	38
Ilustración 53 – Aplicación web del servidor local: sección de Administrar - Cámaras .....	39



Ilustración 54 – Aplicación web del servidor local: sección de Administrar – Cámaras, editar.....	40
Ilustración 55 – Aplicación web del servidor local: sección de Administrar - Usuarios.....	41
Ilustración 56 – Aplicación web del servidor local: sección de Administrar – Usuarios, editar.....	41
Ilustración 57 – Aplicación web del servidor local: sección de Configuración - Monitoreo.	42
Ilustración 58 – Aplicación web del servidor local: sección de Configuración - Red .....	43
Ilustración 59 – Dispositivo de medición - Hardware .....	46

## Lista de Tablas

Tabla 1 – Base de Datos Remota - Vehículos.....	24
Tabla 2 – Base de Datos Remota - Alarmas .....	24
Tabla 3 – Base de Datos Remota - Variables .....	24
Tabla 4 – Base de Datos Local - Usuarios .....	44
Tabla 5 – Base de Datos Local - Cámaras.....	44
Tabla 6 – Resultados de mediciones .....	48
Tabla 7 – Costos de recursos materiales .....	49
Tabla 8 – Costos de recursos humanos.....	49

## **Lista de Abreviaciones**

CSI – Camera Serial Interface (Interfaz Serie para Cámaras)

Html – HyperText Markup Language (Lenguaje de marcas de hipertexto)

IP – Internet Protocol (Protocolo de internet)

LAN – Local Area Network (Red de área local)

PDI – Procesamiento digital de imágenes

POE – Power Over Ethernet (Alimentación a través de Ethernet)

## **Dedicado a:**

*Queremos dedicar este proyecto especialmente a nuestras familias que nos han acompañado durante toda la carrera y a la institución por brindarnos la educación y enseñanza tanto profesional como personal.*

## Capítulo 1: Introducción

Durante la última década ha habido un constante incremento de la cantidad de vehículos y con esto un aumento del tráfico. El congestionamiento generado trae efectos negativos en el medio ambiente y en los ciudadanos, genera accidentes de tránsito, contaminación acústica y también estrés en los ciudadanos. La cantidad de tráfico en un cruce de calles y su distribución varía en relación a factores como el horario, el día de la semana, la época de año, cortes de calles por manifestación o mantenimiento, etc.

Para obtener un control de estos factores y dar una solución, surgió la idea de implementar un sistema de cámaras de medición automática del tránsito que sean controladas por personal en las centrales de monitoreo y que permita obtener información en tiempo real y generar estadísticas.

El dispositivo se ubicará junto al semáforo y permitirá visualizar en una página web las imágenes en tiempo real de la vía de tránsito y distintos indicadores, tanto la cantidad de vehículos que están circulando como otros factores deseados por la central, por ejemplo, la velocidad, detenciones por largo tiempo, entre otros.

Esto permitirá dos tipos de controles. Un control en tiempo real que servirá para conocer el estado actual y, de ser necesario, realizar ajustes en el sistema de tránsito en el momento. Un control histórico que permitirá generar una base de datos de la cual se tomen estadísticas que posibiliten una futura mejora del sistema actual.

El dispositivo funciona mediante la toma de imágenes a través de una cámara y el procesamiento de las mismas para luego visualizarlas en la web y mostrar y almacenar los datos obtenidos.

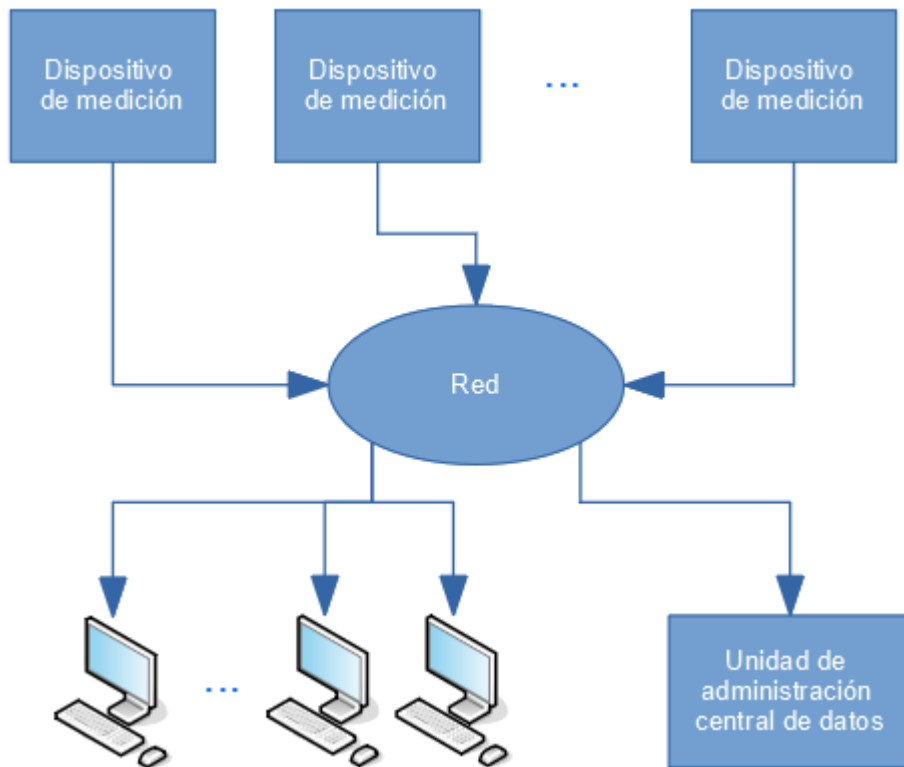
Lo que agrega valor a este producto, es que la información puede ser accedida por el usuario sin importar el dispositivo que esté utilizando y lo libera de instalaciones de software y compatibilidades de sistemas operativos, proporcionando mayor comodidad y sencillez a la hora de adquirirlo.

El producto está dirigido a centrales de tránsito municipales, policiales y particulares, con el objetivo de brindar un complemento indispensable para el sistema de control de las grandes ciudades, beneficiando indirectamente a todos los ciudadanos que circulen en transporte público o privado.

## Capítulo 2: Desarrollo

### 2.1- Desarrollo general

Como elemento de partida, se muestra a continuación el diagrama de bloques general del proyecto:



*Ilustración 1 – Estructura general del sistema*

Este proyecto consta de 3 elementos básicos: el dispositivo de medición, la red y la unidad de administración central de datos. Cabe aclarar que se ha diseñado suponiendo una red ya existente, por lo que, en esta ocasión, se han desarrollado los otros dos elementos y se explicará el funcionamiento y características de cada uno de ellos de forma detallada en las siguientes secciones.

## 2.2- Dispositivo de medición

### 2.2.1- Estructura

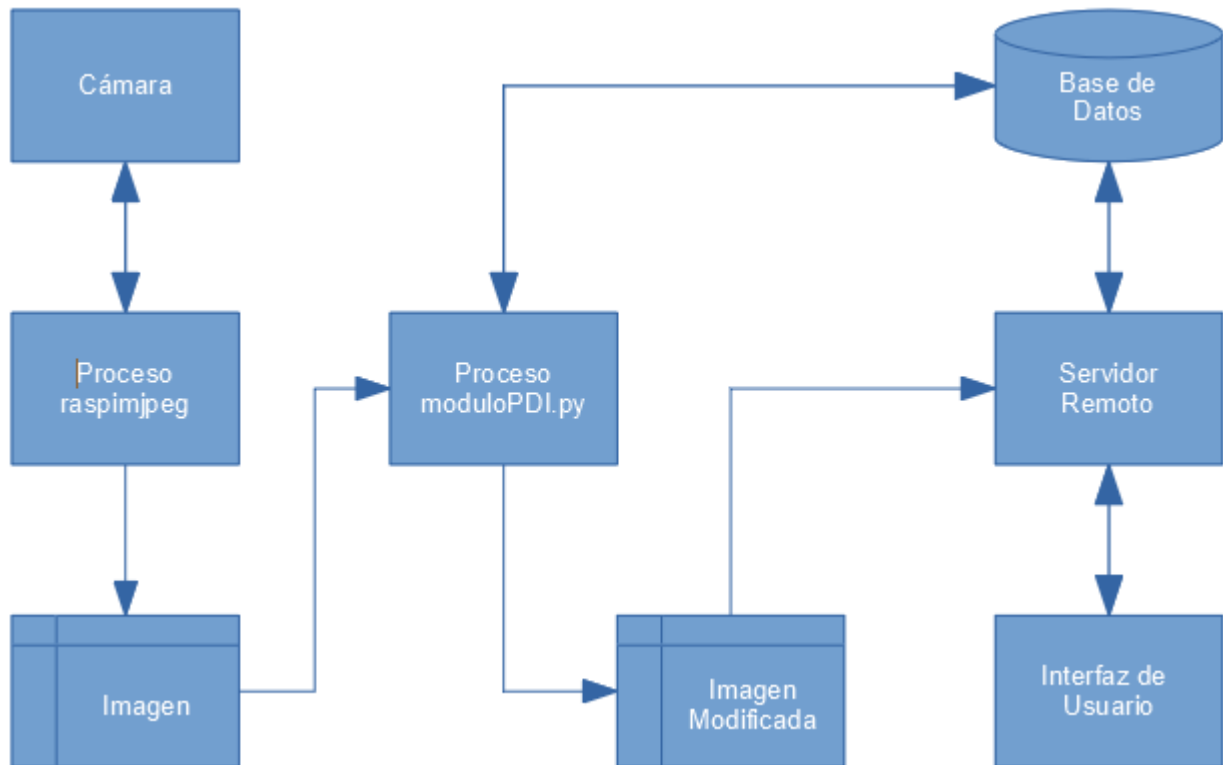


Ilustración 2 – Estructura del dispositivo de medición

### 2.2.1- Investigación de componentes

Las características de este proyecto requieren de un dispositivo no sólo capaz de realizar un gran procesamiento sino también de permitir crear un servidor. Por tal motivo, se evaluaron dos posibles opciones que permitirían cumplir con estas necesidades de una forma sencilla e integrada: *Raspberry Pi* y *BeagleBone Black*.

Se seleccionó la *Raspberry Pi 3* por ofrecer mayor velocidad de procesamiento, mayor memoria RAM, conectividad WiFi incorporada y menor precio

Estas son todas las características del modelo:

- Procesador de 1.2GHz 64-bit quad-core ARMv8
- 802.11n *Wireless LAN*
- *Bluetooth 4.1*
- *Bluetooth Low Energy (BLE)*
- 1GB RAM
- 4 puertos USB

- 40 pines GPIO
- Puerto *Full HDMI* (interfaz multimedia de alta definición)
- Puerto *Ethernet*
- Entrada de audio y video
- Cámara interfaz
- Display interfaz
- Ranura para tarjeta *Micro SD*
- Gráficos VideoCore IV 3D

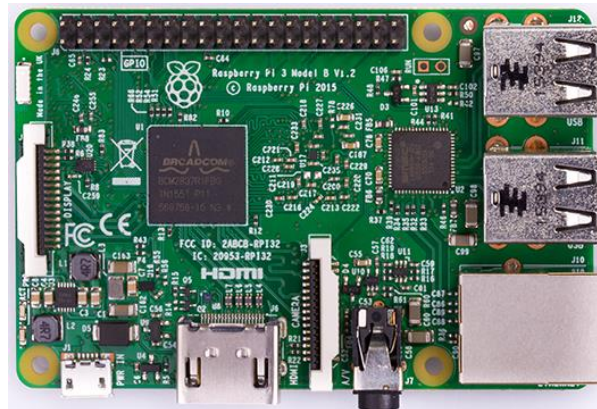


Ilustración 3 – Raspberry Pi 3<sup>1</sup>

En este caso, no han sido utilizadas todas las prestaciones ya que no fueron necesarias.

Como sensor de imagen se ha seleccionado un módulo de cámara compatible con este dispositivo, *Camera V2.1*.

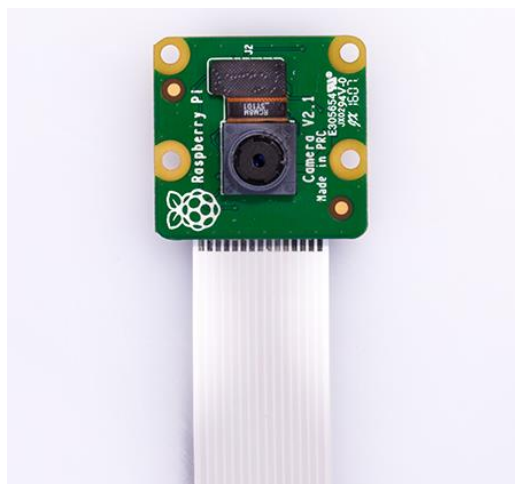


Ilustración 4 – Módulo de cámara para Raspberry Pi<sup>2</sup>

<sup>1</sup> Fuente: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

<sup>2</sup> Fuente: <https://www.raspberrypi.org/products/camera-module-v2/>



Primero se comenzó por instalar el sistema operativo *Raspbian* en la Raspberry, luego se instaló un servidor *Apache II* con un módulo compatible con el lenguaje de programación *PHP* y una base de datos *MySQL*. Además, para realizar pruebas y administrar la base de datos desde la web, se descargó la herramienta *phpmyadmin*.

Para la visualización y obtención de la imagen desde la cámara, se ha partido de una interfaz web de acceso libre, "*RPi Cam Web Interface*", desarrollada para el módulo Pi Camera. Esta interfaz instala un proceso en la Raspberry que se encarga de tomar la imagen de la cámara para luego almacenarla en un directorio. A partir de ésta, se ha desarrollado el procesamiento de imagen y la aplicación web *TrafficVC*.

Respecto a la alimentación del dispositivo, se investigó la forma en la que se conectan las cámaras IP en la ciudad, y por lo general se utiliza *Power Over Ethernet* (POE). Por este motivo, se da esta opción como alimentación del dispositivo.

### 2.2.2- Propuestas de circuitos o esquemas

El dispositivo puede alimentarse mediante POE. El cable de red más alimentación ingresa a un divisor de POE que separa *Ethernet* para conectar directamente la *Raspberry* y alimentación con 5V y GND que pasan luego por un selector.

En el caso de que no se tenga una red con este tipo de alimentación incluida, se puede conectarlo al voltaje de línea de 220V. Para esto, se utiliza una fuente reguladora de voltaje *Step Down* que provee 5V y GND, los cuales ingresan al selector Fuente/POE.

El esquema de conexión entre los componentes es el siguiente:

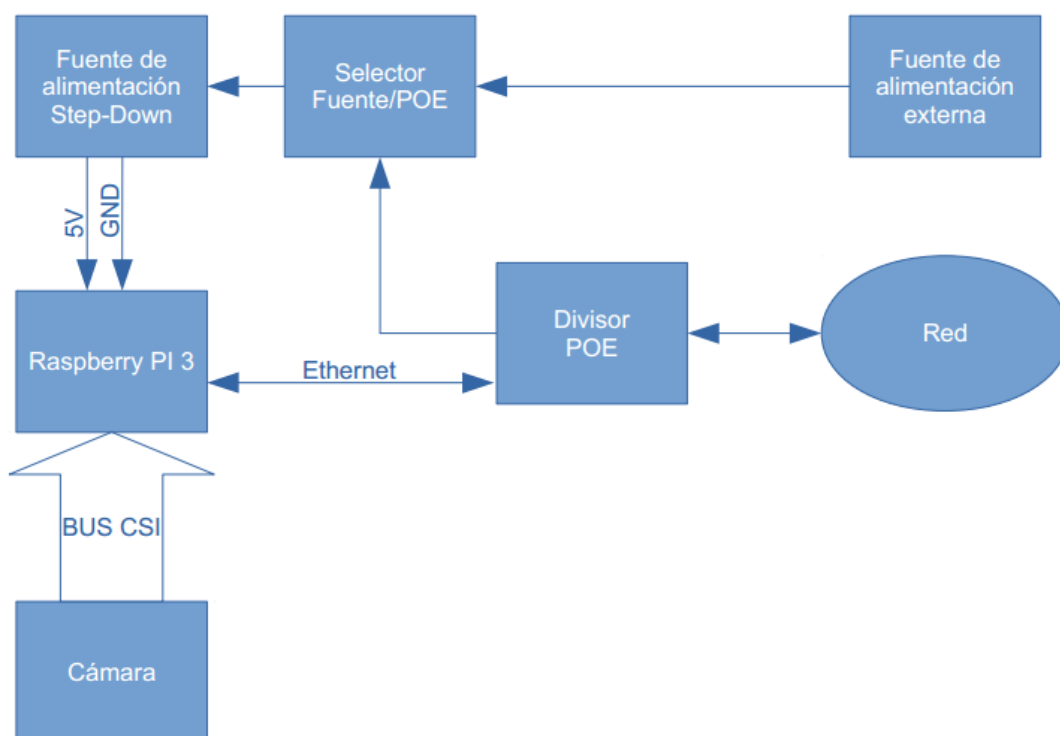


Ilustración 5 – Esquema de conexión del dispositivo de medición

En el siguiente circuito se muestra la entrada de POE (RJ1) y cómo se divide en Ethernet (pines 1, 2, 3 y 6) y alimentación (pines D1 y GND). La alimentación externa ingresa al circuito a través de J2 (D2 y GND). Las tensiones D1 y D2 pasan por el selector compuesto por dos diodos. Uno de estos voltajes junto con el filtro (formado por R1, C1 y C2) conforman la alimentación de la *Raspberry Pi*.

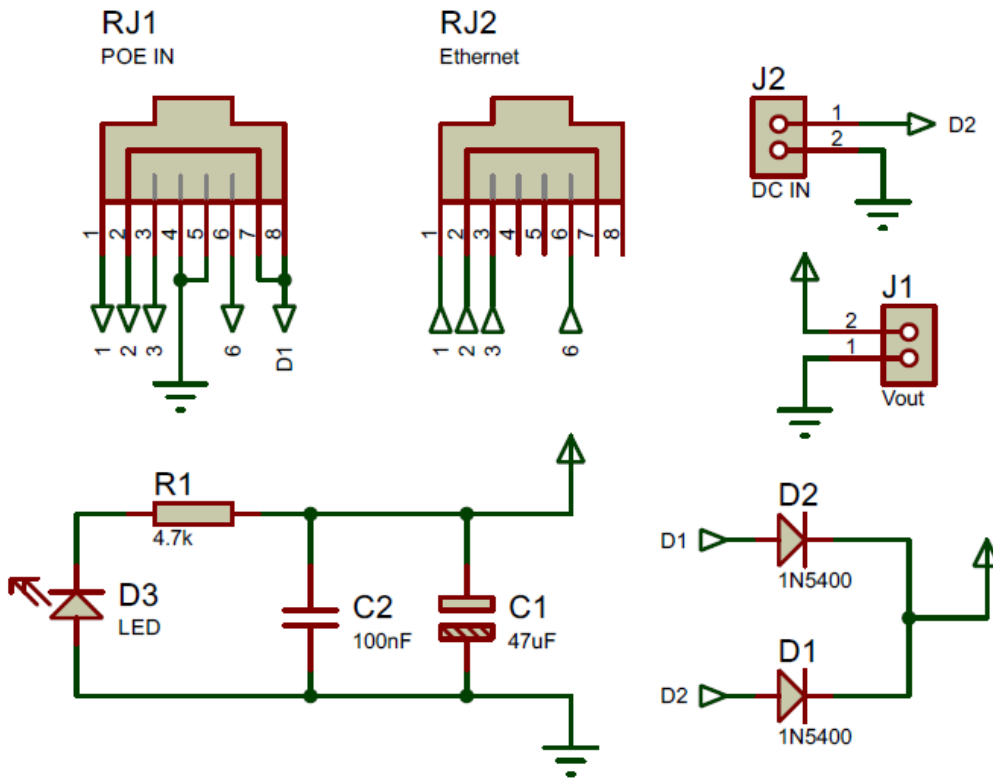


Ilustración 6 – Circuito de conexión del dispositivo de medición



Ilustración 7 – Fuente Step Down <sup>3</sup>

<sup>3</sup> Fuente: [https://mla-s2-p.mlstatic.com/716311-MLA20531293840\\_122015-C.jpg](https://mla-s2-p.mlstatic.com/716311-MLA20531293840_122015-C.jpg)

El dispositivo se ubicará, preferentemente, junto al semáforo de la vía de tránsito que se desee monitorear, la instalación del mismo debe cumplir con las características que se detallan a continuación. La orientación debe ser paralela a la calle, es decir al flujo vehicular. A su vez, debe tener una inclinación de 45° con la cámara apuntando hacia el suelo.

### 2.2.3- Desarrollo del software o firmware

El procesamiento digital de imágenes es llevado a cabo por el módulo “moduloPDI”, el cual está implementado bajo el lenguaje de programación *Python* y utilizando funciones de la librería *OpenCV*.

Este módulo, es el encargado de tomar los *frames* del video en tiempo real, generados por el módulo “raspimjpeg”, y procesarlos con el fin de contar vehículos y establecer alarmas.

El funcionamiento de este módulo puede ser dividido en los siguientes cuatro bloques:

1. Inicialización y obtención de parámetros.
2. Detección de movimiento.
3. Seguimiento y conteo de vehículos.
4. Generación de alarmas.

Se puede ver el diagrama de bloques del funcionamiento en la siguiente ilustración. A continuación, se explicará detalladamente cada uno de ellos

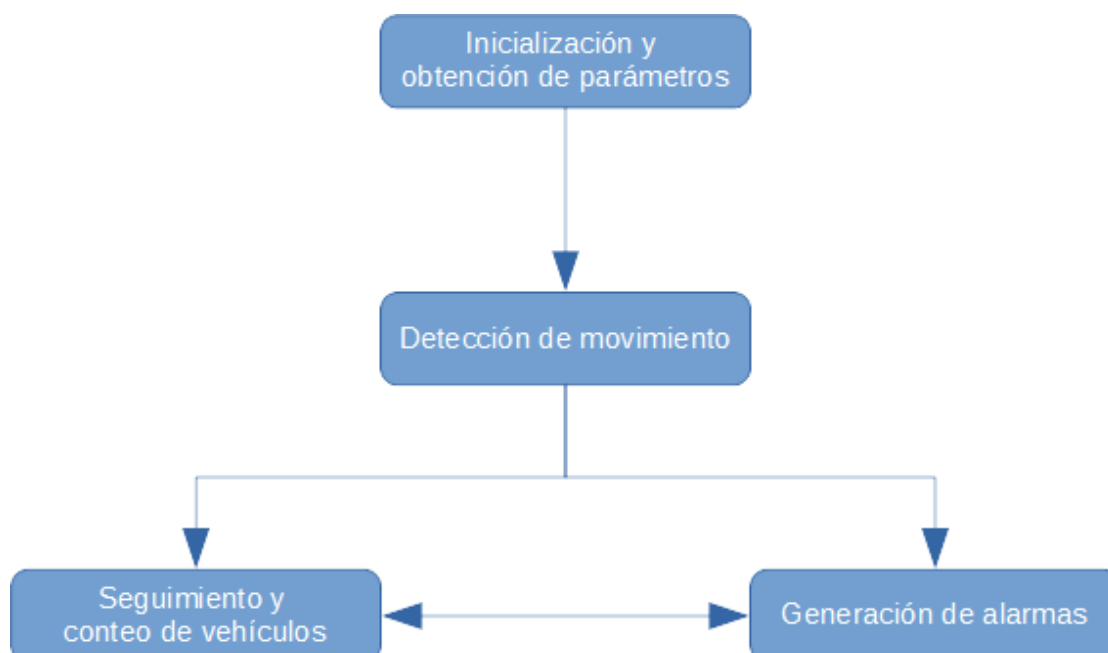


Ilustración 8 – Diagrama de bloques del funcionamiento del PDI

2.2.3.1- Inicialización y obtención de parámetros

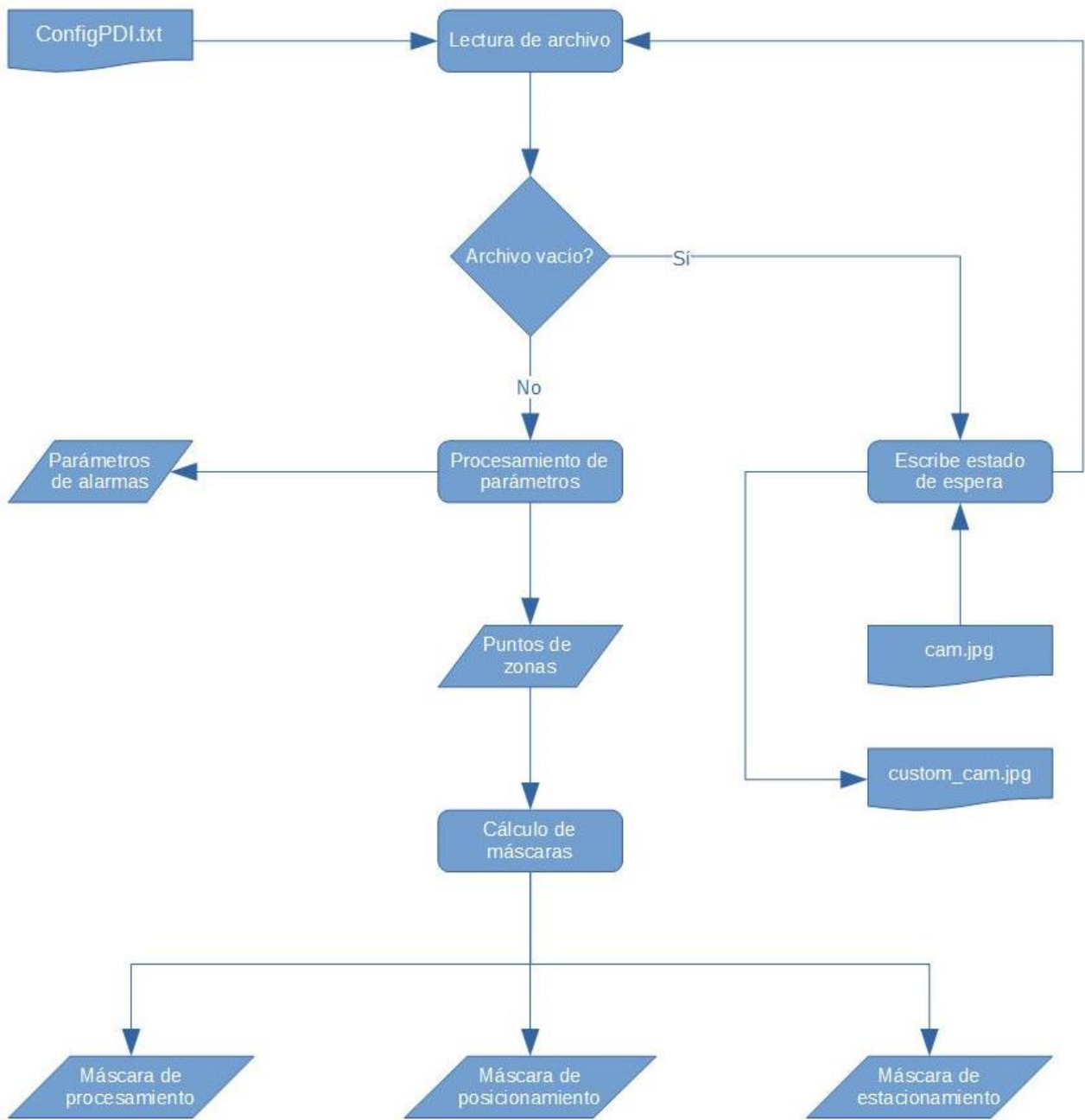


Ilustración 9 – Diagrama de flujo: Inicialización y obtención de parámetros

Cuando el “moduloPDI.py” es lanzado, como primera acción busca el archivo de configuración “ConfigPDI.txt” e intenta leerlo. Si el archivo no existe o se encuentra vacío, indica que la configuración no ha sido establecida y el mismo genera un video de salida, el cual consiste en superponer sobre el video de la cámara una inscripción que indica que la configuración no ha sido cargada; continúa así, hasta que se revierta esta situación.



Ilustración 10 – Video esperando configuración

Si puede leer el archivo, el módulo obtiene los siguientes parámetros:

- Puntos de barrera: Se trata de cuatro puntos con sus correspondientes coordenadas dentro de la imagen, los cuales definen la “*barrera de cruce*”. Esta barrera cumple con dos funciones fundamentales dentro del procesamiento de imágenes: establecer un offset de conteo para ayudar a disminuir el error producido por pequeñas oscilaciones en los puntos de seguimiento, y marcar la perspectiva de la calle respecto a la posición de la cámara para poder identificar correctamente la zona de procesamiento dentro de la imagen.
- Puntos de zona de estacionamiento: Estos puntos pueden estar incluidos o no, dependiendo de si el usuario estableció un sector de estacionamiento o no. Al igual que en el caso anterior, se trata de cuatro puntos que delimitan una zona dentro de la imagen, en la cual, al tratarse de un sector de estacionamiento, las alarmas por detención no tendrán efecto.
- Flujos máximos: Son dos valores numéricos ingresados por el usuario y expresados en vehículos por minutos (veh/minutos), éstos definen el flujo máximo para el cual el módulo debe establecer las alarmas por flujo. El valor más pequeño es para la alarma de advertencia de flujo elevado, mientras que el otro define la alarma de flujo crítico.
- Tiempos de detención máximos: También se trata de dos valores ingresados por el usuario, expresados en minutos, que sirven para definir las alarmas por detención. Uno de los tiempos define cuánto un auto debe estar detenido para establecer una alarma de advertencia, mientras que el otro es un valor que define una alarma crítica de detención.

Luego de obtener estos parámetros, el módulo almacena en variables locales los parámetros que definen alarmas y utiliza los puntos de barrera y zona de estacionamiento

para generar tres máscaras de filtrado. Estas máscaras poseen las mismas dimensiones que los *frames* de video y son utilizadas en las subsiguientes etapas del procesamiento.

La primera de estas es la que se denominó como “*máscara de procesamiento*”, creada a partir de los puntos de barrera. Se trata de una imagen binaria en la cual los píxeles blancos representan la zona en la que se realizará el procesamiento. Su función principal es limitar la detección de movimientos en la región de la imagen comprendida por la vía de tránsito.

La segunda es la “*máscara de posicionamiento*”, geoméricamente es idéntica a la anterior con la diferencia de que la zona de procesamiento ahora se encuentra dividida en tres subzonas, a éstas se les asignan distintos colores grises intermedios para diferenciarlas. Estas subzonas corresponden a las regiones comprendidas detrás, dentro y delante de la “barrera”, respecto a la vía de tránsito. Su función en el módulo de procesamiento es indicar de manera sencilla y eficiente la posición de los puntos de interés dentro de la imagen, para esto sólo deben compararse las coordenadas del punto en cuestión con las de la máscara y, dependiendo el valor que se obtenga, se determina en qué zona de la imagen se encuentra dicho punto. Esta técnica permite ahorrar cálculos en tiempo real, mejorando notablemente el desempeño del módulo.

La última es la “*máscara de estacionamiento*” que es una imagen binaria, al igual que la primera, cuyos píxeles blancos corresponden al sector donde los vehículos pueden detenerse, sin que se active ninguna de las alarmas por detención. Su empleo es similar al de la segunda, dado que antes de procesarse algún punto para conocer su estado de movilidad, es comparado con el valor homónimo dentro de la máscara y en el caso de obtener un pixel blanco como resultado, este punto no es procesado.

Para una mejor comprensión de lo anteriormente descrito, se tomará como referencia la siguiente imagen de configuración y se aplicarán las máscaras. En ella, el recuadro rojo representa la zona de barrera y el verde la de estacionamiento.

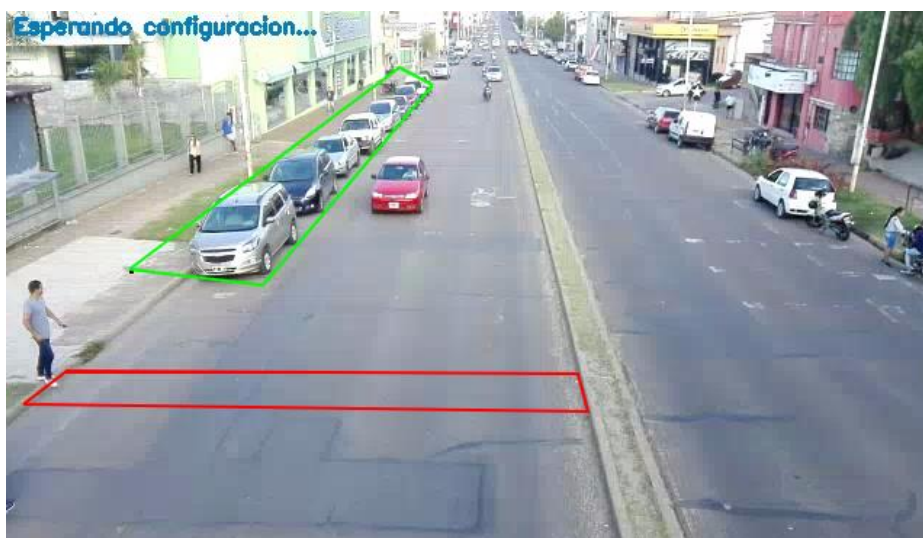
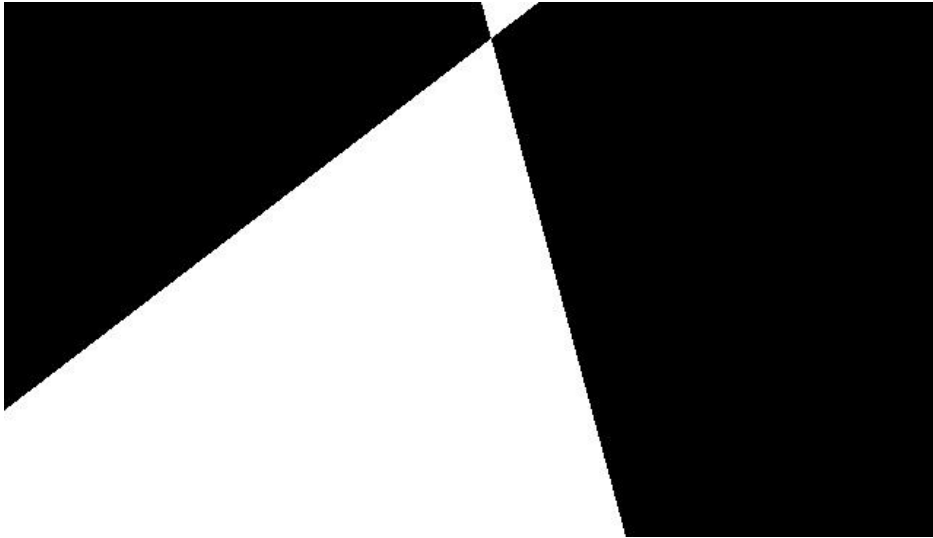
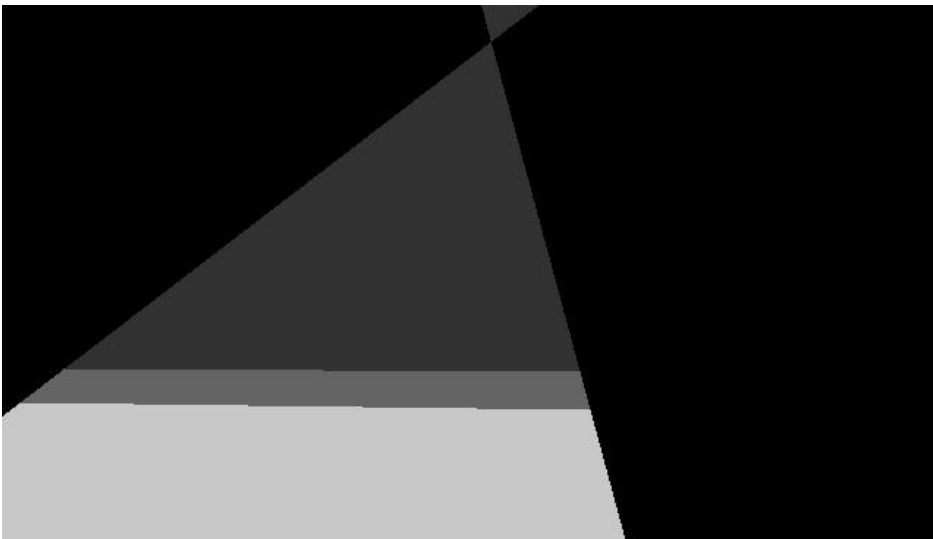


Ilustración 11 – Delimitación de zonas de barrera y de estacionamiento



*Ilustración 12 – Máscara de procesamiento*



*Ilustración 13 – Máscara de posicionamiento*



*Ilustración 14 – Máscara de estacionamiento*

### 2.2.3.2- Detección de movimiento

La detección de movimiento consiste en determinar, para cada píxel, si hubo o no un desplazamiento del mismo, comparando el *frame* actual con el inmediatamente anterior, al que definimos como “fondo”. En otras palabras, el módulo obtiene una imagen binaria donde los píxeles blancos representen movimiento y los negros reposo.

En el diagrama de flujos correspondiente puede observarse, de manera general, el funcionamiento de dicho proceso.

La detección de movimiento consta de las siguientes etapas:

#### 1. Lectura de un *frame*:

Se lee el *frame* que es obtenido por el módulo “raspimjpeg” y establecido como una imagen de nombre “cam.jpg”.

#### 2. Conversión a escala de grises:

El *frame* es convertido a escala de grises, dado que resulta óptimo y menos complejo trabajar con este tipo de imágenes.

#### 3. Suavizado de imagen y disminución de ruido:

Este proceso se lleva a cabo a través de un filtro de promediado, el cual desenfoca la imagen y ataca al ruido gaussiano, siendo éste el más preponderante en las imágenes digitales. El ruido tiene su origen en diferencias de ganancias del sensor, en la digitalización, etc.

#### 4. Sustracción de fondo:

Se realiza la resta píxel a píxel entre la imagen filtrada y la imagen establecida como fondo. Cabe aclarar que al fondo se le aplicó el mismo proceso de conversión y filtrado que a la imagen actual, haciendo que ambas sean totalmente compatibles.

Como resultado de esta acción, se obtiene una imagen cuyos píxeles representan la diferencia entre el *frame* actual y el fondo. Esta diferencia está dada en su mayoría por el movimiento de objetos y en menor medida por ruido.

#### 5. Umbralización y dilatación de bordes:

Con el objetivo de obtener la imagen binaria anteriormente mencionada, se realiza una umbralización de los píxeles de la imagen diferencia, con un valor umbral conveniente.

Además, se aplica una dilatación de bordes para fusionar zonas que estén separadas por unos pocos píxeles, y así reducir al mínimo la incidencia del ruido.

#### 6. Aplicación de máscara de procesamiento:

En este paso, se aplica la máscara de procesamiento obtenida en el proceso anterior. Con ésta se eliminan las zonas de movimiento que ocurren fuera del área de procesamiento.



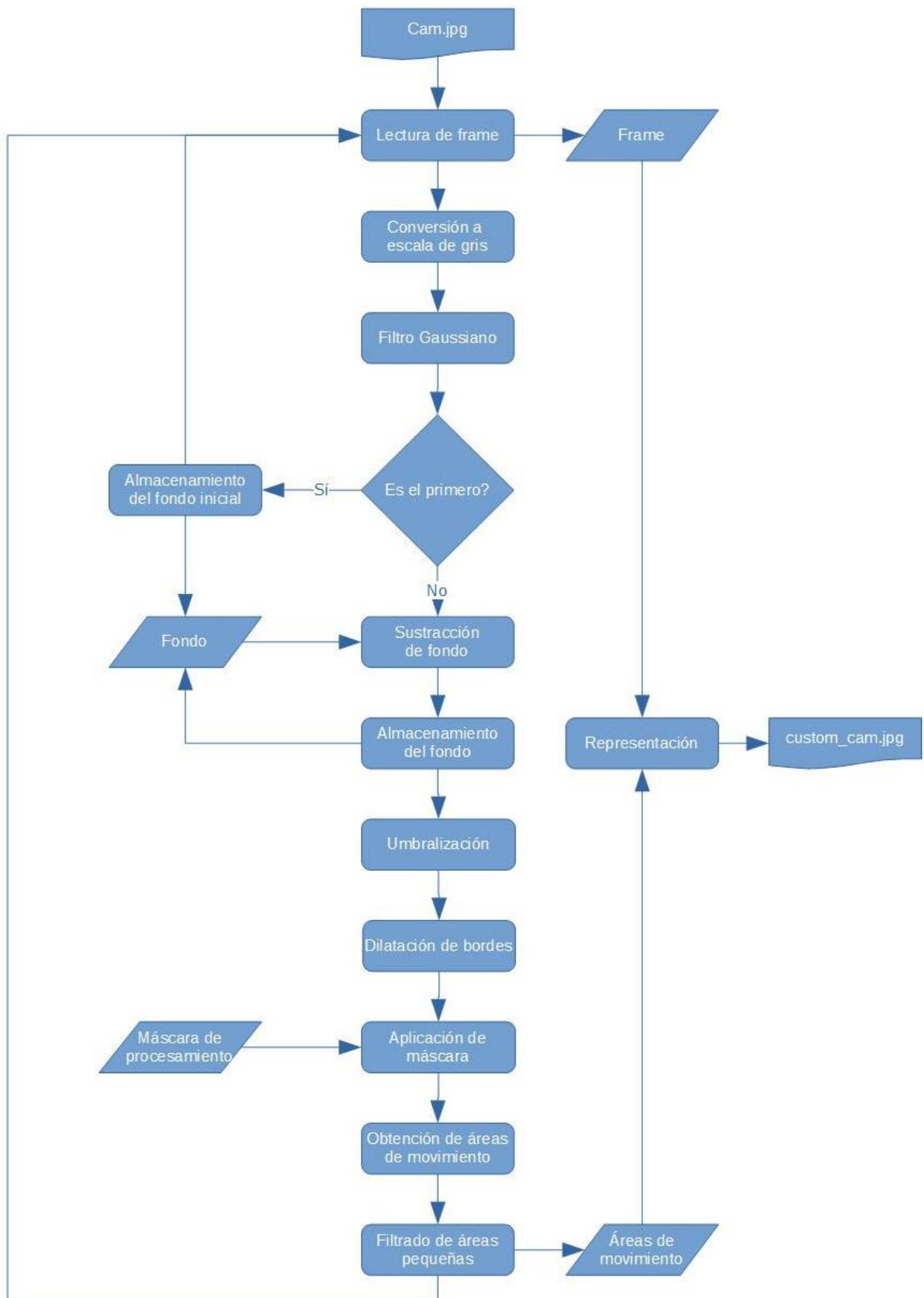


Ilustración 15 – Diagrama de flujo: Detección de movimiento

### 7. Detección y filtrado de contornos:

Una vez que se tiene la imagen con píxeles blancos y negros, el módulo procede a detectar los contornos o *blobs*. Un *blob* es un conjunto de píxeles que están conectados entre sí; es decir, tiene píxeles vecinos con el mismo valor. Con el fin de procesar sólo objetos de tamaño relativamente grande, como pueden ser vehículos, todos los *blobs* que no cumplen con un determinado tamaño mínimo son descartados.

### 8. Representación de los objetos en movimiento:

Basándose en los *blobs* obtenidos en el punto anterior, el módulo grafica sobre el *frame* original un rectángulo con un tamaño y posición de modo tal que encierre al objeto en movimiento identificado.

A continuación, se muestran los pasos descritos para una imagen tomada como ejemplo.



Ilustración 16 – Lectura de un frame



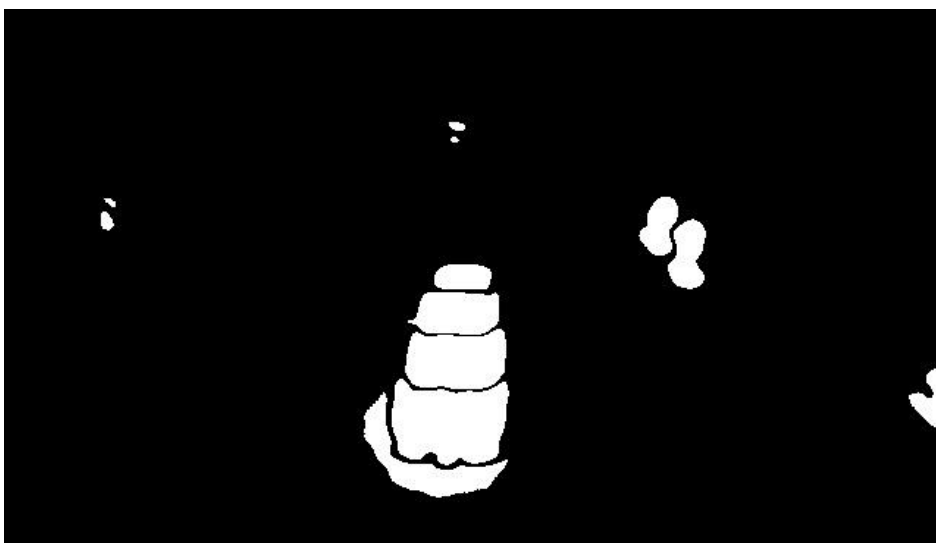
Ilustración 17 – Conversión a escala de grises



*Ilustración 18 – Suavizado de imagen y disminución de ruido*



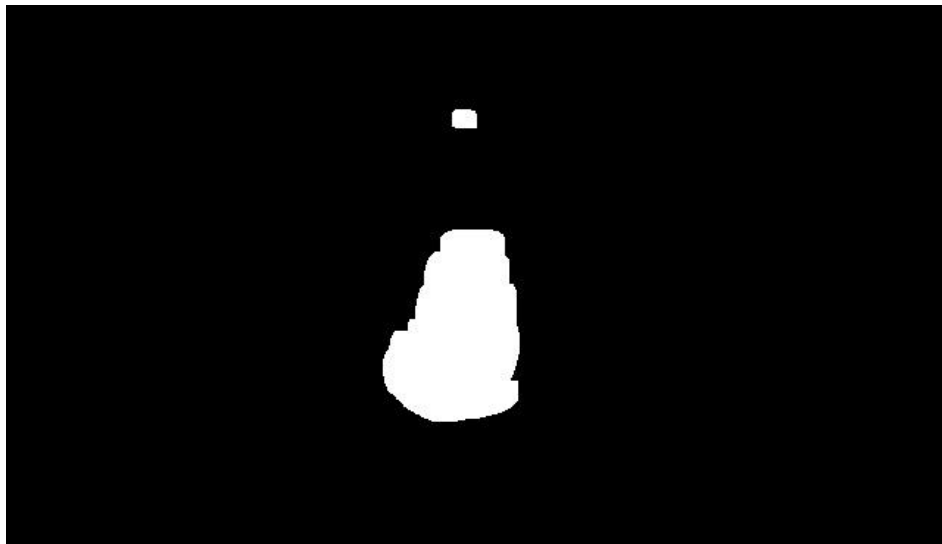
*Ilustración 19 – Sustracción de fondo*



*Ilustración 20 – Umbralización*



*Ilustración 21 – Dilatación de bordes*



*Ilustración 22 – Aplicación de máscara de procesamiento*

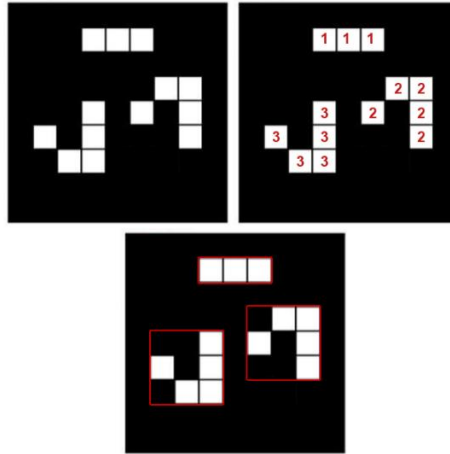


Ilustración 23 – Detección y filtrado de contornos<sup>4</sup>



Ilustración 24 – Representación de los objetos en movimiento

De esta forma, se ha identificado el vehículo en movimiento; esto es importante para luego poder obtener la cantidad que están circulando.

### 2.2.3.3- Seguimiento y conteo de vehículos

Se muestra el diagrama de flujo correspondiente.

Una vez individualizadas las zonas de movimiento correspondientes a vehículos circulantes por la vía de sensado, se le asigna un punto de seguimiento a cada una de ellas. Este punto es establecido como la posición del píxel central de cada área y cumplirá la función de conocer la posición de un vehículo en todo momento, aun así si éste se encuentra detenido.

<sup>4</sup> Fuente: <https://programarfacil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/>

En otras palabras, cada punto representa un vehículo dentro de la imagen que fue identificado inicialmente por la detección del movimiento del mismo. Cada vez que se detecte una zona de movimiento, que no posea punto asignado, se agrega un elemento a este conjunto de puntos de seguimiento, y se infiere que un nuevo vehículo ha aparecido dentro de la imagen.

Estos puntos son seguidos dentro de la imagen desde que son creados hasta que salen de la zona de medición. Esto es llevado a cabo mediante el cálculo del flujo óptico de los puntos de seguimiento. El flujo óptico es el patrón de movimiento aparente de los objetos, en este caso de los píxeles, entre dos cuadros consecutivos de una secuencia fotográfica, causado por el movimiento de los mismos.

Conociendo el cuadro actual, el inmediatamente anterior y la posición de los puntos en éste, el módulo calcula el flujo óptico y obtiene la posición en tiempo real de cada punto de seguimiento.

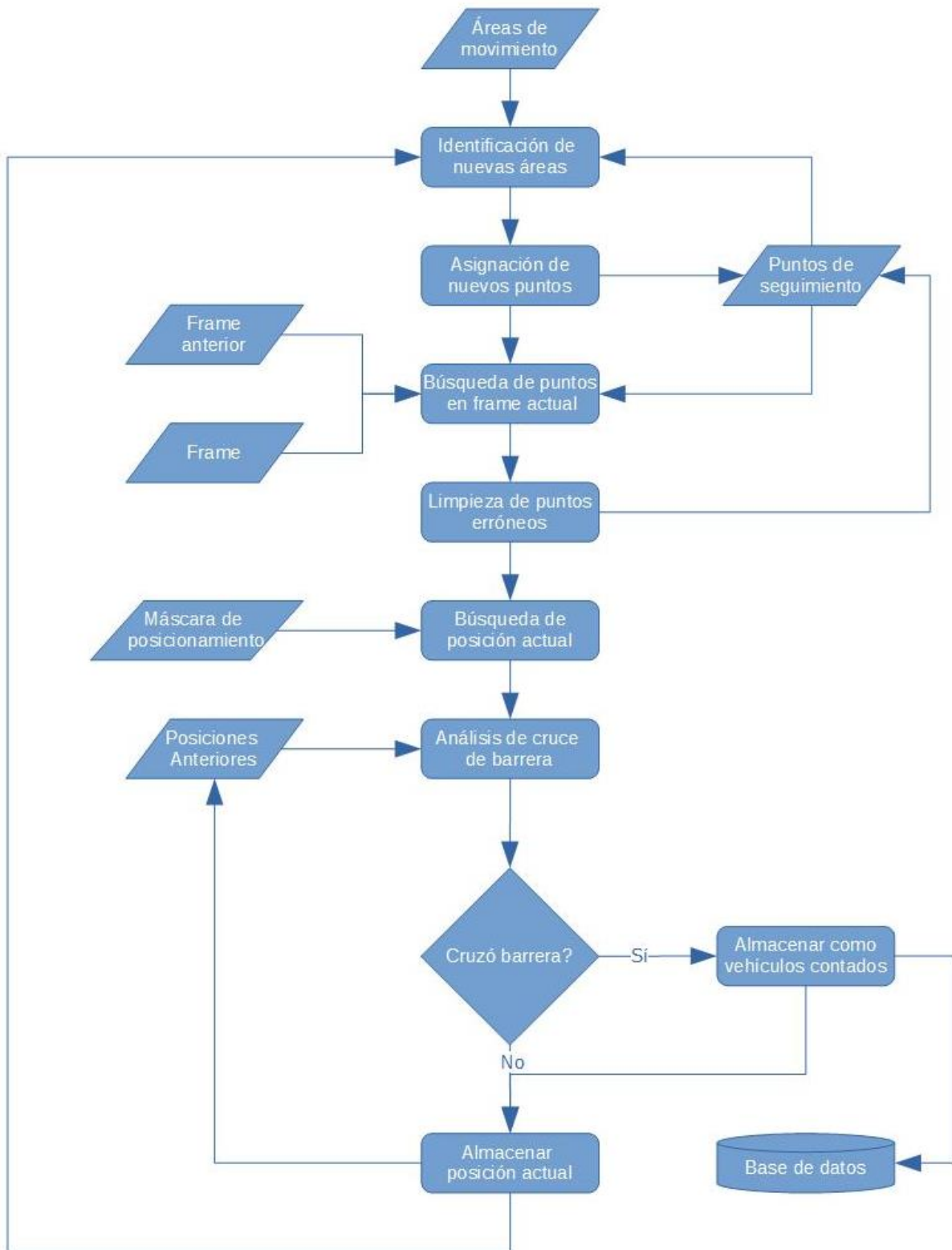


Ilustración 25 – Diagrama de flujo: Seguimiento y conteo de vehículos

Una vez obtenida la posición actual dentro de la imagen, el módulo analiza a cuál de las áreas definidas en la máscara de posicionamiento corresponde y almacena dicho estado, para realizar futuras comparaciones. Si el punto se encuentra delante de la barrera y en tiempos anteriores estuvo en la zona anterior a la misma, o viceversa, se considera que el vehículo cruzó por la barrera y es contado.

Cuando un vehículo es contado, el módulo almacena un elemento en la base de datos, con la fecha y hora de la ocurrencia de dicho evento.

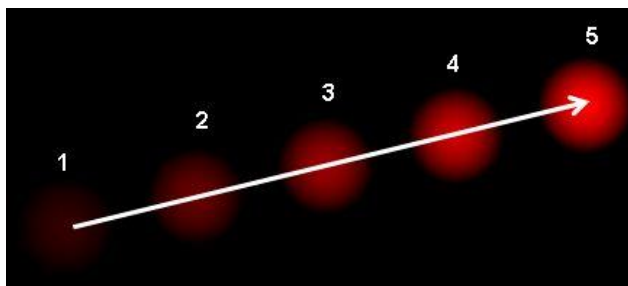


Ilustración 26 – Flujo óptico<sup>5</sup>

#### 2.2.3.4- Generación de alarmas

El módulo genera alarmas diferenciadas según su tipo y prioridad. El tipo está dado por la variable que las disparan; éstas son:

- Alarmas por flujo.
- Alarmas por detención.

Mientras que la prioridad, es una medida del nivel de atención que debe considerarse sobre la alarma en cuestión, a saber:

- Alarmas de precaución.
- Alarmas de peligro.

Los parámetros de las variables para que se generen estos diferentes tipos de alarmas son ingresados por el usuario y obtenidos por el módulo en la etapa de inicio como fue descrito anteriormente.

A continuación, se describirá el proceso de generación de cada tipo de alarma.

##### Alarmas por flujo:

Como su nombre lo dice, estas alarmas son generadas por el nivel de flujo vehicular. Cuando existe una congestión, el nivel de flujo vehicular se elevará considerablemente, con lo cual, mediante la configuración de valores máximos de flujo para la condición de “precaución” y “peligro”, el usuario podrá ser alertado de congestiones en el tránsito de la vía monitoreada.

Para el establecimiento de este tipo de alarmas, el módulo calcula el flujo actual y lo contrasta con los valores máximos ingresados por el usuario. Para obtener una medición representativa, se tienen en cuenta los siguientes parámetros:

---

<sup>5</sup> Fuente: [http://docs.opencv.org/3.1.0/d7/d8b/tutorial\\_py\\_lucas\\_kanade.html](http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_lucas_kanade.html)



- Frecuencia de medición: Es el tiempo transcurrido entre una medición y otra. Para su establecimiento se tiene en cuenta el “ciclo de semaforización”, el cual es definido como la duración de la fase verde más la amarilla y la roja. Según normas de orden práctico, el ciclo debe estar comprendido entre 30 y 120 segundos, por tal motivo se estableció la frecuencia de medición en 2 minutos y así asegurar que entre mediciones siempre exista una fase verde, y con esta una tanda de conteo.
- Intervalo de medición: El flujo vehicular es la cantidad de vehículos contados por unidad de tiempo; es decir:

$$\text{Flujo} = \frac{\text{Cantidad de vehículos}}{\text{Intervalo de tiempo}}$$

Para su cálculo, el módulo lee de la base de datos la cantidad de vehículos detectados en un determinado intervalo de tiempo.

Para que la medición sea más continua y sin variaciones abruptas, el intervalo de tiempo debe comprender varias tandas de conteo, como consiguiente, ser mayor a la frecuencia de medición. Tras reiteradas pruebas, se infirió que el valor adecuado debe ser cinco veces mayor al de la frecuencia de medición, lo que determina un tiempo de 10 minutos.

En resumen, el módulo lee cada 2 minutos de la base de datos la cantidad de vehículos contados en los 10 minutos anteriores al momento del cálculo, luego obtiene un valor de flujo que compara con los valores máximos considerados para “precaución” y “peligro”, arrojando, de ser necesario, las alarmas correspondientes. En la ilustración del diagrama de flujos correspondiente se explica el procedimiento de la generación de alarmas.

#### Alarmas por detención:

Las alarmas por detención serán activadas cuando un vehículo se detenga por un determinado tiempo en la vía de tránsito. Este tiempo es establecido por el usuario en la configuración.

Para su análisis, el módulo toma los puntos de seguimiento y obtiene el tiempo desde la última vez que cada uno se encontró dentro de un área de movimiento. Es decir, el tiempo transcurrido desde el último desplazamiento del vehículo. Seguidamente, contrasta este tiempo con los correspondientes a las prioridades de “precaución” y “peligro”. Si éste supera dichos tiempos, el módulo genera la alarma respectiva y almacena en la base de datos la fecha, hora, prioridad y zona donde se encuentra detenido el vehículo. En caso de tener prioridad “peligro”, se almacena una captura de la imagen, señalizando el vehículo en cuestión, dando por terminado el análisis para el mismo.

Es importante destacar que, en caso de que se haya configurado una zona de estacionamiento, a partir de la máscara de estacionamiento generada en la primera etapa el módulo descarta del análisis los puntos que se encuentren dentro de dicha zona. Esto permite que no se generen falsas alarmas por vehículos estacionados correctamente.

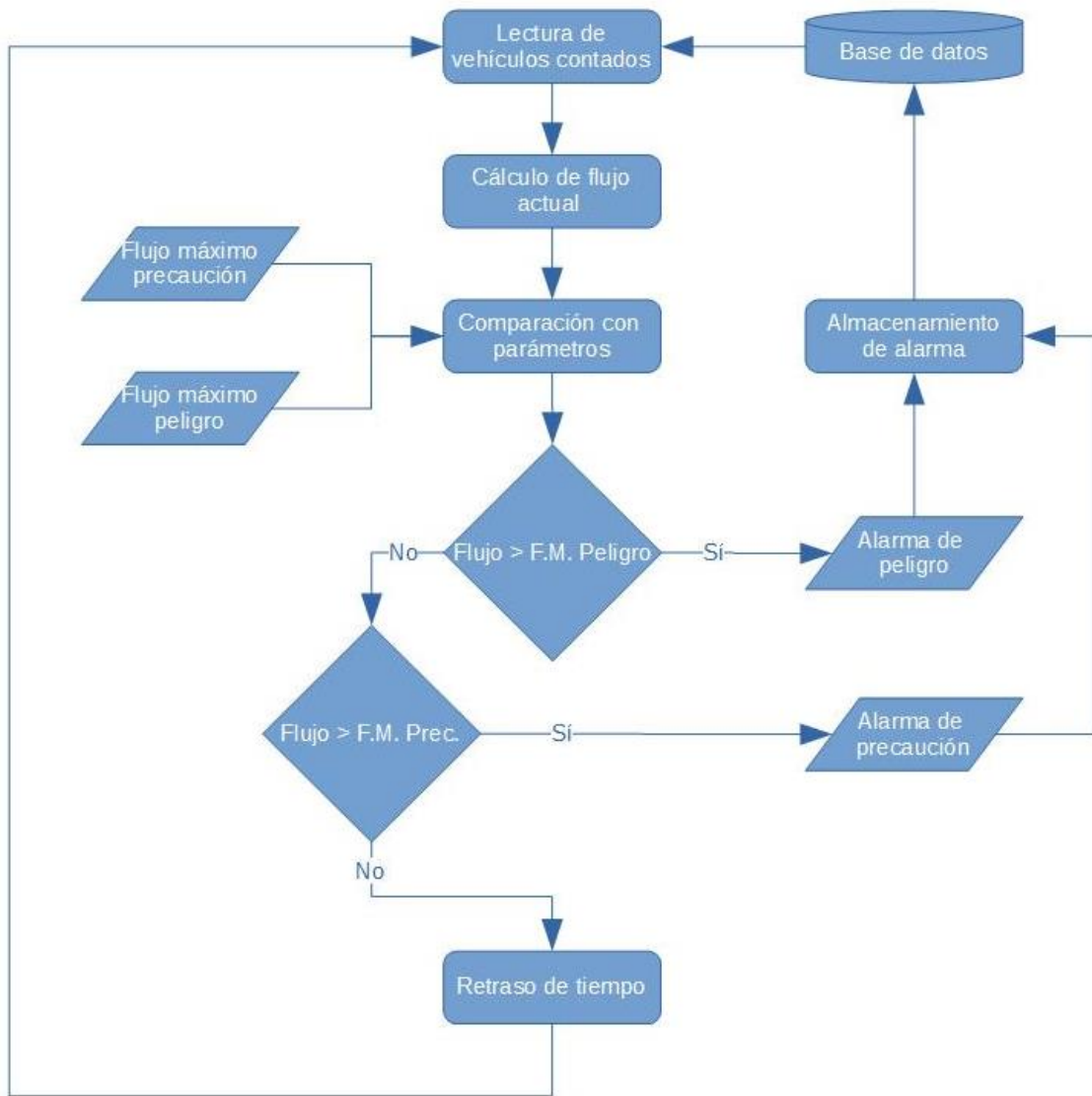


Ilustración 27 – Diagrama de flujo: Generación de alarmas por flujo

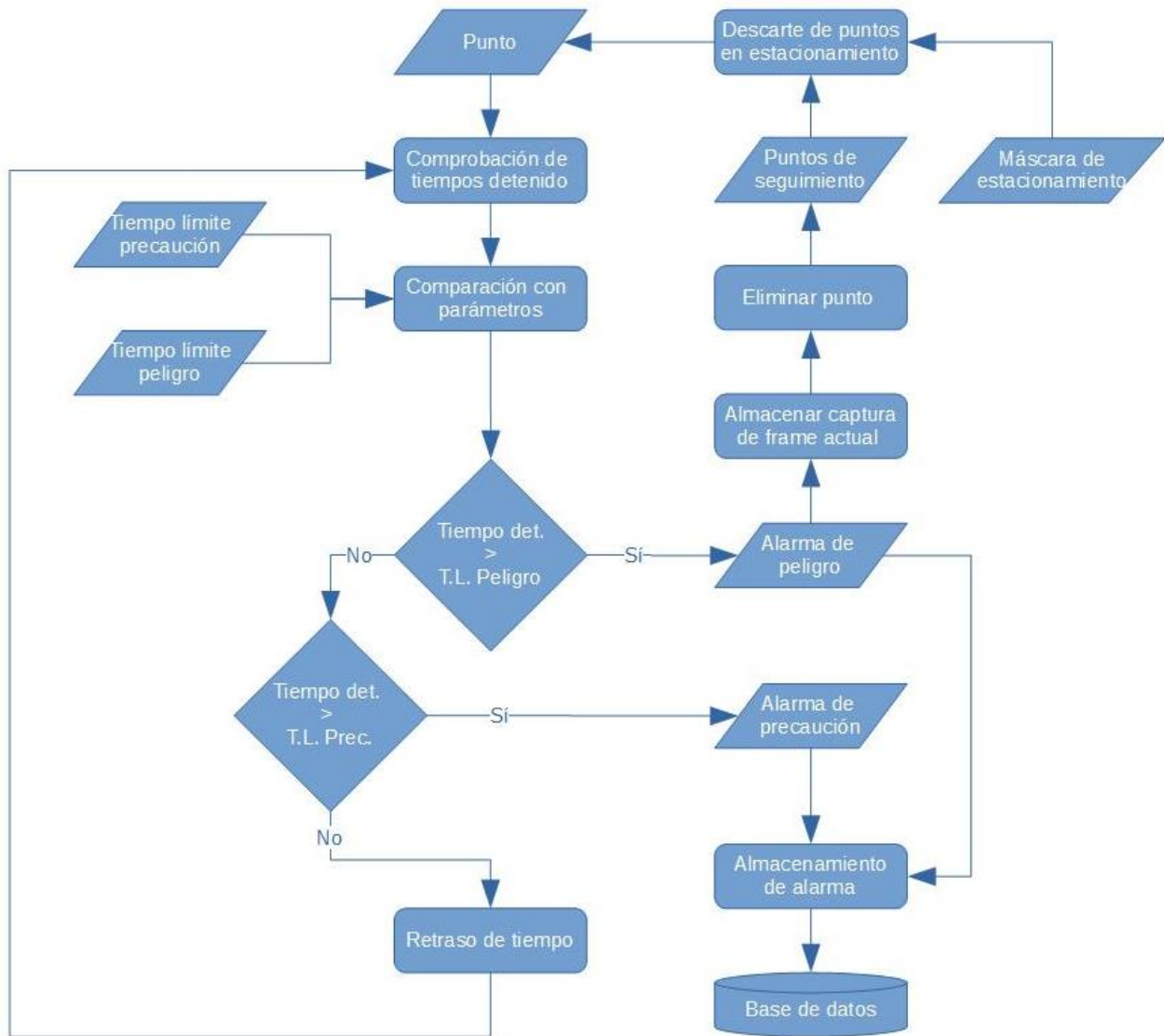


Ilustración 28 – Diagrama de flujo: Generación de alarmas por detención

#### 2.2.4- Análisis del funcionamiento

El funcionamiento comienza con el proceso “raspimjpeg” que crea una conexión con la cámara y genera un flujo continuo de capturas en el directorio “/dev/shm/mjpeg”, todas con el mismo nombre “cam.jpg”. Esta carpeta es de memoria RAM por lo que no afecta la memoria de la tarjeta SD.

El procesamiento de la imagen se realiza mediante un proceso denominado “moduloPDI” desarrollado en el lenguaje python. Éste es el encargado de tomar cada *frame* generado por el módulo raspimjpeg y procesarlo con el fin de realizar el conteo vehicular y establecer los parámetros deseados. En la sección anterior (2.2.4) se explica el proceso en detalle, el funcionamiento puede ser dividido en los siguientes cuatro bloques:

1. Inicialización y obtención de parámetros.
2. Detección de movimiento.
3. Seguimiento y conteo de vehículos.
4. Generación de alarmas.

Los datos obtenidos en este procesamiento son almacenados en una base de datos MySQL. La base de datos se denomina “Transito”, ésta contiene tres tablas que van almacenando distinta información: “VEHICULOS”, “ALARMAS” y “VARIABLES” (es importante aclarar que tanto los nombres como los campos no deben contener tildes).

La primera tabla, VEHICULOS, tiene la siguiente estructura:

Fecha	Hora	Direccion

Tabla 1 – Base de Datos Remota - Vehículos

En esta tabla se almacenan la información de fecha, hora y dirección en la que pasa cada vehículo de una zona al otro extremo. En este caso, el campo dirección sirve para identificar que hubo paso en contramano.

La tabla ALARMAS, contiene la siguiente información:

Fecha	Hora	Alarma	Prioridad	Estado

Tabla 2 – Base de Datos Remota - Alarmas

Se determinaron dos tipos de alarmas, como se explicó en la sección anterior, “vehículo detenido en zona X” y “flujo vehicular” (excesivo o crítico). Las prioridades son: precaución y peligro. El tiempo transcurrido en la detención de los vehículos que generan las alarmas es el que determina el tipo de prioridad que tenga, este tiempo es configurado por el usuario administrador desde la aplicación web. También se ingresa desde la aplicación el valor máximo de vehículos por minuto para considerar un flujo como precaución o peligro.

La tabla VARIABLES contiene los siguientes campos:

Flujo	Cantidad	Servidor	MyID

Tabla 3 – Base de Datos Remota - Variables

El campo “Flujo” contiene el flujo actual calculado como cantidad de vehículos que han circulado en los últimos 10 minutos. “Cantidad” es un contador de 0 a 999 que se utiliza para mostrar en la página web la cantidad de vehículos que se van contando. “Servidor” contiene el *hostname* del servidor de administración (no el propio). “MyID” lleva un número de identificación que asigna el servidor de administración al dispositivo de medición.

Esta base de datos es accesible por el usuario desde la web. El mismo puede realizar consultas para conocer históricos. El análisis de estas variables y elementos del flujo

vehicular permite conocer el comportamiento del tránsito y las características del mismo, y obtener requisitos básicos para el planeamiento, proyecto y operación de calles y obras complementarias del sistema de transporte.

Como se comentó anteriormente, se ha instalado un servidor en el dispositivo. Es por esto que se desarrolló una aplicación web en el mismo que permite el acceso a toda la información contenida, tanto la base de datos como el flujo de imágenes obtenidas del procesamiento que permiten visualizar un video en *streaming*.

A continuación, se muestra una imagen de la aplicación web.

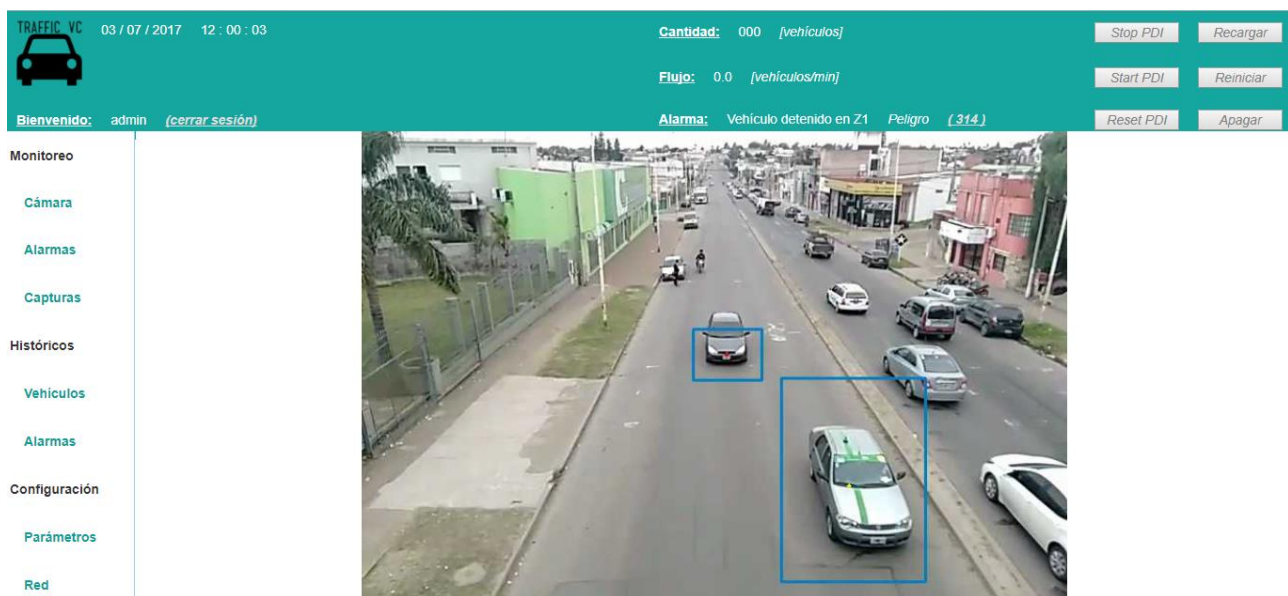


Ilustración 29 – Aplicación web del dispositivo: página principal

Como se aprecia en la imagen, la aplicación web permite en su vista principal la visualización del video del tránsito en tiempo real. En la parte superior, se muestran los parámetros asociados: cantidad de vehículos contados (de 0 a 999), el flujo instantáneo (vehículos/minutos) y las distintas alarmas que se van generando. También existe la posibilidad de iniciar, apagar o reiniciar el procesamiento mediante los botones situados a la derecha.

En la parte izquierda del cuerpo de esta página, se encuentra el menú para navegar y acceder a otras funciones.

La sección “Monitoreo” contiene tres partes. La primera corresponde a la página donde se observa el video. Para la visualización, la imagen de cada *frame* analizado se guarda en un directorio. Ésta se incluye al documento *html* como un elemento imagen (`<img>`). La imagen fuente del elemento es actualizada cada un determinado tiempo, la nueva imagen es almacenada durante el procesamiento y contiene el mismo nombre que la anterior. De esta manera se logra la transmisión (o *streaming*) en tiempo real del video.

Luego, hay una página donde se encuentran las alarmas que no han sido atendidas, aquí se da la opción de seleccionar las que se deseen “atender”, esto significa eliminarlas de esta la lista. Por último, la página donde se pueden ver, descargar y eliminar las

capturas realizadas por la cámara al momento en que se producen alarmas críticas. La opción “eliminar” está solo permitida para usuarios con permiso de administrador. Cada captura tiene un nombre con fecha y hora en la que se produjo tal evento.

**Alarmas sin atender**

Fecha	Hora	Alarma	Prioridad	Captura	
<input type="checkbox"/>	2017-07-03	12:00:24	Vehículo detenido en Z3	Peligro	-
<input type="checkbox"/>	2017-07-03	12:00:14	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	12:00:12	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	11:58:05	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	11:58:04	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	11:57:21	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	11:57:19	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	11:54:55	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:54:55.jpg
<input type="checkbox"/>	2017-07-03	11:53:51	Vehículo detenido en Z3	Peligro	Detenido2017-07-03@11:53:51.jpg
<input type="checkbox"/>	2017-07-03	11:53:42	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:53:42.jpg
<input type="checkbox"/>	2017-07-03	11:51:28	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:51:28.jpg
<input type="checkbox"/>	2017-07-03	11:51:21	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:51:21.jpg
<input type="checkbox"/>	2017-07-03	11:50:49	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:50:49.jpg
<input type="checkbox"/>	2017-07-03	11:50:48	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:50:48.jpg
<input type="checkbox"/>	2017-07-03	11:49:36	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:49:36.jpg
<input type="checkbox"/>	2017-07-03	11:49:30	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@11:49:30.jpg
<input type="checkbox"/>	2017-07-03	09:39:32	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@09:39:32.jpg
<input type="checkbox"/>	2017-07-03	09:39:22	Vehículo detenido en Z1	Peligro	Detenido2017-07-03@09:39:22.jpg
<input type="checkbox"/>	2017-07-03	09:39:21	Vehículo detenido en Z3	Peligro	Detenido2017-07-03@09:39:21.jpg

Ilustración 30 – Aplicación web del dispositivo: sección de Monitoreo – Alarmas sin atender

**Capturas**

Nombre	Fecha	Descargar	Eliminar
Detenido2017-07-03@11:54:55.jpg	03/07/2017 - 11:54:55	Descargar	Eliminar
Detenido2017-07-03@11:53:51.jpg	03/07/2017 - 11:53:51	Descargar	Eliminar
Detenido2017-07-03@11:53:42.jpg	03/07/2017 - 11:53:42	Descargar	Eliminar
Detenido2017-07-03@11:51:28.jpg	03/07/2017 - 11:51:28	Descargar	Eliminar
Detenido2017-07-03@11:51:21.jpg	03/07/2017 - 11:51:21	Descargar	Eliminar
Detenido2017-07-03@11:50:49.jpg	03/07/2017 - 11:50:49	Descargar	Eliminar
Detenido2017-07-03@11:50:48.jpg	03/07/2017 - 11:50:48	Descargar	Eliminar
Detenido2017-07-03@11:49:36.jpg	03/07/2017 - 11:49:36	Descargar	Eliminar
Detenido2017-07-03@11:49:30.jpg	03/07/2017 - 11:49:30	Descargar	Eliminar
Detenido2017-07-03@09:39:32.jpg	03/07/2017 - 09:39:32	Descargar	Eliminar
Detenido2017-07-03@09:39:22.jpg	03/07/2017 - 09:39:22	Descargar	Eliminar
Detenido2017-07-03@09:39:21.jpg	03/07/2017 - 09:39:22	Descargar	Eliminar
Detenido2017-07-03@09:39:11.jpg	03/07/2017 - 09:39:11	Descargar	Eliminar
Detenido2017-07-03@09:39:10.jpg	03/07/2017 - 09:39:10	Descargar	Eliminar
Detenido2017-07-03@09:38:55.jpg	03/07/2017 - 09:38:55	Descargar	Eliminar
Detenido2017-07-03@09:38:36.jpg	03/07/2017 - 09:38:37	Descargar	Eliminar
Detenido2017-07-03@09:38:30.jpg	03/07/2017 - 09:38:31	Descargar	Eliminar
Detenido2017-07-03@09:38:20.jpg	03/07/2017 - 09:38:20	Descargar	Eliminar
Detenido2017-07-03@09:38:17.jpg	03/07/2017 - 09:38:17	Descargar	Eliminar

Ilustración 31 – Aplicación web del dispositivo: sección de Monitoreo – Capturas

La sección “Históricos” se divide en dos partes, “Vehículos” y “Alarmas”. El objetivo de ambas es el mismo: seleccionar un intervalo de fecha y hora en el cual se desea obtener un histórico de datos almacenados en el servidor. Una vez seleccionados “desde” y “hasta”, se visualiza una tabla con estos datos. Es posible descargar esta tabla en formato “.csv” al presionar el ícono de hoja de cálculo. Además, se da la opción de ver gráficas que permiten dar otro sentido a esta información.



Ilustración 32 – Aplicación web del dispositivo: sección de Históricos – Vehículos

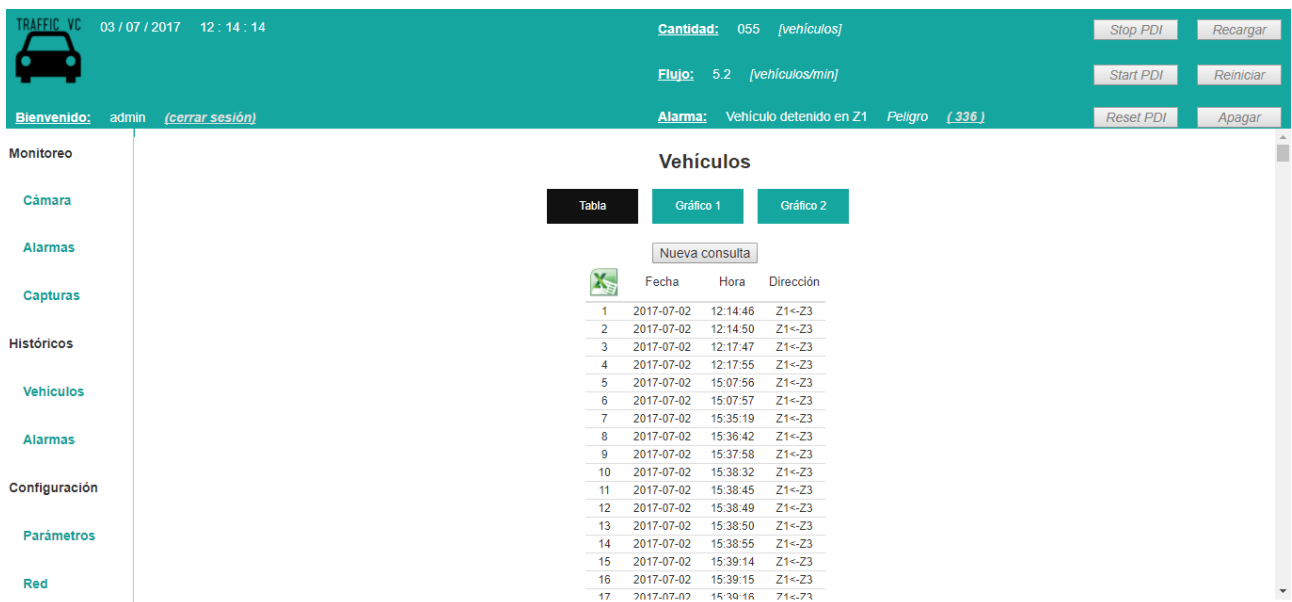


Ilustración 33 – Aplicación web del dispositivo: consulta tabla vehículos

Para el caso de los vehículos, en la página se muestran dos gráficas:

- Flujo instantáneo. La base de datos arroja fecha y hora en la cual se contabilizó cada vehículo. Para obtener la curva del flujo, se dividió el intervalo de consulta en 200 muestras. Cada una de estas muestras considera un valor acumulado de la cantidad de vehículos que existen entre la actual, la anterior y un porcentaje antes que ésta. De esta forma se logra continuidad en la curva, ya que el flujo de vehículos no es terminante en el tiempo, sino que hay un cierto intervalo en el cual estos vehículos circulan por la vía hasta desagotarse por completo.
- Volumen de flujo por minuto, hora o día, dependiendo del intervalo principal seleccionado. Éste es un gráfico de barras en donde se muestra la cantidad de



vehículos que han circulado en cada intervalo fijo, puede ser por minuto, por hora o día, dependiendo de si la consulta que se ha realizado abarca un intervalo de horas, días o meses (respectivamente).

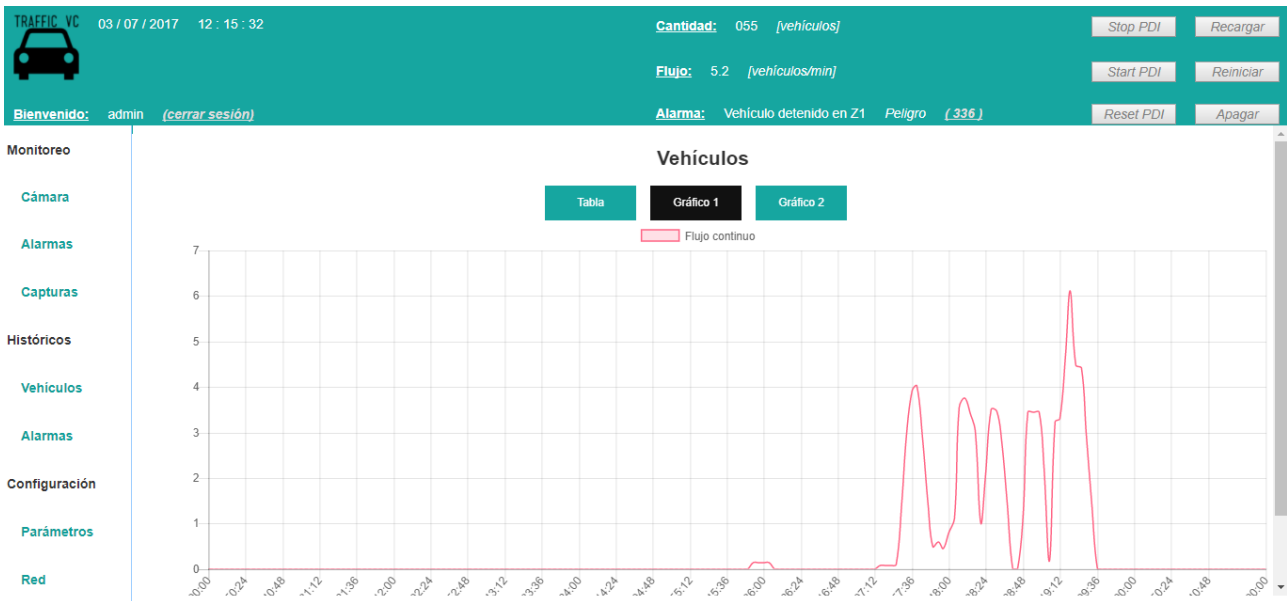


Ilustración 34 – Aplicación web del dispositivo: consulta gráfico 1 vehículos

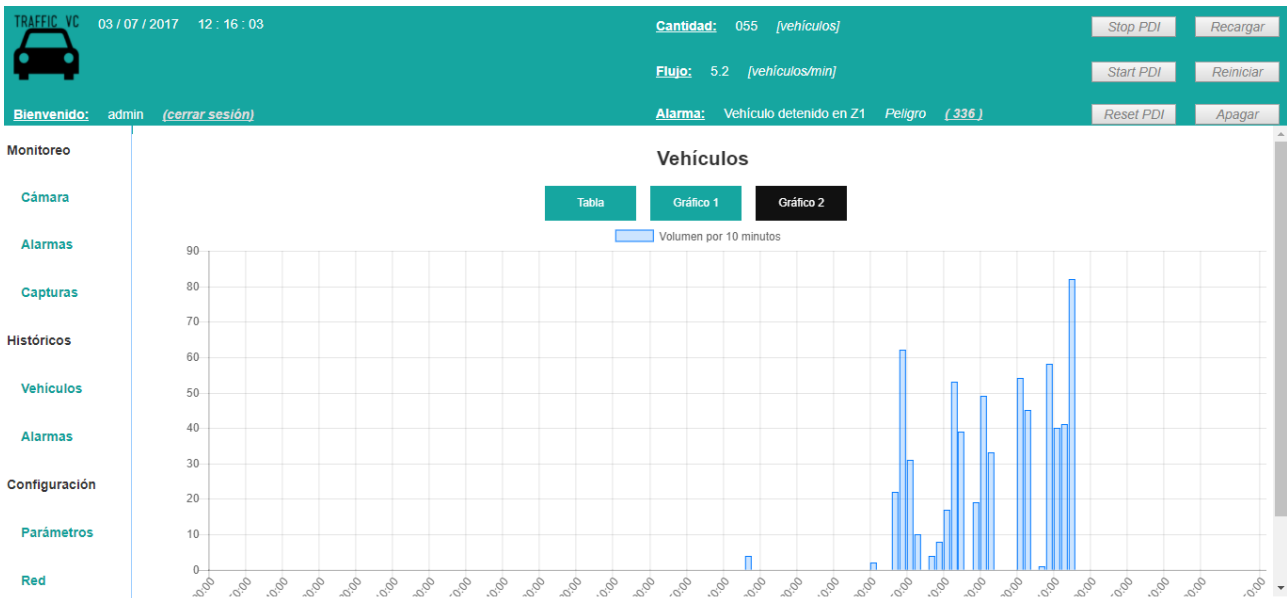


Ilustración 35 – Aplicación web del dispositivo: consulta gráfico 2 vehículos

En el caso de las alarmas, se muestran las siguientes gráficas:

- Porcentaje de detenciones por cada zona. Se muestra un gráfico de torta con las tres porciones representando los porcentajes de alarmas de detención ocurridas en cada una de las zonas delimitadas en la vía de tránsito.
- Cantidad de cada alarma por minuto, hora o día, dependiendo del intervalo principal seleccionado. El cálculo es el mismo que en el caso de vehículos, se muestran barras por cada alarma, en el mismo gráfico.



Ilustración 36 – Aplicación web del dispositivo: sección de Históricos – Alarmas

	Fecha	Hora	Alarma	Prioridad	Estado
1	2017-07-02	12:14:51	Vehículo detenido en Z3	Precaución	Atendida
2	2017-07-02	12:14:59	Vehículo detenido en Z3	Peligro	Atendida
3	2017-07-02	12:15:05	Vehículo detenido en Z3	Precaución	Atendida
4	2017-07-02	12:15:13	Vehículo detenido en Z3	Peligro	Atendida
5	2017-07-02	12:15:20	Vehículo detenido en Z3	Precaución	Atendida
6	2017-07-02	12:15:29	Vehículo detenido en Z3	Precaución	Atendida
7	2017-07-02	12:15:29	Vehículo detenido en Z3	Peligro	Atendida
8	2017-07-02	12:15:32	Vehículo detenido en Z3	Precaución	Atendida
9	2017-07-02	12:15:38	Vehículo detenido en Z3	Peligro	Atendida
10	2017-07-02	12:15:41	Vehículo detenido en Z3	Peligro	Atendida
11	2017-07-02	12:15:43	Vehículo detenido en Z3	Precaución	Atendida
12	2017-07-02	12:15:52	Vehículo detenido en Z3	Peligro	Atendida
13	2017-07-02	12:16:09	Vehículo detenido en Z3	Precaución	Atendida
14	2017-07-02	12:16:17	Vehículo detenido en Z3	Peligro	Atendida
15	2017-07-02	15:07:59	Vehículo detenido en Z3	Precaución	Atendida
16	2017-07-02	15:08:07	Vehículo detenido en Z3	Peligro	Atendida
17	2017-07-02	15:08:14	Vehículo detenido en Z3	Precaución	Atendida

Ilustración 37 – Aplicación web del dispositivo: consulta tabla alarmas

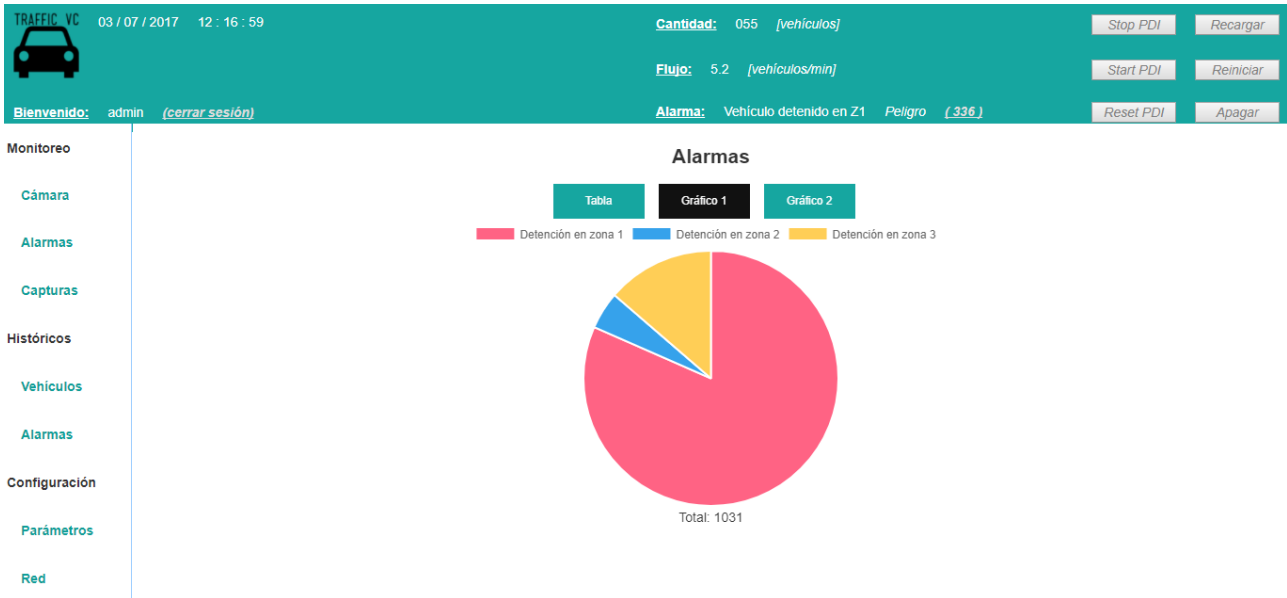


Ilustración 38 – Aplicación web del dispositivo: consulta gráfico 1 alarmas

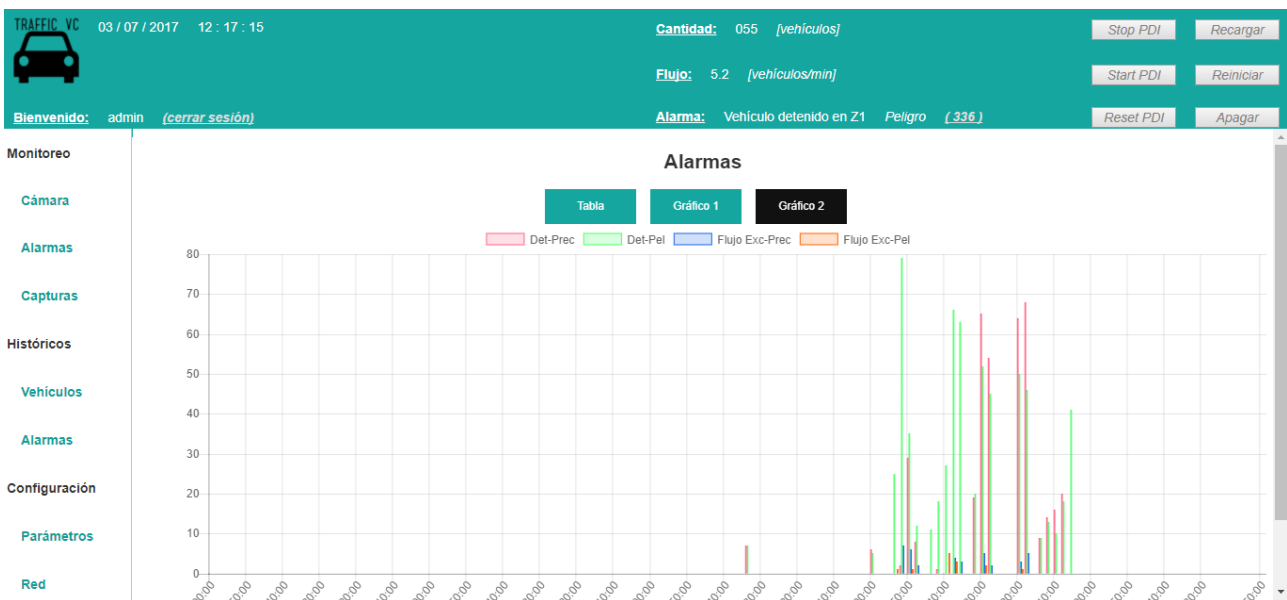


Ilustración 39 – Aplicación web del dispositivo: consulta gráfico 2 alarmas

La sección “Configuración” permite al usuario administrador posicionar la ubicación de la barrera que sirven para delimitar las zonas y además configurar los parámetros considerados para determinar la prioridad de las alarmas.

Para posicionar la barrera, el usuario deberá realizar cuatro *click* con el mouse sobre la imagen de la calle. A medida que se marcan los puntos, se forman las líneas que determinan un cuadrilátero. En el caso de que la calle tenga lugar para estacionamiento, se puede graficar, de la misma manera, el área para definir la zona.

Al presionar el botón “Guardar”, los parámetros son enviados mediante una petición *Ajax* por *url* a un archivo ubicado en el servidor desarrollado en PHP, quien se encarga de guardarlos en el archivo “ConfigPDI.txt”, el mismo que es leído por el módulo “moduloPDI.py” al momento de iniciar el procesamiento.

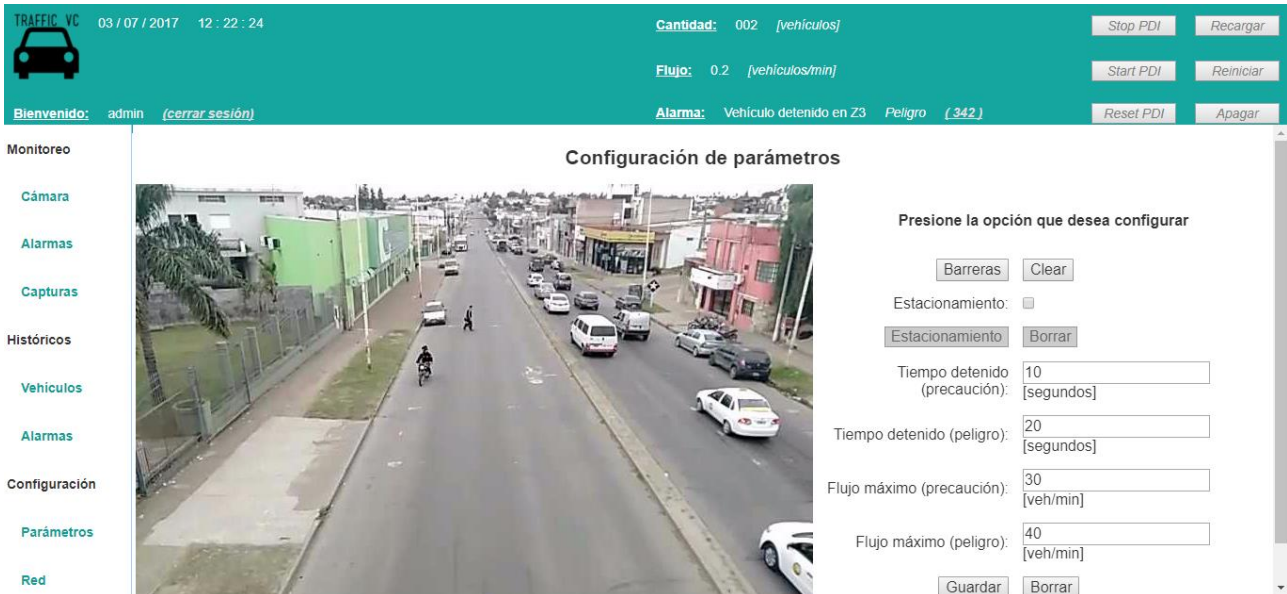


Ilustración 40 – Aplicación web del dispositivo: sección de Configuración de parámetros

Por último, se permite en esta sección configurar aspectos referentes a la red. Es decir, definir el *hostname*, establecer la dirección IP como estática o dinámica, habilitar/deshabilitar conexión *wireless* (inalámbrica), definir la puerta de enlace, entre otros.

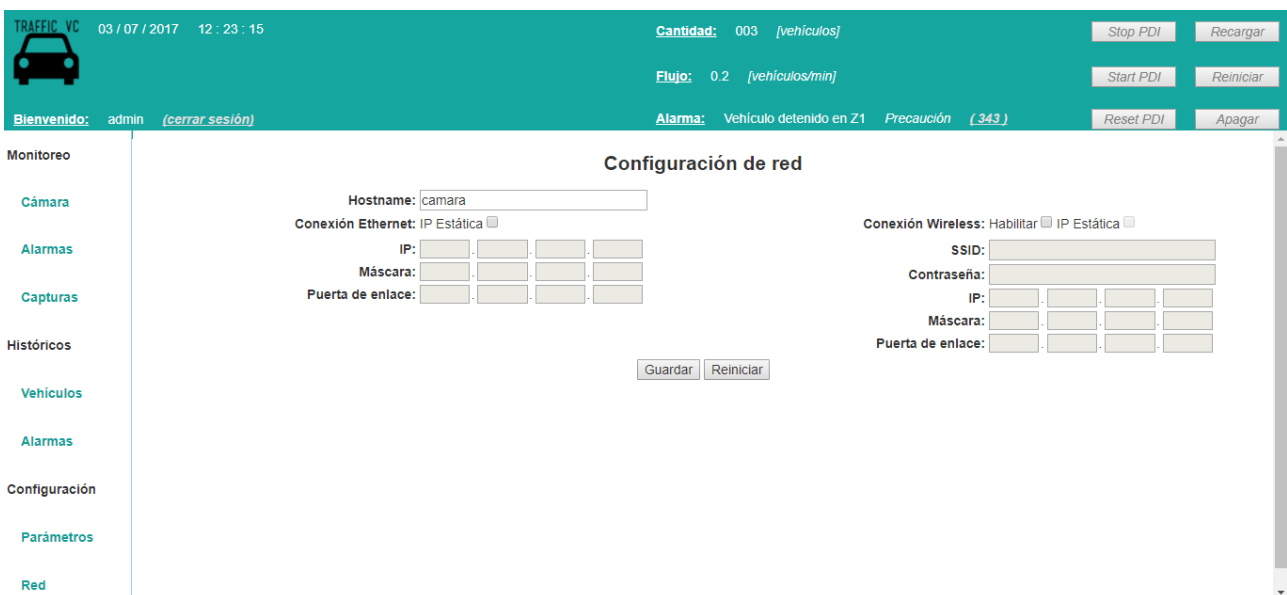


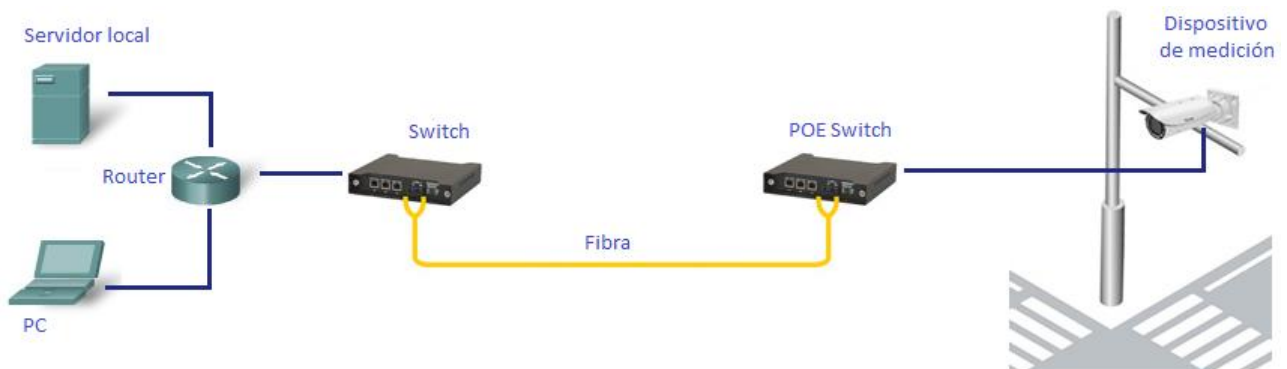
Ilustración 41 – Aplicación web del dispositivo: sección de Configuración de red

### 2.3- Red

Como se mencionó anteriormente, el proyecto se diseñó suponiendo una red LAN ya existente.

Por ejemplo, en la ciudad de Paraná existe una red que conecta los semáforos con la central de monitoreo de tránsito. La idea es conectar el dispositivo de medición junto al semáforo y la unidad de administración central de datos (servidor local) en la central.

La conexión del sistema a la red tendría un esquema similar al siguiente:



*Ilustración 42 – Esquema aproximado de conexión del sistema a la red*

El esquema anterior puede variar dependiendo de la instalación de red que tenga cada ciudad en la que implemente el sistema.

## 2.4- Unidad de administración central de datos

### 2.4.1- Estructura

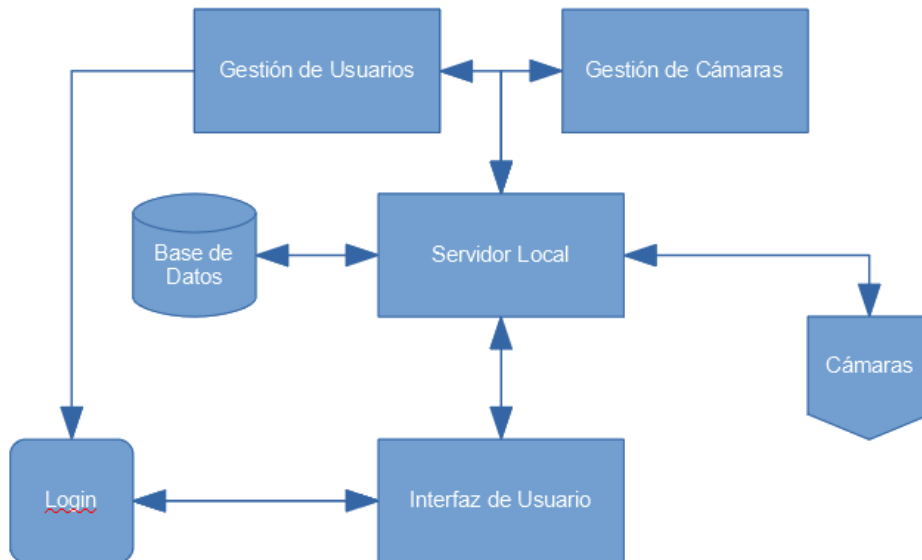


Ilustración 43 – Estructura de la unidad de administración central de datos

### 2.4.2- Investigación de componentes

Las características necesarias para el servidor central, no son tan elevadas como las requeridas para el dispositivo final de medición, sólo se requiere de un dispositivo capaz de soportar un servidor web y configuraciones básicas de red.

Con estas condiciones podría haberse optado por desarrollar un sistema embebido con conectividad LAN. Sin embargo, por cuestiones de compatibilidad en la programación y diseño, se optó por usar el mismo sistema embebido *Raspberry Pi 3* que se analizó anteriormente para el dispositivo de medición. Si bien, el costo del insumo sería superior debido a la cantidad de características que posee pero que no se utilizan, y que un dispositivo dedicado no tendría, la ventaja de utilizar uno ya desarrollado con soporte del sistema operativo *Raspbian* permitió ahorrar mucho tiempo de desarrollo y programación.

Sobre la *Raspberry Pi* se instaló el sistema operativo *Raspbian*, que es una versión de *Linux* modificada específicamente para funcionar en la placa *Raspberry*. El servidor web se implementa mediante un servidor *Apache II* y la programación mediante el módulo *PHP 5*. La base de datos es *MySQL* y se administra con la herramienta *phpmyadmin*. Todos estos módulos y herramientas se descargan directamente desde repositorios web por medio de comandos incluidos en el sistema operativo.

La alimentación requiere de una fuente de 5 Volts 2 Ampere, aplicada a través del puerto micro *USB*. La conexión *LAN* puede realizarse mediante conexión *Ethernet* o *WiFi*.

### 2.4.3- Propuestas de circuitos o esquemas.

La placa *Raspberry PI* posee en su interior todos los circuitos y componentes necesarios para el funcionamiento. Para el conexionado se debe garantizar la alimentación eléctrica y la conexión *LAN* (si la conexión deseada es *Ethernet*).

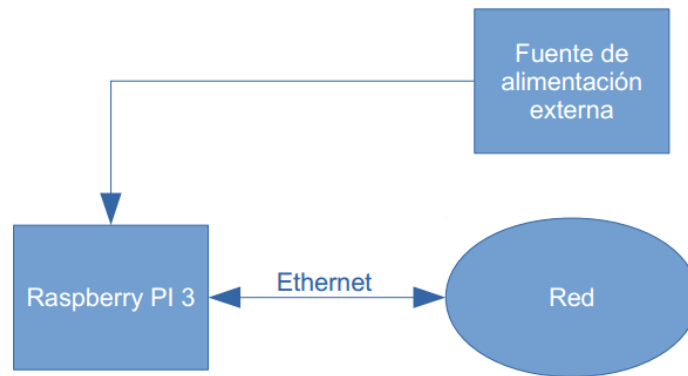


Ilustración 44 – Esquema de conexión del servidor local

### 2.4.4- Desarrollo del software o firmware

Las funciones específicas que realiza el servidor fueron programadas en lenguaje *PHP* versión 5. Las opciones de visualización se observan en formato *html* con el implemento de código *javascript*. La mayoría de las funciones son accesibles al usuario a través de la página web, aunque hay otras que son transparentes al mismo y son mayormente las utilizadas en la comunicación con los dispositivos remotos.

Las unidades básicas incluidas en el servidor son las enumeradas a continuación:

#### 2.4.4.1- Monitoreo Proactivo

Es importante mantener control sobre los dispositivos conectados al sistema. Para esto, cada 1 segundo se intenta establecer una conexión a cada una de las cámaras que se encuentren activas mediante una petición “curl” a dicho *host*, si no se recibe una respuesta en determinado tiempo, se considera que el remoto está desconectado.

Luego, para los dispositivos que se consideran conectados, se realiza una petición similar, pero a una *url* que devuelve, en formato *json*, el número de alarmas actualmente activas.

Estos dos parámetros son actualizados en el sector superior de la interfaz de usuario. Para analizar esta información en profundidad, se coloca en cada valor un enlace a las páginas de “[Monitoreo: Estado](#)” y “[Monitoreo: Alarmas](#)” respectivamente.



Ilustración 45 – Servidor Local: Monitoreo Proactivo

#### 2.4.4.2- Monitoreo: Cámaras

Muestra en tiempo real la imagen de 2, 4 o 6 cámaras al mismo tiempo. La cantidad de cámaras y la elección de las mismas se realiza desde la unidad de “Configuración: Monitoreo”.

Se toma el archivo generado por dicha unidad donde queda guardada la configuración deseada y en base a esto se lee la base de datos para extraer la información necesaria y dar formato a la página de visualización de cámaras. Este formato incluye un llamado a la dirección *html* de cada cámara donde se muestra la imagen en tiempo real. Esto se realiza mediante una etiqueta *iframe* del lenguaje *html*.

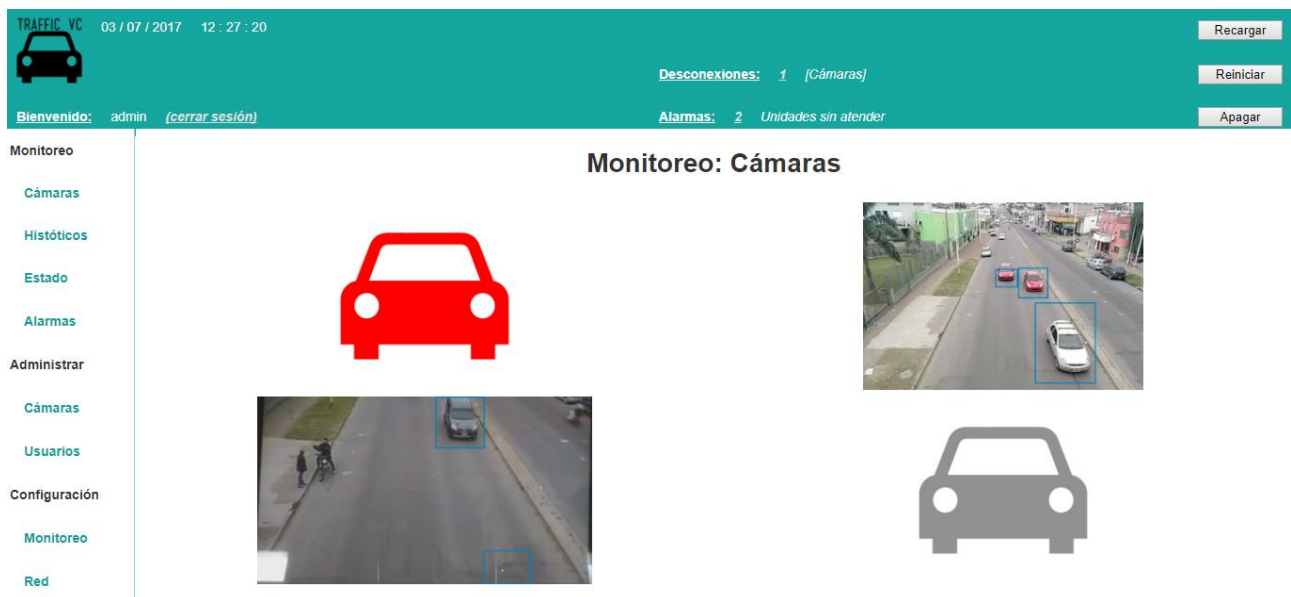


Ilustración 46 – Aplicación web del servidor local: sección de Monitoreo - Cámaras

#### 2.4.4.3- Monitoreo: Históricos

En este caso, el proceso consta de dos etapas. En primer lugar, se listan las cámaras guardadas en la base de datos y se da la posibilidad de elegir entre el histórico de tráfico y de alarmas que generan los dispositivos remotos.

Una vez seleccionado el tipo de datos en la cámara deseada, se realiza una llamada al mismo código *html* que se ejecuta en la cámara deseada para el tipo de alarma escogido.

Se agrega a la tabla una identificación visual respecto de si el remoto está en funcionamiento o no, ya que, de no estarlo, la petición será incorrecta.

Cámara	Host	Gráfico	Estado	IR
Fomento2	192.168.1.2	Vehículos ▼		IR
Terminal	terminal	Vehículos ▼		IR
Fomento1	fomento1	Vehículos ▼		IR
Formosa575	formosa575	Vehículos ▼		IR

Ilustración 47 – Aplicación web del servidor local: sección de Monitoreo – Históricos

**Vehículos**

Desde: 03 Julio 2017 00:00 hs  
 Hasta: 03 Julio 2017 23:59 hs

Mostrar

Ilustración 48 – Aplicación web del servidor local: sección de Monitoreo – Estadísticas, acceso remoto

Al general la petición, el resultado es el mismo que se obtiene del remoto:



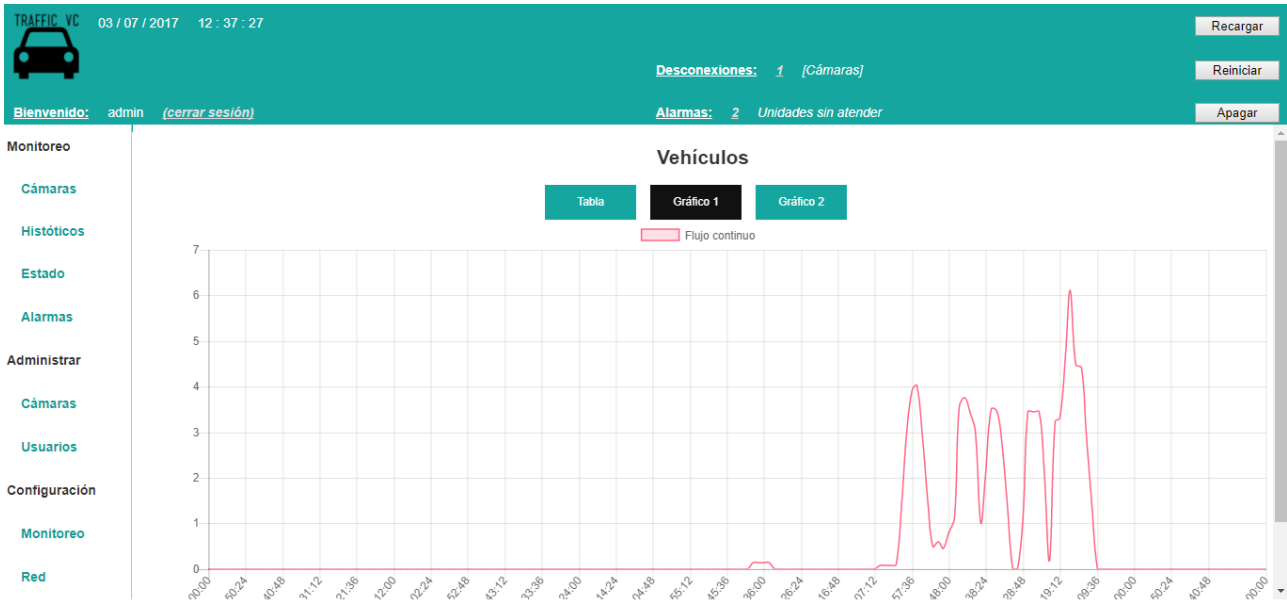


Ilustración 49 – Aplicación web del servidor local: sección de Monitoreo – Históricos, acceso remoto, gráfico.

#### 2.4.4.4- Monitoreo: Estado

Es importante conocer si se pierde contacto con alguna de los dispositivos. Para esto existe un código *php* ejecutándose periódicamente que hace contacto con todas las cámaras registradas en la base de datos. En caso de que no se pueda contactar con alguna de las cámaras, se muestra una notificación en el área de alarmas. En esta área existe un enlace a la página de “Monitoreo: Estado” que muestra el estado actual de cada cámara.

Cámara	Host	Descripción	Estado	IR
Fomento2	192.168.1.2	9 de Julio 3200 - Sur		IR
Terminal	terminal	Av Ramirez - Sur		IR
Fomento1	fomento1	9 de Julio 3200 - Norte		IR
Formosa575	formosa575	Almafuerte al 900		IR

Ilustración 50 – Aplicación web del servidor local: sección de Monitoreo – Estado

Esta página, toma las cámaras registradas en la base de datos y arma una tabla con la información de las mismas indicando el estado con un icono verde (conectada) o rojo

(desconectada). Cada línea de la tabla cuenta con un enlace que permite acceder a la interfaz de cada dispositivo.

### 2.4.4.5- Monitoreo: Alarmas

El monitoreo de alarmas consiste en realizar una petición de tipo *curl* a una *url* específica dentro de cada cámara la cual devuelve en formato *json* la cantidad de alarmas desatendidas en la misma. Periódicamente se realiza esta petición y la cantidad de remotas que muestran alarmas sin atender es mostrada en la sección de alarmas superior.

Cámara	Host	Alarmas activas	Estado
Fomento2	192.168.1.2		
Terminal	terminal	369	
Fomento1	fomento1		
Formosa575	formosa575	127	

Ilustración 51 – Aplicación web del servidor local: sección de Monitoreo – Alarmas

Fecha	Hora	Alarma	Prioridad	Captura	
<input type="checkbox"/>	2017-07-03	12:28:48	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	12:28:40	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:28:31	Vehículo detenido en Z3	Precaución	-
<input type="checkbox"/>	2017-07-03	12:28:12	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:28:11	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:27:58	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:27:50	Vehículo detenido en Z2	Precaución	-
<input type="checkbox"/>	2017-07-03	12:27:42	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:27:40	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	12:27:32	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:27:30	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:25:57	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	12:25:49	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:25:34	Vehículo detenido en Z2	Peligro	-
<input type="checkbox"/>	2017-07-03	12:25:30	Vehículo detenido en Z1	Peligro	-
<input type="checkbox"/>	2017-07-03	12:25:26	Vehículo detenido en Z2	Precaución	-
<input type="checkbox"/>	2017-07-03	12:25:23	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:25:22	Vehículo detenido en Z1	Precaución	-
<input type="checkbox"/>	2017-07-03	12:25:11	Vehículo detenido en Z2	Precaución	-

Ilustración 52 – Aplicación web del servidor local: sección de Monitoreo – Alarmas, acceso remoto

Es importante destacar que no se muestran la cantidad de alarmas, sino la cantidad de dispositivos remotos que presentan alarmas sin atender. Para analizar en detenimiento esta información, en la sección “Monitoreo: Alarmas” se listan las cámaras registradas con una columna dedicada a la cantidad de alarmas que cada una presente.

Al final de cada línea se agrega un enlace a la página en la que se administran las alarmas en la remota correspondiente a esa entrada.

#### 2.4.4.6- Administrar: Cámaras

Para administrar las cámaras se genera, en primer lugar, una lista de las remotas guardadas en la base de datos, junto con el estado de las mismas. Se da la posibilidad de editar una de las cámaras existentes o agregar una nueva.

Cualquiera sea la opción elegida, se realiza una llamada al mismo archivo de código *PHP*. Si se pasa la variable ID como *parámetro url*, se interpreta que se desea modificar la cámara con dicho ID. Para esto se realiza una nueva lectura en la base de datos, pero solo de la cámara con dicha identificación. La información obtenida se carga en campos editables por el usuario y se da la opción de eliminar la cámara o guardar la información editada.

TRAFFIC VC 03 / 07 / 2017 12 : 44 : 09

Desconexiones: 1 [Cámaras]

Bienvenido: admin (cerrar sesión) Alarmas: 2 Unidades sin atender

### Administrar: Cámaras

Nombre	Host	Descripción	Estado	
Fomento2	192.168.1.2	9 de Julio 3200 - Sur		Editar
Terminal	terminal	Av Ramirez - Sur		Editar
Fomento1	fomento1	9 de Julio 3200 - Norte		Editar
Formosa575	formosa575	Almafuerte al 900		Editar

Nueva Cámara

Ilustración 53 – Aplicación web del servidor local: sección de Administrar - Cámaras

Si se elige la opción de guardar, se actualiza la base de datos con la nueva información y se retorna. Si en cambio se elige eliminar la entrada, dicha información se borra de la base de datos.

Cuando se elige la opción de agregar una nueva cámara, el código es llamado sin identificador ID. En este caso, los campos editables están vacíos y la única opción posible es la de guardar. El usuario debe llenar los campos con la información requerida y al guardar, se agrega una nueva entrada a la base de datos.

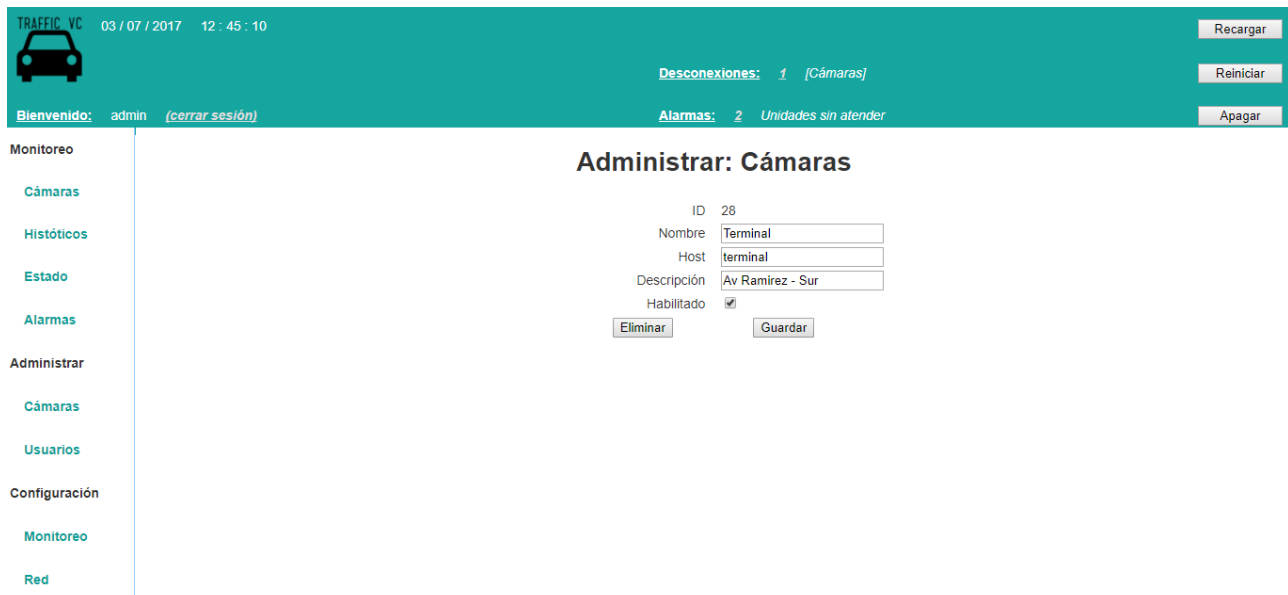


Ilustración 54 – Aplicación web del servidor local: sección de Administrar – Cámaras, editar

Al editar o agregar una nueva cámara se tiene en cuenta que el *hostname* no coincida con ninguno anteriormente agregado, ya que de esta manera existiría una ambigüedad.

Cuando se edite o agregue una nueva cámara, además se llama a ejecución un proceso dentro del remoto. A este proceso se le pasan el *host* correspondiente al servidor y el número de ID asignado a la cámara para ser registrado en la misma. Esta información es utilizada en el proceso de *login*.

#### 2.4.4.7- Administrar: Usuarios

Los usuarios son utilizados por todo el sistema para aprobar o denegar el acceso. Además, los usuarios pueden tener permisos de administrador que les permitan realizar modificaciones en el sistema. Un usuario sin permisos, sólo puede visualizar la información.

La administración de los mismos se realiza de forma parecida a las cámaras. En primer lugar, se listan los mismos y se da la posibilidad de editar cada usuario o crear uno nuevo.

Al elegir la edición, se llama el archivo de edición de usuarios con el parámetro ID pasado por *url*. Teniendo en cuenta este valor, se lee la información desde la base de datos y se coloca en campos editables. El usuario tiene la posibilidad de editar esta información y actualizar así la base de datos. También se da la posibilidad de eliminar el usuario, con lo que se elimina el registro de la base de datos.

Nuevamente, para agregar un usuario nuevo se llama al mismo código, pero sin el parámetro ID pasado como *parámetro url*. Los campos editables están vacíos para ser completados por el usuario.

Usuario	Nombre	Apellido
admin		
mfsoto	María Fernanda	Soto
DiegoMasu	Diego	Massutti
JuanP	Juan	Pighin

Ilustración 55 – Aplicación web del servidor local: sección de Administrar - Usuarios

Al agregar un nuevo usuario, el nombre de Usuario debe ser único, por lo cual se compara con el resto de los usuarios ya registrados, y en caso de ya existir, se retorna un valor de error

Ilustración 56 – Aplicación web del servidor local: sección de Administrar – Usuarios, editar

#### 2.4.4.8- Configuración: Monitoreo

La configuración de monitoreo es la página que permite decidir cuantas cámaras se mostrarán en la sección de *Monitoreo: Cámaras*, como también cuáles serán las mismas.

Para esto, primero se debe elegir la cantidad de cámaras que se van a mostrar. Se muestran en la misma página la misma grilla que se arma en la página de visualización en base a la cantidad de cámaras seleccionadas. Esto se realiza mediante código *javascript*

que muestra u oculta las diferentes grillas disponibles al cambiar el número de cámaras. Cada cuadro de la grilla posee una lista desplegable en la que figuran todas las cámaras disponibles para visualizar.

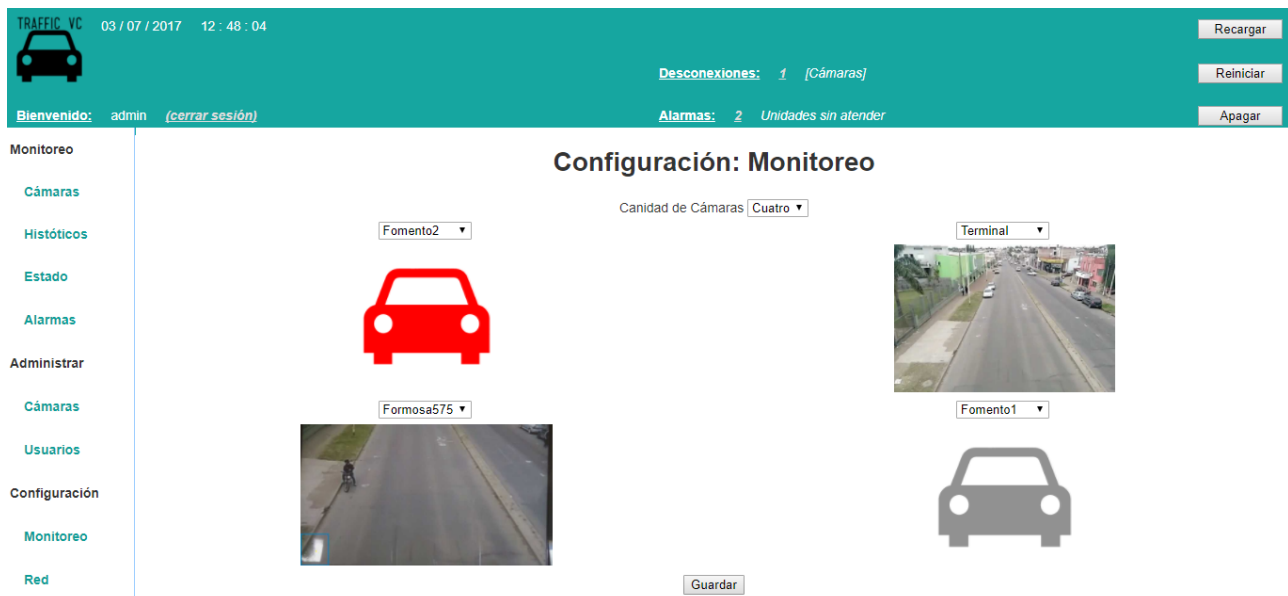


Ilustración 57 – Aplicación web del servidor local: sección de Configuración - Monitoreo

Una vez seleccionada la disposición requerida, se procede a guardar los datos en un archivo de configuración que luego es leído cuando se requiera la página de visualización.

El archivo es un archivo de texto que en la primera línea indica cuántas cámaras se desean mostrar y a continuación, en el orden indicado, el número de ID de cada cámara.

#### 2.4.4.9- Configuración: Red

La configuración de red permite seleccionar para el servidor un *hostname*, con el que se lo identifica dentro de la red, y el tipo de conectividad *LAN* que se utiliza. La conexión *Ethernet* está siempre disponible, pero se agrega la posibilidad de agregar una red *WiFi* para conectividad inalámbrica. Ambas conexiones, además, soportan la posibilidad de establecer una dirección *IP* fija, con su máscara de subred y puerta de enlace.

Esta configuración se realiza modificando los archivos *dhcpcd.config* (dentro del directorio */etc*), *wpa\_supplicant.config* (en el directorio */etc/wpa\_supplicant*), *hostname* y *hosts* (ambos dentro de */etc*).

- El archivo *dhcpcd.config* se utiliza para asignar una dirección de *IP* fija a para cada tipo de conexión agregando ciertas líneas al mismo.
- El archivo *wpa\_supplicant.config* se utiliza para guardar el nombre de la red y la contraseña en caso de que se utilice una red *WiFi*.
- Para configurar el *hostname* se utilizan los otros archivos *hostname* y *hosts*.

TRAFFIC VC 03 / 07 / 2017 12 : 48 : 53

Recargar

Desconexiones: 1 [Cámaras] Reiniciar

Bienvenido: admin (cerrar sesión) Alarmas: 2 Unidades sin atender Apagar

### Configuración: Red

Conexión Ethernet: IP Estática

IP:  .  .  .

Mascara:  .  .  .

Puerta de enlace:  .  .  .

Conexión Wireless: Habilitar  IP Estática

SSID:

Contraseña:

IP:  .  .  .

Mascara:  .  .  .

Puerta de enlace:  .  .  .

Guardar Reiniciar

Monitoreo

Cámaras

Históricos

Estado

Alarmas

Administrar

Cámaras

Usuarios

Configuración

Monitoreo

Red

Ilustración 58 – Aplicación web del servidor local: sección de Configuración - Red

Para que estos archivos puedan ser modificados se requirieron permisos especiales dentro del sistema *Raspbian* ya que no pueden ser modificados por cualquier usuario. El servidor *Apache II* es ejecutado por el usuario *www-data*, por lo cual fue necesario agregar a este usuario al grupo dueño de los archivos y agregar permisos de escritura a dicho grupo.

El proceso comienza cargando la información actual en los campos correspondientes dentro de la página *html*. Luego de que el usuario realice las modificaciones necesarias, debe guardar los datos, para lo cual se ejecuta un nuevo código *PHP*.

Antes de guardar los datos, se realizan ciertas comprobaciones para que las direcciones *IP* sean válidas, como también las máscaras de subred. Si se encuentra algún error, se vuelve a la página de configuración anterior con el mensaje de error correspondiente. Si las comprobaciones son correctas, se procede a sobrescribir los archivos correspondientes con la nueva configuración. Una vez finalizado, se vuelve a la sección anterior con un mensaje de aprobación.

Para que los cambios tengan efecto, se deben realizar ciertos procesos en base a los cambios realizados. Para esto se implementa un botón de *reinicio* que provoca el reinicio de la placa *Raspberry PI*.

#### 2.4.5- Análisis del funcionamiento

La unidad de administración central de datos, o servidor local, se encarga de la administración del sistema. Este dispositivo almacena los usuarios que pueden acceder al sistema y aprueba su ingreso, con lo que es indispensable para poder analizar los datos, aunque el dispositivo de medición no requiere conexión con éste para funcionar. También permite la visualización de hasta 6 cámaras a la vez en la misma pantalla.

Cuenta con un servidor *Apache II* que permite conectividad *html* dentro de la red en la que se embebe el sistema, de modo que toda la administración y visualización se realiza en forma de página web. El dispositivo posee conectividad *Ethernet* o *Wifi*, y por defecto la asignación de la dirección *IP* se realiza mediante *dhcp* por el puerto *Ethernet* con el nombre de dominio “*servidor*”. El tipo de conectividad es configurable, así como la posibilidad de asignar una *IP* estática y cambiar el *hostname*.

Se utiliza una base de datos *MySQL* para almacenar tanto las cámaras que integran al sistema como los usuarios que tienen acceso al mismo. La base de datos se llama “Proyecto” y las tablas correspondientes “Camaras” y “Usuarios”.

Los usuarios se almacenan en la tabla “Usuarios” identificados por un “Usuario”, identificador único para cada entrada, con el requerimiento de una “Contraseña”. Los campos “Nombre” y “Apellido” son opcionales. A cada uno se le asigna automáticamente un número de “ID” único utilizado para identificarlo. Además, se debe establecer si el usuario tiene privilegios de “Administrador” o no, lo cual le permite (o niega) realizar cambios en el sistema. Usuarios sin privilegios, sólo tienen permitida la visualización de los datos, no pueden realizar modificaciones al sistema. Un esquema de la tabla se puede ver a continuación:

<b>ID</b> (Identificador único - Autogenerado)	<b>Usuario</b> (Identificador único)	<b>Contraseña</b> (Elegido por el usuario)	<b>Nombre</b> (Opcional)	<b>Apellido</b> (Opcional)	<b>Administrador</b> (SI/NO)

Tabla 4 – Base de Datos Local - Usuarios

Las cámaras (dispositivos de medición) que integraran el sistema se guardan en la tabla “Camaras” y se identifican por un “Nombre”. Se debe indicar un “Host” que puede ser tanto una dirección *IP* como un nombre de dominio (*hostname*). Este campo debe corresponder con el seleccionado en el dispositivo de medición ya que es la forma de reconocerlo dentro de la red. El campo “Descripcion” está pensado para agregar un breve comentario respecto de la función o ubicación. El sistema cuenta con la capacidad de deshabilitar cámaras con la intención de desactivar el monitoreo sobre las mismas. En caso de que una cámara se encuentre deshabilitada, si la misma es elegida para ser mostrada en la página de visualización no se muestra la imagen de la misma. El control del estado y las alarmas queda también desactivado en este caso.

<b>ID</b> (Identificador único - Autogenerado)	<b>Nombre</b> (Identificador)	<b>Host</b> (Identificador dentro de la red)	<b>Descripcion</b> (Función o ubicación)	<b>Habilitado</b> (SI/NO)

Tabla 5 – Base de Datos Local - Cámaras

La gestión de usuarios involucra la creación, edición y eliminación de los mismos. También se encarga de la identificación mediante un sistema de *login*. Los usuarios que



deseen interactuar con el sistema deben identificarse con un nombre de usuario y una contraseña. Esto les permitirá visualizar los distintos parámetros que se encuentran a disposición y, sólo en caso de que sea usuario administrador, realizar modificaciones de configuración.

El sistema de identificación alojado en el servidor local, también es utilizado para ingresar al dispositivo remoto. Al iniciar una comunicación con el dispositivo remoto, se consulta el sistema de *login* del servidor local y, si éste lo aprueba, la conexión se finaliza. Esto implica que el servidor local deba estar encendido para poder visualizar los datos, pero no es necesario para que la cámara funcione y haga las mediciones pertinentes. Es menester, a su vez, que la cámara esté registrada en el servidor para que éste apruebe la conexión de un usuario.

En cuanto a la gestión de cámaras, se guardan los datos de *host* para poder acceder a las mismas y realizar las visualizaciones simultáneas. En este caso también se pueden agregar, editar y eliminar. Además, se tiene la posibilidad de extraer estadísticas directamente desde la cámara, sin necesidad de acceder a la misma.

Un sistema de monitoreo proactivo advierte en caso de que se pierda conexión con alguno de los dispositivos de medición, como también en caso de que alguno presente alarmas sin atender.

## 2.5- Diseño Completo

El diseño final del sistema consta de dos partes independientes: un dispositivo que se ubicará en lo posible junto a un semáforo y con la cámara filmará la vía de tránsito de vehículos, éste se conecta a una red LAN; otro dispositivo funcionará como servidor y estará ubicado probablemente en una central de monitoreo, también debe conectarse a la misma red.

La interfaz de usuario consiste en una aplicación web a la cual deberá ingresarse con “usuario” y “contraseña”, existiendo distintos niveles de usuarios con permisos diferentes. En esta aplicación es posible realizar configuraciones referentes a los dispositivos ubicados en las calles, a la red, a los usuarios, a la visualización de la información, entre otras. Además, se observa el video en tiempo real de las vías de tránsito, permitiendo hasta seis cámaras a la vez. En esta página pueden verse las alarmas que se están produciendo en el momento. Si se desea, se puede ingresar a la aplicación web de cada dispositivo y así obtener información de cada uno, la cantidad de vehículos que están circulando, el flujo instantáneo actual, descargar históricos y tablas, visualizar gráficos, configurar zonas, reiniciar el procesamiento, entre otras funciones.

La versatilidad otorgada por la aplicación web es una de las principales características de este sistema ya que no requiere de la instalación de un software ni la descarga de actualizaciones.

A continuación, se muestra una foto del dispositivos de medición.



Ilustración 59 – Dispositivo de medición - Hardware

## Capítulo 3: Resultados

Se trata de un sistema capaz de:

- Monitorear el tránsito en tiempo real desde una central a través de una red ya configurada mediante una aplicación web.
- Crear distintos usuarios con permisos diferentes.
- Configurar las cámaras disponibles y agregar nuevas al sistema.
- Monitorear varias cámaras a la vez o ingresar directamente a la interfaz de una de ellas para conocer más detalles.
- Contar la cantidad de vehículos que circulan (obtener un flujo vehicular) por una vía de tránsito de una mano.
- Identificar detenciones indebidas sobre la calle, y a la vez identificar detenciones sobre el área de estacionamiento. Obtener alarmas sobre estos eventos.
- Identificar flujo excesivo y flujo crítico de vehículos. Obtener alarmas sobre estos eventos.
- Identificar vehículos en contramano.
- Iniciar, detener o reiniciar el procesamiento de imágenes.
- Tomar capturas de imagen cuando se producen eventos críticos.
- Configurar parámetros relativos a las prioridades de las alarmas.
- Configurar zonas en las calles.
- Configurar *hostname*, dirección IP, conexión *wireless*, entre otros, del dispositivo de medición.
- Obtener históricos de las alarmas, visualizar gráficas referidas a esta información.
- Obtener históricos de la cantidad de vehículos que han circulado, visualizar gráficas con esta información.

Para conocer los resultados que arroja el dispositivo en el procesamiento, se grabaron videos del tránsito con las condiciones necesarias de instalación (altura, inclinación y orientación) para simular un normal funcionamiento de la cámara. De esta manera se pudo obtener el error que tiene en el conteo de vehículos, cotejando con la cantidad real que circulan en cada video.

En la siguiente tabla se muestran los valores obtenidos.

<b>Video</b>	<b>Duración</b>	<b>Cantidad real de vehículos</b>	<b>Cantidad contada de vehículos</b>	<b>Error absoluto</b>	<b>Error relativo (%)</b>
1	03:05	78	76	-2	-2,56
2	04:37	52	53	1	1,92
3	04:02	77	73	-4	-5,19
4	03:16	49	45	-4	-8,16
5	02:23	33	32	-1	-3,03
6	03:40	45	45	0	0,00
7	05:11	84	81	-3	-3,57
8	03:26	61	59	-2	-3,28
9	04:10	70	65	-5	-7,14
10	02:31	25	25	0	0,00
<b>Error promedio</b>				-2	-3,10
<b>Desviación Estándar</b>				0,918936583	1,443782814

Tabla 6 – Resultados de mediciones

Para conocer el funcionamiento relacionado con la detección de alarmas, se utilizaron los mismos videos comentados anteriormente, en los cuales se aprecian detenciones de vehículos y momentos en los que se da un flujo excesivo y flujo peligroso.

## Capítulo 4: Análisis de Costos

A continuación, se detallan los costos que llevaron a la realización de este proyecto:

<b>Recursos materiales</b>			
<b>Insumo</b>	<b>Cantidad</b>	<b>Costo / u (\$)</b>	<b>Total (\$)</b>
Raspberry Pi 3 + Alimentación + Carcasa + Disipadores + Tarjeta micro SD	3	1.500	4.500
Módulo Cámara de video	1	879	879
Caja estanco - dispositivo de medición	1	49	49
Regulador de tensión	1	100	100
Conectores y otros insumos	-	50	50
Cables de red	3	25	75
<i>Total</i>			\$5.653

Tabla 7 – Costos de recursos materiales

<b>Recursos humanos</b>			
<b>Concepto</b>	<b>Cantidad</b>	<b>Costo / ing (\$)</b>	<b>Total (\$)</b>
Ingenio	640	200	128.000
Ingenio	640	200	128.000
Ingenio	640	200	128.000
<i>Total</i>	1.920		\$384.000

Tabla 8 – Costos de recursos humanos

Para el cálculo del costo de horas dedicadas de trabajo, se tomó en cuenta el costo del ingenio y los honorarios provistos por el Colegio de Ingenieros Especialistas de Entre Ríos (CIEER)<sup>6</sup>. El trabajo llevó un tiempo de realización de 8 meses. El costo total del proyecto es de \$389.653.

Un posible plan de venta es a la municipalidad de Paraná. En este caso, se vendería un solo dispositivo de administración y varios de sensado de imagen (tantos como calles se deseen monitorear). El producto se podría llegar a vender en otros municipios en una cantidad suficiente para amortizar los costos de ingenio y conseguir un precio de venta medio.

<sup>6</sup> En el sitio web del CIEER se encuentra la información disponible: <http://www.cieer.org.ar>

## **Capítulo 5:       Discusión y Conclusión.**

El sector del mercado donde se ubicará el producto está precariamente desarrollado en el país, lo que brindaría una gran oportunidad para introducirlo de manera satisfactoria en el mismo. Existen en el mercado nacional escasos equipos para la medición de tránsito y estos difieren del presentado en este proyecto en varios aspectos.

Algunos de ellos utilizan otro tipo de sensores como por ejemplo, piezoeléctricos, de manguera o espiras inductivas directamente sobre el piso de la carretera. Éstos por lo general están orientados a mediciones en rutas o autopistas, y presentan mayor facilidad de deterioro o rotura y a su vez resultan más invasivos.

Existe además en Argentina, un módulo de procesamiento a partir de cámaras, que permite conectar varias de ellas centralizando la configuración de las mismas. Esto genera una dependencia de varios sensores a un mismo módulo de control, haciendo que el sistema total esté sujeto a posibles fallas del controlador. Lo cual no ocurriría con este sistema presentado, ya que la independencia entre los dispositivos es una de sus principales virtudes.

De esta manera, el resultado obtenido con este proyecto es un producto que posiblemente se introduzca satisfactoriamente en el mercado, con prestaciones favorables y mejores a la competencia y con un costo medio.

Cumple con los objetivos planteados al principio: conseguir un sistema capaz de monitorear el tránsito, contabilizar vehículos e identificar lentitud en el tránsito. No sólo se ha logrado cumplir con esos objetivos, sino que también se ha añadido valor al contar con una aplicación web, solicitar históricos, ver gráficos, obtener capturas del video en momentos críticos, entre otras prestaciones. Es necesario recordar que el propósito de este sistema es ayudar en la planificación y control del tránsito de las ciudades. Se ha diseñado de tal manera que permita crear una central de monitoreo donde existan operadores capacitados para observar los parámetros obtenidos y actuar en consecuencia, por ejemplo extendiendo el tiempo de un semáforo si el flujo vehicular es excesivo.

Debido a que es un sistema con una interfaz de usuario capaz de mejorarse y/o crecer tanto como necesidades se presenten, se han planteado algunas posibles mejoras a futuro:

- Capacidad para almacenar video y que actúe además como sistema de vigilancia.
- Accionar directamente sobre los tiempos del semáforo cuando el sistema considere necesario.
- Clasificación de vehículos según tamaño.
- Identificar patentes.
- Funcionamiento para calles de doble vía.

Además, sería una buena implementación la adaptación del procesamiento de imagen a las cámaras IP instaladas en la ciudad o a cualquier otra cámara disponible. En este proyecto, esto puede realizarse, pero es necesario efectuar cambios para que así funcione.

Al iniciar este proyecto, se produjeron varios inconvenientes al momento de configurar la *Raspberry*. Durante la instalación del servidor y la base de datos resultó necesario realizar varios pasos para que funcione correctamente, lo que llevó un tiempo considerable. Una vez instalados, surgieron problemas relacionados con los permisos de escritura/lectura de *Raspbian* para ciertos directorios y/o archivos.

También se presentó la necesidad de aprender nuevos lenguajes de programación: *html*, *python*, *php*, *javascript*. Para esto fue necesario investigar y aprender varias funciones a medida que se desarrollaba el proyecto.

Otra dificultad que surgió fue encontrar una resolución de video que permita buena calidad para su visualización y que a la vez tenga un tamaño considerable para realizar el procesamiento sin saltos ni demoras.

## Capítulo 6: Bibliografía Consultada.

- Teoría sobre el flujo vehicular:

Mozo Sánchez, J. (2012). *Capítulo 1: Teoría del flujo vehicular* (Tesis de grado). Universidad Nacional Autónoma de México. México. Recuperado de: <http://132.248.52.100:8080/xmlui/handle/132.248.52.100/417>

- Curso de PHP/MySQL.

Gomez, Juan [pildorasinformaticas]. (2015). *Curso de PHP/MySQL*. Recuperado de: <https://www.youtube.com/playlist?list=PLU8oAIHdN5BkinrODGXToK9oPAInJxmW>

- Documentación sobre librerías y lenguajes de programación, datasheets:

Chart.js. <https://www.chartjs.org/>

OpenCV Docs. <http://docs.opencv.org/master/index.html>

PHP. <https://www.php.net/>

RPi Cam Web Interface. <http://elinux.org/RPi-Cam-Web-Interface>

Raspberry Pi. [http://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1\\_0.pdf](http://www.raspberrypi.org/documentation/hardware/computemodule/RPI-CM-DATASHEET-V1_0.pdf)

w3schools. <http://www.w3schools.com/>

geekytheory. <https://geekytheory.com/>



**Grilla Evaluación:**

	Puntaje Máximo	Puntaje asignado
Complejidad	0 – 30 pts	
Innovación - Aplicabilidad	0 – 30 pts	
Equipo	0 – 20 pts	
Clase Pública	0 – 20 pts	
Total		

**Pañoni Sergio****Ramos Hector****Maggiolini Lucas****Docente Director:**

Universidad Tecnológica Nacional  
Facultad Regional Paraná

Se certifica que ..... , DNI: ..... ha realizado la  
dirección del Proyecto Final:

.....  
.....  
.....

De los alumnos:

- .....
- .....
- .....

Realizada durante el ciclo lectivo: ....., obteniendo el grupo un calificación final de:

.....

A fin de ser emitida la correspondiente certificación por el departamento de electrónica, se extiende  
la siguiente constancia.

**Pañoni Sergio**

**Ramos Hector**

**Maggiolini Lucas**