



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Electrónica Industrial y Automática**

**Sistema inteligente de gestión de tráfico**

**Autor:**

**González Caballero, Javier**

**Tutor:**

**González de la Fuente, José  
Manuel**

**Tecnología Electrónica**

**Valladolid, julio de 2019.**





## RESUMEN

El aumento del tránsito de vehículos tanto públicos como privados hace necesario un cambio en la forma de organizar tanto la secuencia de los semáforos como el tiempo en el que permiten o no el paso de automóviles. En este proyecto se propone un sistema alternativo capaz de modificar en tiempo real la duración en verde o en rojo de los semáforos según el flujo de vehículos y la presencia o no de peatones. Para ello, se analiza la tecnología de control de los semáforos que se emplea actualmente y las investigaciones realizadas. Además, se realiza un estudio de la ingeniería del tráfico, de los componentes hardware que existen en el mercado, del tratamiento de imágenes con visión artificial y de la comunicación entre programas que se encuentran en el mismo o en diferentes máquinas. A continuación, se desarrolla un sistema capaz de controlar y analizar el flujo de vehículos mediante el tratamiento de imágenes con visión artificial y, por último, se evalúa el algoritmo sobre secuencias de imágenes.

### Palabras clave

Estudio del tráfico, semáforos inteligentes, análisis de video, detección de objetos en movimiento, control de procesos, smart cities.





## ABSTRACT

The increase in the number of vehicles make necessary a change in the way of organize the sequence of the traffic lights and in the time of passage of the vehicles. The propose of this project is generate an alternative system capable of modify in real time the duration of the traffic lights according to the traffic flow attending the presence of pedestrians. This estudy analyze the current tecnologia of the traffic lights and the researches carried out. Furthermore, the traffic engineering, the components hardware that there are in the market, the treatment of the images with artificial visión and the comunicacion between programs that could be in the same or in different machines, will be analyzed. The programs control and analyze the traffic flow thanks to the treatment of images with artificial visión. The check of the program's algorithm is evaluated with sequences of images.

### Keywords

Study of traffic, intelligence traffic lights, video analysis, detection of moving objects, control of processes, smart cities.





A mis padres por su apoyo a lo  
largo de estos años





## ÍNDICE

1.	INTRODUCCIÓN .....	3
1.1	JUSTIFICACIÓN DEL PROYECTO .....	3
1.2	OBJETIVOS .....	4
1.2.1	SITUACIÓN ACTUAL DE LA TECNOLOGÍA.....	4
1.2.2	HARDWARE .....	4
1.2.3	SOFTWARE.....	4
1.2.4	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.....	5
1.3	ESTRUCTURA DE LA MEMORIA.....	6
2.	FUNDAMENTOS TEÓRICOS .....	9
2.1	TECNOLOGÍAS ACTUALES E INVESTIGACIONES.....	9
2.2	INGENIERÍA DE TRÁFICO.....	11
2.2.1	PROPÓSITO DE LA REGULARIZACIÓN SEMAFÓRICA.....	11
2.2.2	ELEMENTOS CONSTITUTIVOS DE LOS SEMÁFOROS .....	12
2.2.3	CONCEPTOS PARA LA REGULACIÓN SEMAFÓRICA .....	13
2.2.4	CÁLCULO DEL CICLO SEMAFÓRICO Y DE REPARTO .....	14
2.3	HARDWARE .....	16
2.3.1	SENSORES.....	16
2.3.1.1	FOTOELÉCTRICOS.....	16
2.3.1.2	ELECTROMAGNÉTICOS .....	18
2.3.2	CÁMARAS DE VIDEOVIGILANCIA.....	19
2.3.3	MICROPROCESADORES Y MICROCONTROLADORES.....	21
2.3.4	SISTEMAS DE COMUNICACIÓN .....	22
2.3.4.1	CABLES ELÉCTRICOS .....	23
2.3.4.2	CABLES ETHERNET .....	23
2.3.4.3	MÓDULOS WIFI.....	24
2.3.4.4	MÓDULOS GSM .....	25
2.4	SOFTWARE .....	26
2.4.1	VISIÓN ARTIFICIAL.....	26
2.4.1.1	OBTENCIÓN IMAGEN Y TRANSFORMACIÓN DIGITAL.....	27
2.4.1.2	PROCESAMIENTO IMÁGENES: FILTRO Y TRANSFORMACIONES GEOMÉTRICAS.....	30
2.4.1.3	SEGMENTACIÓN .....	34



2.4.1.4	CLASIFICACIÓN .....	36
2.4.2	COMUNICACIÓN ENTRE PROCESOS.....	38
2.4.2.1	COMUNICACIÓN CLIENTE-SERVIDOR.....	39
2.4.2.2	MEMORIA COMPARTIDA EN SYSTEM V .....	40
2.4.2.3	SECCIONES CRÍTICAS Y SEMÁFOROS .....	42
2.4.2.4	SEÑALES.....	44
2.4.2.5	SOCKETS.....	46
2.5	EL INTERNET DE LAS COSAS .....	50
2.5.1	HARDWARE .....	52
2.5.2	LENGUAJES DE DESARROLLO .....	55
2.5.3	PLATAFORMAS.....	56
2.5.4	APIs.....	59
2.5.5	CLOUD COMPUTING .....	60
3.	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....	65
3.1	ARQUITECTURA DEL SISTEMA.....	65
3.1.1	SELECCIÓN DE DISPOSITIVOS .....	65
3.1.2	UBICACIÓN DE DISPOSITIVOS.....	68
3.2	FUNCIONAMIENTO GENERAL DEL SISTEMA .....	68
3.2.1	PUESTA EN MARCHA .....	69
3.2.2	EJECUCIÓN CONTINUA.....	71
3.3	DIAGRAMA DE FUNCIONAMIENTO DEL SISTEMA.....	73
3.4	IMPLEMENTACIÓN DEL SISTEMA.....	78
3.4.1	PROGRAMA CCT.....	78
3.4.2	PROGRAMA IOT.....	80
3.4.3	PROGRAMA SCT.....	81
3.4.4	PROGRAMA SIV .....	92
3.4.5	PROGRAMA SIP.....	104
4	RESULTADOS EXPERIMENTALES.....	121
4.1	PROGRAMA SCT .....	123
4.2	PROGRAMA IOT Y THINGSPEAK.....	141
4.3	PROGRAMA CCT .....	144
4.4	Programa SIV.....	147



4.5	PROGRAMA SIP .....	163
5	CONCLUSIONES .....	177
5.1	TRABAJO FUTURO .....	178
6.	BIBLIOGRAFÍA.....	181





## ÍNDICE DE FOTOGRAFÍAS

Fotografía 1. Sensor receptor UK0ARCTL2 .....	17
Fotografía 2. Reflector XUZC50 .....	17
Fotografía 3. Sensor emisor XUK0ARCTL2T .....	17
Fotografía 4. Sensor receptor XUB0APSNM12 .....	18
Fotografía 5. Sensor emisor XUB0BKSNM12T .....	18
Fotografía 6. Sensor emisor-receptor XUK1ARCNL2 .....	18
Fotografía 7. Sensor lazo inductivo LDP1SA1BM24.....	19
Fotografía 8. Sensor lazo inductivo Matrix .....	19
Fotografía 9. Cámara IP Foscam FI9804W .....	20
Fotografía 10. Cámara IP Foscam FI9805E .....	20
Fotografía 11. Cámara IP Dahua X-Security .....	20
Fotografía 12. Cámara IP Nivian ONV524 .....	20
Fotografía 13. Arduino DUE .....	21
Fotografía 14. Arduino GENUINO YÚN SHIELD.....	21
Fotografía 15. Raspberry Pi 3 Model B+.....	22
Fotografía 16. Raspberry Pi Zero W .....	22
Fotografía 17. Raspberry Pi Zero.....	22
Fotografía 18. Cable RS PRO .....	23
Fotografía 19. Cable TE Connectivity .....	23
Fotografía 20. Cable Ethernet OEM .....	24
Fotografía 21. Cable Ethernet Digitus.....	24
Fotografía 22. ESP8266EX .....	25
Fotografía 23. ESP32 .....	25
Fotografía 24. Módulo 4G para Raspberry Pi.....	25
Fotografía 25. Módulo GPRS para Raspberry Pi .....	26
Fotografía 26. Módulo GPRS para Arduino.....	26
Fotografía 27. Esquema eléctrico para la simulación del programa SCT .....	131





## ÍNDICE DE FIGURAS

Figura 1. Trayectoria que recorre el haz de luz al atravesar la lente biconvexa .....	28
Figura 2. Los 8 vecinos del píxel p .....	29
Figura 3. Distancia euclídea y distancia Manhattan.....	30
Figura 4. Elemento estructural A, elemento a dilatar B y resultado final.....	33
Figura 5. Elemento estructural A, elemento a erosionar B y resultado final.....	33
Figura 6. Imagen original. Imagen con apertura .....	33
Figura 7. Imagen original. Imagen con cierre .....	34
Figura 8. Diagrama flujo control sockets .....	48
Figura 9. Diagrama puesta en marcha .....	74
Figura 10. Diagrama ejecución continua.....	76
Figura 11. Diagrama flujo programa CCT .....	79
Figura 12. Diagrama flujo programa IOT.....	81
Figura 13. Diagrama flujo programa SCT .....	82
Figura 14. Diagrama de flujo modificación tiempos de fases.....	90
Figura 15. Diagrama flujo programa SIV.....	93
Figura 16. Diagrama flujo programa SIP .....	105
Figura 17. Circuito simulación programa SCT .....	129
Figura 18. Ecuación de la recta del carril izquierdo.....	155
Figura 19. Ecuación de la recta del carril central .....	156
Figura 20. Ecuación de la recta del carril derecho .....	157
Figura 21. Píxeles captados por la cámara situada a 2,4 metros .....	168
Figura 22. Triángulo que permite obtener el número de píxeles para 2 metros de longitud.....	168





## ÍNDICE DE TABLAS

Tabla 1. Señales de control de trabajo .....	45
Tabla 2. Tarifas plataforma Samsung Artik Cloud.....	57
Tabla 3. Tarifas plataforma aREST.....	57
Tabla 4. Tarifas plataforma Thinger.io .....	58
Tabla 5. Tarifas plataforma ThingSpeak .....	59
Tabla 6. Flujos de saturación iniciales óptimos .....	124
Tabla 7. Tiempos de fase óptimos .....	126
Tabla 8. Tiempos de fase máximos.....	127
Tabla 9. Flujos de saturación para $\rho_1 = 0\%$ , $\rho_2 = 0\%$ y $\rho_3 = 28\%$ .....	132
Tabla 10. Tiempos de fase óptimos para $\rho_1 = 0\%$ , $\rho_2 = 0\%$ y $\rho_3 = 28\%$ .....	133
Tabla 11. Flujos de saturación para $\rho_1 = 3\%$ , $\rho_2 = 4\%$ y $\rho_3 = 16\%$ .....	135
Tabla 12. Tiempos de fase óptimos para $\rho_1 = 3\%$ , $\rho_2 = 4\%$ y $\rho_3 = 16\%$ .....	136
Tabla 13. Tiempos de fase óptimos para $\rho_1 = 3\%$ , $\rho_2 = 4\%$ y $\rho_3 = 16\%$ con peatones en Fase 2 y 3.....	138
Tabla 14. Flujos de saturación para $\rho_1 = 0\%$ , $\rho_2 = 0\%$ y $\rho_3 = 0\%$ .....	139
Tabla 15. Tiempos de fase óptimos para $\rho_1 = 0\%$ , $\rho_2 = 0\%$ y $\rho_3 = 0\%$ .....	140
Tabla 16. Tiempos de fase óptimos para $\rho_1 = 0\%$ , $\rho_2 = 0\%$ y $\rho_3 = 0\%$ con peatones en Fase 2 y 3 .....	141
Tabla 17. Ajuste por mínimos cuadrados carril izquierdo .....	154
Tabla 18. Ajuste por mínimos cuadrados carril central .....	155
Tabla 19. Ajuste por mínimos cuadrados carril derecho.....	156
Tabla 20. Áreas para 4 coordenadas Y carril izquierdo.....	157
Tabla 21. Áreas para 4 coordenadas Y carril central .....	158
Tabla 22. Áreas para 4 coordenadas Y carril derecho .....	158
Tabla 23. Peso píxeles por fila.....	159





## ÍNDICE DE ECUACIONES

Ecuación 1. Flujo equivalente de automóviles .....	15
Ecuación 2. Flujo de saturación .....	15
Ecuación 3. Tiempo pérdidas .....	15
Ecuación 4. Tiempo de ciclo .....	16
Ecuación 5. Tiempo semáforo verde.....	16
Ecuación 6. Tiempo semáforo rojo.....	16
Ecuación 7. Distancia euclídea .....	29
Ecuación 8. Distancia Manhattan .....	30
Ecuación 9. Matriz de traslación .....	31
Ecuación 10. Matriz de escalado .....	31
Ecuación 11. Matriz de rotación.....	31
Ecuación 12. Densidad ocupación.....	71
Ecuación 13. Flujo de saturación para garantizar el tiempo de cruce de los peatones .....	84
Ecuación 14. Flujo de saturación máximo.....	85
Ecuación 15. Movimiento rectilíneo uniforme, MRU .....	136
Ecuación 16. Pendiente de una recta.....	154
Ecuación 17. Intersección en y.....	154





# CAPÍTULO 1

## INTRODUCCIÓN





## 1. INTRODUCCIÓN

### 1.1 JUSTIFICACIÓN DEL PROYECTO

Desde hace bastante tiempo, el parque automovilístico ha experimentado un fuerte crecimiento originando problemas de congestión de tráfico en ciudades. La congestión de tráfico, además de ser una pérdida de tiempo para los vehículos, también implica un aumento en el gasto de combustible, una mayor contaminación, un mayor riesgo de accidentes y un incremento del estrés de los conductores.

El impacto económico de la congestión, no solo afecta al aumento de combustible, sino también al tiempo en que tardan en llegar las mercancías a su destino y por ello, a la competitividad de la empresa. Para paliar este problema, se eleva el precio de los productos para costear las pérdidas originadas por la congestiones.

La congestión del tráfico supone, según un estudio realizado por la Universidad de Barcelona y la Universidad Rovira i Virgili <sup>[1]</sup>, una pérdida de tiempo de 119 horas al año, 14 días laborables, en Barcelona y de 105 horas al año, 13 días laborables, en Madrid. Estas pérdidas de tiempo, conllevan un coste económico de 175,5 millones de euros en Barcelona y de 187,5 millones de euros en Madrid.

Los atascos no solo suponen costes económicos, sino también ambientales, por lo que afecta a la salud. La Agencia Europea de Medio Ambiente (AEMA) en un informe elaborado en julio de 2017 <sup>[2]</sup>, dicto que se producían en España 31520 fallecimientos al año por contaminación, sobre todo debido a las partículas en suspensión, dióxido de nitrógeno.

Una correcta regulación del tráfico contribuiría a reducir las retenciones en las ciudades ya que si los tiempos en que los semáforos permiten o no el paso de los vehículos son insuficientes o excesivos, se originarían atascos.

Actualmente, el tráfico se maneja con semáforos controlados por tiempo. Este tiempo, se calcula a partir de la intensidad de vehículos que pasan por él en un tiempo concreto. Debido a ello, se da el caso de que pueda haber más o menos intensidad de vehículos en otra franja horaria o en otros días. Otro caso que se puede dar y que afecta tanto al aumento de combustible, a la contaminación, o al estrés de los conductores, es la espera en un semáforo en rojo sin que haya vehículos en el carril que se encuentra en verde.

Los problemas descritos anteriormente hacen necesario un cambio en la forma de manejar los semáforos para que estos sean más eficientes y disminuyan los problemas expuestos. Para ello, se realiza el estudio y la implantación de un sistema capaz de controlar el tráfico en las intersecciones según el flujo de vehículos y la presencia de peatones.



## 1.2 OBJETIVOS

El objetivo de este proyecto se basa en el estudio y desarrollo de un sistema capaz de regular los semáforos que constituyen una intersección. El tiempo en que cada semáforo está en verde, se establece a partir del flujo de vehículos que hay en ese instante y en la presencia o no de peatones.

Para la realización de este proyecto, se efectúa un estudio de los diferentes métodos de análisis de tráfico que existen en la actualidad, los elementos hardware disponibles en el mercado, el software que se emplea y el diseño e implantación del sistema.

### 1.2.1 SITUACIÓN ACTUAL DE LA TECNOLOGÍA

En el desarrollo de un proyecto, en primer lugar, se realiza un análisis del estado del arte. En este caso, se revisarán los sistemas de tráfico actuales, las investigaciones relacionadas con la gestión del tráfico y el marco teórico correspondiente a cada etapa del proyecto.

### 1.2.2 HARDWARE

La implantación de este proyecto, se puede realizar mediante sensores o cámaras de video vigilancia que detecten vehículos y personas. Además, se necesitarán componentes capaces de transmitir información, tanto de forma interna como externa (IOT), y de permitir la salida de señales digitales. En la actualidad, hay en el mercado una gran cantidad de dispositivos que realizan las funciones descritas. Para ello, se efectuara un estudio de características y de mercado para así obtener un producto final fiable y rentable, es decir, competitivo.

La transmisión de información entre los distintos dispositivos hardware, se puede realizar mediante cable o señal WiFi. Se analizará tanto su coste como su fiabilidad para que el producto sea lo más viable posible.

### 1.2.3 SOFTWARE

El tratamiento de la información recibida por los elementos hardware escogidos para medir la afluencia de vehículos y la presencia o no de peatones, se puede realizar con diferentes programas, tanto gratuitos como de pago, por lo que se estudiarán las ventajas y los inconvenientes de cada uno.



En el caso de que se emplee el uso de cámaras para obtener la afluencia de vehículos y la presencia o no de peatones, se implementarían algoritmos que permitan el tratamiento de las imágenes extraídas del vídeo.

#### 1.2.4 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

La detección de vehículos y peatones, se puede realizar por sensores o por cámaras.

En el caso de que la detección se realizase con sensores, se diseñaría un circuito capaz de captar el paso de los vehículos y la presencia de los peatones. Este circuito se conectaría a un microprocesador que permitiría adaptar el tiempo en verde de los semáforos mediante un software instalado en él. Además, sería el encargado de cambiar los discos de los semáforos mediante el envío de una señal a los dispositivos hardware instalados en ellos.

Si se utilizan cámaras de videovigilancia, la detección de vehículos y la presencia de peatones, se realizaría mediante software por lo que no se diseñaría ningún circuito electrónico para su detección. El software, a partir de las imágenes que se obtendrían de las cámaras, adquiriría la región de interés de la carretera y de los peatones. De esta forma, se obtendría la afluencia de vehículos y la presencia de peatones para el control de los semáforos. Las cámaras se podrían conectar al microprocesador por cable o mediante señal WiFi,

Para el desarrollo del software si se realiza el proyecto mediante cámaras, se emplearían vídeos y fotografías expuestas en internet. En el caso de que fuese necesario, se realizarían vídeos si no se dispusiesen en internet o sus características no fuesen aptas para el funcionamiento correcto del sistema.

Tanto si se emplean sensores como si se emplean cámaras, se estudiarán los casos en que sea necesario incrementar o disminuir el tiempo de los semáforos en verde y en rojo, según el flujo de los vehículos y la presencia o no de peatones. Además, se permitirá la conexión, a través de internet, IOT, del sistema para que se permita desde un centro de control, cambiar el semáforo activo cuando lo disponga el operario.

Para la puesta en marcha del sistema y su funcionamiento continuo, los dispositivos de medición y detección de personas, se ubicarían en determinados lugares que permitan su correcto funcionamiento. Además, se estudiará el emplazamiento del microprocesador para disminuir costes en el caso de que se conecten los elementos hardware por cable o para garantizar el alcance de la señal WiFi a todos los dispositivos.

La característica principal de cualquier proyecto, además de ser la fiabilidad, es la rentabilidad. Debido a ello, el criterio de elección de los dispositivos que se



empleen se basará en las características que poseen y en la viabilidad de su instalación y mantenimiento.

### 1.3 ESTRUCTURA DE LA MEMORIA

La memoria se compone de los capítulos dispuestos a continuación.

- Capítulo 1. Introducción

Descripción de las causas que han propiciado la realización de este proyecto, los objetivos que se desean alcanzar con respecto a su desarrollo e implantación y la disposición de los diferentes capítulos que constituye la memoria

- Capítulo 2. Fundamentos teóricos

Estudio de la tecnología que se emplea actualmente, las investigaciones realizadas en el ámbito de los sistemas de gestión de tráfico y el marco teórico necesario tanto para el desarrollo del sistema como para su implantación.

- Capítulo 3. Diseño e implementación del sistema

Elección y ubicación de los dispositivos hardware, explicación del funcionamiento general del sistema, tanto su puesta en marcha como su ejecución continua, flujo de datos entre procesos, y desarrollo del algoritmo de cada programa.

- Capítulo 4. Resultados experimentales

Evaluación de las diferentes fases que constituye la puesta en marcha y la ejecución continua de los diferentes programas.

- Capítulo 5. Conclusiones

Objetivos desarrollados y mejoras que se pueden aplicar al proyecto

- Capítulo 6. Bibliografía



# CAPÍTULO 2

## FUNDAMENTOS TEÓRICOS





## 2. FUNDAMENTOS TEÓRICOS

### 2.1 TECNOLOGÍAS ACTUALES E INVESTIGACIONES

En la actualidad, el tiempo de encendido de los distintos discos del semáforo, se controla por tiempo fijo. Este tiempo, dependiendo de la ciudad, puede variar entre la mañana, la tarde y la noche. Este sistema no es eficiente para el tráfico actual ya que no analiza el flujo de tráfico que hay en el momento, sino que depende del día y la hora en que el operario haya tomado el número de vehículos que pasan por el cruce. El tipo de tecnología empleada en la regularización semafórica no permite la conexión a internet de los semáforos que constituyen una intersección, por lo no se puede cambiar los discos de los semáforos en el caso de que un operario, ubicado en un centro de control, lo considere oportuno.

El control de tiempo de los semáforos a tiempo fijo, se puede realizar mediante el programa TRANSYT <sup>[3]</sup>. Este programa se basa en un índice de rendimiento. Este índice puede aumentar o disminuir según las condiciones desarrolladas en el algoritmo. En el caso de que se reduzca el índice, el tiempo en verde de los semáforos disminuye y si dicho índice aumenta, el tiempo en verde se incrementa. Para el cálculo de este índice, se tiene en consideración el tiempo que tardan los vehículos en alcanzar el semáforo y el coste de las detenciones. El programa TRANSYT, además, permite la sincronización entre varias intersecciones para que no se detenga el vehículo por un semáforo en rojo en una vía principal, lo que se conoce como “onda verde”.

El control de semáforos a tiempo variable se puede realizar mediante el programa SCOOP <sup>[4]</sup>. Este sistema ajusta el tiempo en verde de los semáforos a partir del flujo de vehículos. La adaptación del tiempo con respecto al flujo de vehículos se realiza cada poco tiempo. Si se produce un cambio repentino en el flujo de vehículos, la adaptación del tiempo al nuevo flujo sería de forma progresiva pero no instantánea. El sistema SCOOP adquiere el flujo de vehículos mediante sensores, normalmente lazos inductivos, instalados en dirección opuesta a la línea de detención, lo más lejos posible.

El sistema SCAT <sup>[5]</sup>, también se caracteriza por controlar el tiempo de los semáforos de forma variable. En este caso, se adquiere el flujo de vehículos a través de lazos inductivos instalados por debajo del asfalto o mediante cámaras de videovigilancia. Este tipo de software obtiene las duraciones de los ciclos y de cada fase según el grado de saturación. Los datos recopilados por los sensores se tratan en un módulo lógico que contiene los parámetros personalizados para el control del ciclo semafórico. Este módulo también se encarga de controlar los discos de encendido de los semáforos. El sistema SCAT se conecta a red, por lo que se puede



observar desde un centro de control los datos de tráfico recogidos por los sensores. Además, permite desde dicho centro modificar la fase activa.

Otro sistema capaz de controlar el tiempo es el denominado RHODES [6]. Este sistema, igual que los anteriores, calcula el tiempo de los semáforos a partir del flujo de vehículo con la diferencia de que es capaz de detectar vehículos individuales o un conjunto de vehículos en el caso de que haya congestión de tráfico. El flujo se puede adquirir mediante sensores de lazo inductivo instalados por debajo del asfalto o a partir de cámaras de videovigilancia. Las decisiones con respecto al control semafórico se toman en 3 niveles de jerarquía. En primer lugar, se analizan las características de tráfico, a continuación, se asigna el tiempo en verde de los semáforos a partir de dichas características y por último, el sistema decide el momento óptimo para realizar el cambio de fase.

Estudios de investigación, como el realizado por la Universidad Señor de Sipán [7], exponen que disminuiría la congestión de tráfico si el tiempo de los semáforos fuese variable, es decir, que se estableciese el tiempo según la afluencia de vehículos. El desarrollo de esta investigación se basa en la lógica difusa por lo que se puede obtener los tiempos de los semáforos a partir de las entradas, imágenes que se adquieren de las cámaras de videovigilancia. Estas imágenes, se tratan mediante un programa, OpenCV, instalado en el módulo Pingüino. El cálculo del flujo vehicular, se obtiene a partir del número de píxeles que cambian debido a la iluminación de la imagen. El cambio de los discos de los semáforos se produce a través del microcontrolador mencionado. Los resultados de la investigación fueron favorables ya que se disminuía la congestión en un 84%.

Investigadores de la Universidad Politécnica de Tianjin [8] publicaron en el año 2011, un artículo de investigación en el que expusieron la necesidad de cambiar la forma de control de los semáforos debido al aumento del flujo de vehículos sobre todo en las áreas metropolitanas originado por el rápido desarrollo económico. El sistema desarrollado está formado por una cámara de videovigilancia, un módulo para capturar el vídeo, un módulo DSP y otro módulo para controlar las luces de semáforo. La cámara, en tiempo real, capta las imágenes del vídeo y las procesa en el módulo de captura de video. La señal de vídeo se envía al módulo DSP, que procesa todas las señales de video de las distintas cámaras y analiza de afluencia de tráfico a partir de un algoritmo. Este módulo, también modifica el encendido de los discos de los semáforos.

El control de los semáforos de la investigación anterior, se basa en la lógica Fuzzy. Esta lógica se caracteriza por transformar el comportamiento humano en un algoritmo de control, por lo que el sistema simulará el modo de pensar del ser humano. En este tipo de lógica las entradas son inciertas, los procesos no son lineales, se emplean múltiples variables y no tienen un modelo matemático definido.



## 2.2 INGENIERÍA DE TRÁFICO

La ingeniería de tráfico se encarga de regular el tránsito tanto de los vehículos como de los peatones que cruzan la calzada por el paso de peatones. El análisis de dicha ingeniería<sup>[9]</sup> se subdivide en los siguientes apartados: propósito de la regularización semafórica en la sección 2.2.1, elementos constitutivos de los semáforos en la sección 2.2.2, conceptos para la regularización semafórica en la sección 2.2.3 y cálculo del ciclo semafórico y de reparto en la sección 2.2.3.

### 2.2.1 PROPÓSITO DE LA REGULARIZACIÓN SEMAFÓRICA

La elevada intensidad de vehículos sobre todo en las grandes urbes, hizo necesaria la implantación de un sistema capaz de dar preferencia de paso a los vehículos en una intersección. Esto dio lugar a la regularización del tráfico a partir de los semáforos. Los semáforos otorgan un elevado nivel de seguridad pero con el inconveniente de originar congestiones de tráfico.

La mayoría de ellos se instalan en las carreteras ubicadas en el interior de las ciudades pero puede haber semáforos en las carreteras de entrada a las urbes, en circunvalaciones o en intersecciones que tengan un alto grado de peligrosidad.

Las intersecciones que no están reguladas por semáforos, la preferencia la tiene el vehículo que está a la derecha o el carril que disponga la señal de preferencia. Si la intersección está regulada por semáforos, la disposición anterior cambia ya que la preferencia la puede tener otro carril que no esté a la derecha del vehículo.

Los semáforos se encargan de establecer el derecho de paso a los vehículos que están en una intersección. Debido a ello, si el semáforo está en rojo se produce una congestión de vehículos por lo que hay que asignar el espacio necesario para que todos los vehículos esperen a que el semáforo se ponga en verde de forma segura.

El empleo de semáforos facilita los movimientos de los vehículos sobre todo en isletas ya que se permite la derivación de los vehículos a otras vías mediante la secuencia de paso establecida por los semáforos. En las intersecciones que no hay semáforos, los movimientos de los vehículos están más limitados, el número de vehículos puede sobrepasar el espacio asignado para su espera segura y el tiempo en llegar al destino es mayor ya que el conductor disminuye la velocidad por seguridad o puede quedar retenido hasta que no hayan vehículos por su derecha.

El objetivo de la regularización semafórica se basa en mejorar la movilidad de los vehículos con una mayor seguridad, en disminuir el tiempo que tarda el conductor en llegar a su destino por las detenciones que tenga que realizar y en evitar congestiones de tráfico. Estos objetivos consiguen reducir el consumo de combustible y la contaminación ambiental.



### 2.2.2 ELEMENTOS CONSTITUTIVOS DE LOS SEMÁFOROS

Los semáforos se componen principalmente de dos elementos cabeza y columna y opcionalmente, o según las condiciones de visibilidad, de dos elementos secundarios, báculo y semáforo de repetición.

La cabeza del semáforo está constituida por los componentes luminosos. Esta pieza contiene las luces de indicación para la regulación del tráfico vehicular y peatonal, por lo que se orienta cada cabeza según la región a la que afecte. Cada cabeza que apunta a una región determinada, se le denomina cara del semáforo. La cara del semáforo está compuesta por 2 unidades lumínicas en el caso de que se oriente a los peatones o de 3 o más unidades lumínicas si se orienta a los vehículos. Las luces que se integran en la cara del semáforo, se instalan dentro de un reflector cóncavo para dirigir el haz de luz en una determinada dirección. Además, las caras de los semáforos poseen un vidrio circular de color, lente, para así indicar a los vehículos la permisión de paso, verde, la detención obligatoria, rojo y el aviso de detención en un tiempo próximo, amarillo.

En determinados momentos del día los rayos del sol inciden sobre las unidades lumínicas. Este hecho puede causar confusión ya que el conductor o los peatones pueden entender como que tienen o no la prioridad de paso, originando un problema de seguridad. Para solucionar este efecto, se instala una visera sobre cada una de las unidades lumínicas de color negro mate. Además, la colocación de estos componentes, evita confusiones con otros semáforos que constituyen la intersección ya que impide que se vea el haz de luz desde los lugares a los que no está dirigido.

Si el conductor se detiene muy próximo al semáforo puede ser que no vea la cara del semáforo por estar colocado a una altura superior de la que alcanza con su ángulo de visión. Para ello, se pone una cara de semáforo, denominado semáforo de repetición a la altura del conductor en la columna.

Las caras de los semáforos se sujetan en las columnas o en los báculos. Las columnas se instalan en el suelo mediante pernos para así soportar la fuerza ejercida por el viento y en el caso de que se instale, también debe soportar la fuerza que ejerce el báculo. En ciudades interiores, se fabrican con acero galvanizado y en ciudades costeras de hierro fundido ya que deben soportar climas húmedos con ambiente salino. La altura de las columnas puede ser de 2 metros o de 2,4 metros.

Los báculos se emplean en los casos de que se requiera una elevada visibilidad a una gran distancia de las unidades lumínicas que constituyen la cara del semáforo. Además, se colocan si la visibilidad de la cara del semáforo instalado en la columna se ve afectada por el arbolado o por vehículos de elevadas dimensiones detenidos en el semáforo o estacionados. La estructura de los báculos se compone de dos tramos rectos unidos por un tramo curvo. Su altura es de 4,5 metros y el tramo



saliente sobre el tramo vertical varía desde los 3,5 metros hasta los 5,5 metros según el número de carriles y la anchura de ellos. Las caras de los semáforos en los báculos, se instalan lo más centrado posible a los carriles a los que va dirigido.

La regulación de los semáforos que controlan el paso de vehículos en una intersección, se realiza mediante un dispositivo denominado regulador. Este aparato se encarga de modificar los cambios de las luces de los semáforos. El tiempo establecido para cada semáforo, se almacena en dicho dispositivo. En algunas intersecciones se instalan detectores que recogen información referente a la intensidad, a la velocidad, a la densidad y a la ocupación del carril. Estos dispositivos se instalan en el interior del asfalto pero se rompen fácilmente en tareas de mantenimiento por lo que si se averían, se cambian por cámaras de videovigilancia.

### 2.2.3 CONCEPTOS PARA LA REGULACIÓN SEMAFÓRICA

Los tiempos de regularización de los semáforos que se ubican en una intersección, se obtienen a partir de unos parámetros que cuantifican la cantidad de vehículos que pasan por el semáforo y si dichos vehículos siguen con idéntica trayectoria o se desvían hacia otro carril.

El tiempo transcurrido desde que un semáforo se pone en verde hasta la repetición de dicha situación se denomina ciclo semafórico. Este tiempo se reinicia después de efectuarse la secuencia completa de los semáforos conectados a un mismo regulador.

El ciclo semafórico se divide en fases. Mientras la fase esté activa, no se modifica el color de los discos de los semáforos. Las fases en que se distribuye un ciclo semafórico hacen referencia a los movimientos que pueden realizar los vehículos que circulan por distinto carril de forma simultánea sin que exista peligro de colisión. La determinación del número de fases en una intersección no está definida por ninguna normativa, sino que depende de las características del tráfico y del trazado.

Los vehículos que acceden a una intersección deben tener tiempo suficiente como para abandonarla y dejar el paso libre para que entren otros vehículos por otra rama sin que exista peligro de colisión entre ellos. Este tiempo se denomina despeje.

Cada movimiento que puede darse simultáneamente en una intersección tiene un tiempo establecido, denominado reparto de ciclo. En el momento en que se termina el tiempo asignado para la fase, se produce un cambio de fase.

La intensidad es la cantidad de vehículos que pasan por un semáforo en una unidad de tiempo, normalmente se efectúa el conteo durante una hora, por lo que se mide en vehículos/hora.



El vehículo que accede a una intersección puede seguir su trayectoria o desviarse de ella, es decir, incorporarse a otro carril si dicho cambio se indica mediante una señal vertical u horizontal. En el caso de que se desvíe de su trayectoria, el conductor no acelera tanto como si siguiese su trayectoria por lo que el tiempo en abandonar la intersección se incrementa. Por ello, se aplica un factor equivalente a los vehículos que se desvían para así incrementar el tiempo en que el semáforo está en verde. De esta forma se garantiza el paso de la mayoría de los vehículos.

La zona de acceso de los vehículos a la intersección está compuesta por uno o más carriles. Si el acceso tiene 2 o más carriles, el tiempo en verde del semáforo es inferior ya que la capacidad de los vehículos es mayor para la misma intensidad. Debido a ello, se obtiene el flujo equivalente de automóviles a partir de la intensidad de los vehículos, del factor equivalente descrito anteriormente y del número de carriles de acceso.

La relación entre el flujo equivalente de automóviles y la cantidad de vehículos por carril y por sentido, establecido en 1800 <sup>[10]</sup>, se denomina índice de saturación. Cada movimiento que constituye una fase tiene un índice de saturación. En el caso de que el vehículo pueda seguir su trayectoria (movimiento 1), o se desvíe de ella (movimiento 2, 3...), el flujo de automóviles para ese carril depende tanto de los que siguen como de los que se desvían, por lo que en este caso, se sumarían los índices de saturación.

Cada fase puede poseer diferentes índices de saturación, por lo que se escoge para el cálculo del tiempo en verde, el índice que sea mayor sea para que así dé tiempo suficiente a que se produzcan todos los movimientos que constituyen una fase.

#### 2.2.4 CÁLCULO DEL CICLO SEMAFÓRICO Y DE REPARTO

La duración del ciclo semafórico debe estar comprendida entre los límites que fija la psicología del conductor. Este límites, se establece entre 35 y 120 segundos. Debido a ello, si el tiempo del ciclo es inferior o superior a los límites descritos, se deberá ajustar el tiempo de cada fase para así acomodarse a la mentalidad tanto de los conductores como de los peatones.

El tiempo en que los semáforos correspondientes a los vehículos están en rojo puede ser que no se ajuste al tiempo que necesitan los peatones para cruzar la calzada. Si se da este caso, se incrementará el tiempo de detección de los vehículos en el semáforo modificando el tiempo en verde de las demás fases.

La detención de los vehículos en un semáforo se efectúa de forma progresiva, por lo que no se puede pasar directamente de verde a rojo. Debido a ello, se utiliza la luz amarilla. Esta luz se caracteriza por indicar a los vehículos que el semáforo se



va a poner en rojo en un tiempo muy próximo por lo que tienen que reducir la velocidad con anterioridad para detenerse antes de cruzar la línea de detención.

El tiempo en que la luz amarilla permanece encendida no está regulado en ninguna normativa, por lo que se deja a la elección del operario. Se aconseja que dentro de una misma urbe se emplee el mismo tiempo en todos los semáforos para que el conductor se habitúe a dicho tiempo y que este comprendido entre 3 y 4 segundos. Además, se recomienda el empleo de un tiempo superior a los 4 segundos si el semáforo se ubica en carreteras con una velocidad superior a la establecida en ciudad.

El cálculo de ciclo semafórico se obtiene a partir de las fases, movimientos, intensidad, número de carriles, factor equivalente, flujo equivalente de automóviles y flujo de saturación.

El flujo equivalente de automóviles es directamente proporcional a la intensidad y al factor equivalente e inversamente proporcional al número de carriles, tal y como se muestra en la siguiente expresión.

$$q_i = \frac{I_i * Feq_i}{n_i}$$

*Ecuación 1. Flujo equivalente de automóviles*

El parámetro  $q$  se refiere al flujo equivalente de automóviles e  $I$  a la intensidad de vehículos que pasan por un semáforo en un tiempo determinado.  $Feq$  hace referencia al factor equivalente, si se produce o no un giro del vehículo,  $n$  al número de carriles que constituyen el acceso a la intersección e  $i$  al número de movimiento.

La siguiente ecuación, se emplea para el cálculo del flujo de saturación, relación entre el flujo equivalente de automóviles y la cantidad de vehículos por carril y por sentido.

$$y_i = \frac{q_i}{1800}$$

*Ecuación 2. Flujo de saturación*

El flujo crítico,  $y$ , suma de los mayores flujos de saturación que componen cada fase, se emplea para el cálculo del tiempo de ciclo. Para este cálculo, además, se emplea el tiempo de pérdidas. Dicho tiempo es directamente proporcional al número de fases y al tiempo en amarillo de los semáforos.

$$T_p \text{ (Tiempo pérdidas)} = N^{\circ} \text{ Fases} * \text{Tiempo Amarillo}$$

*Ecuación 3. Tiempo pérdidas*



$$T_c (\text{Tiempo de ciclo}) = \frac{1,5 * T_p + 5}{1 - y}$$

*Ecuación 4. Tiempo de ciclo*

El tiempo en verde,  $G_i$ , de cada semáforo que se ubica en la misma intersección, se obtiene mediante el flujo de saturación de la fase,  $y_i$ , el flujo crítico,  $y$ , el tiempo de ciclo,  $T_c$ , y el tiempo de pérdidas,  $T_p$ . En este caso, el parámetro  $i$ , se refiere al número de fase.

$$G_i = \frac{y_i}{y} \cdot (T_c - T_p)$$

*Ecuación 5. Tiempo semáforo verde*

El tiempo en que el semáforo de cada fase está en rojo,  $R_i$ , se obtiene de la resta entre el tiempo ciclo, el tiempo en amarillo, y el tiempo en verde de dicha fase.

$$R_i = T_c - T_a - G_i$$

*Ecuación 6. Tiempo semáforo rojo*

## 2.3 HARDWARE

Para la implantación del sistema, se necesitan componentes hardware capaces de adquirir información sobre el grado de ocupación de los carriles, mediante sensores o cámaras de videovigilancia, de transmitir y recibir información a través de cable, de señal WiFi o mediante la red móvil. En el mercado actual existe una variedad muy amplia de dispositivos que realizan las funciones descritas, por lo que se analizan para así elegir los más óptimos y rentables.

### 2.3.1 SENSORES

El grado de ocupación de los vehículos se puede obtener mediante sensores. Dichos sensores deben ser capaces de detectar un vehículo, por ello se analizan sensores fotoeléctricos y electromagnéticos.

#### 2.3.1.1 FOTOELÉCTRICOS

Los sensores fotoeléctricos se caracterizan por transformar el haz de luz que índice en una fotorresistencia, reflector, en una variación de resistencia eléctrica. Debido a ello, se obtiene una tensión eléctrica a partir de la intensidad de la radiación



incidente. Ambos dispositivos deben estar alineados para su correcto funcionamiento [11].

En un lado del carril se colocaría el led infrarrojo y en el otro lado el reflector. En el instante en que pase un vehículo o una persona entre los dispositivos, se cortaría el haz de luz que emite el led y el sistema incrementaría la ocupación en el caso de los vehículos o detectaría la presencia de una persona esperando para cruzar.

El acceso a las intersecciones se realiza mediante uno o varios carriles que pueden tener diferente anchura. Debido a ello, se realiza un estudio de mercado según la distancia máxima que puede abarcar este tipo de sensores y la posibilidad de utilizar el sensor para diferentes distancias.

- Sensor XUKOARCTL2<sup>[12]</sup>
  - Fabricante: Telemecanique Sensors
  - Función Multimodo
  - Distancia
    - o Funcionamiento en difuso con supresión de fondo: 0,28 metros
    - o Funcionamiento en difuso: 0,8 metros
    - o Funcionamiento con reflector XUZC50<sup>[13]</sup>: 4 metros
    - o Funcionamiento con haz fotoeléctrico XUKOARCTL2T<sup>[14]</sup>: 30 metros
  - Precio unidad: 47,90€ IVA incluido
  - Precio reflector unidad: 12,67€ IVA incluido
  - Precio haz fotoeléctrico unidad: 105,14€ IVA incluido



Fotografía 1. Sensor receptor UKOARCTL2



Fotografía 3. Sensor emisor XUKOARCTL2T



Fotografía 2. Reflector XUZC50



- Sensor XUB0APSNM12<sup>[15]</sup>
  - Fabricante: Telemecanique Sensors
  - Función Multimodo
  - Distancia
    - Funcionamiento en difuso con supresión de fondo: 0,12 metros
    - Funcionamiento en difuso: 0,3 metros
    - Funcionamiento con reflector XUZC50: 3 metros
    - Funcionamiento con haz fotoeléctrico XUB0AKSNM12T<sup>[16]</sup>: 20 metros
  - Precio unidad: 72,48€ IVA incluido
  - Precio reflector unidad: 12,67€ IVA incluido
  - Precio haz fotoeléctrico unidad: 44,70€ IVA incluido



Fotografía 4. Sensor receptor XUB0APSNM12



Fotografía 5. Sensor emisor XUB0BKSNM12T

- Sensor XUK1ARCNL2<sup>[17]</sup>
  - Fabricante: Telemecanique Sensors
  - Función modo único
  - Distancia
    - Funcionamiento con reflector XUZC50: 7 metros
  - Precio unidad: 75,92€ IVA incluido
  - Precio reflector unidad: 12,67€ IVA incluido



Fotografía 6. Sensor emisor-receptor XUK1ARCNL2

### 2.3.1.2 ELECTROMAGNÉTICOS

Los sensores electromagnéticos se caracterizan por detectar modificaciones en un campo magnético o eléctrico. Estos campos se forman al circular una corriente eléctrica por uno o varios hilos conductores. En el momento en que pasa un vehículo



por ellos se produce una variación de su impedancia. Esta variación permitiría la detección del vehículo ya que este posee componentes de metal.<sup>[11]</sup>

La detección del vehículo, tal y como se ha descrito, se efectúa mediante un cable conductor que genera un campo magnético. Este cable instala en el asfalto de la carretera. Para ello se realiza una estrecha franja con una radial y se introduce el cable. Este cable se conecta a un dispositivo que detecta la variación de impedancia si pasa un vehículo.

- Sensor LDP1SA1BM24<sup>[18]</sup>
  - Fabricante: Carlo Gavazzi
  - Salida de relé o de pulsos
  - Precio unidad: 157,05€ IVA incluido



Fotografía 7. Sensor lazo inductivo LDP1SA1BM24

- Sensor MATRIX<sup>[19]</sup>
  - Fabricante: BEA
  - Salida de relé
  - Precio unidad: 157€ IVA incluido



Fotografía 8. Sensor lazo inductivo Matrix

### 2.3.2 CÁMARAS DE VIDEOVIGILANCIA

Las cámaras de videovigilancia se pueden emplear para adquirir las características de tráfico de los vehículos que acceden a una intersección o para conocer la presencia o no de peatones esperando en el paso de cebr. Para ello, se tratarían las imágenes procedentes de las cámaras de videovigilancia mediante un programa implementado en el microprocesador.



- Cámara IP Foscam FI9804W<sup>[20]</sup>
  - Cámara exterior
  - 1 Megapíxel
  - Resolución: 1280x720p
  - Ángulo de visión: 70°
  - Distancia nocturna: 20 metros
  - Conectividad: WiFi 2,4 GHz
  - Precio: 122,90€ IVA incluido



Fotografía 9. Cámara IP Foscam FI9804W

- Cámara IP Foscam FI9805E<sup>[21]</sup>
  - Cámara exterior
  - 1,3 Megapíxeles
  - Resolución: 1280x960p
  - Ángulo de visión: 70°
  - Distancia nocturna: 30 metros
  - Conectividad: Ethernet
  - Precio: 149,99€ IVA incluido



Fotografía 10. Cámara IP Foscam FI9805E

- Cámara IP Dahua X-Security<sup>[22]</sup>
  - Cámara exterior
  - 3 Megapíxeles
  - Resolución: 1920x1080p
  - Ángulo de visión: 100°
  - Distancia nocturna: 30 metros
  - Conectividad: WiFi y Ethernet
  - Precio: 71,87€ IVA incluido



Fotografía 11. Cámara IP Dahua X-Security

- Cámara IP Nivian ONV524<sup>[23]</sup>
  - Cámara exterior
  - 1 Megapíxel
  - Resolución: 1280x720p
  - Ángulo de visión: 70°
  - Distancia nocturna: 20 metros
  - Conectividad: WiFi y Ethernet
  - Precio: 78,65€ IVA incluido



Fotografía 12. Cámara IP Nivian ONV524



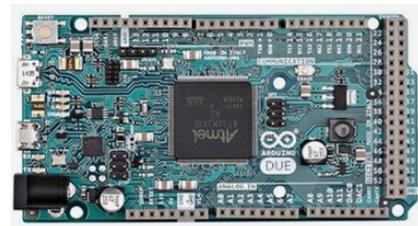
### 2.3.3 MICROPROCESADORES Y MICROCONTROLADORES

Las características adquiridas mediante sensores o cámaras de videovigilancia deben ser procesadas para así establecer el tiempo óptimo de cada fase, el instante en que se activa la siguiente fase y el orden de secuencia de los semáforos de forma óptima.

Los microprocesadores que se pueden emplear en este proyecto, se caracterizan por poseer una memoria interna que disponga de capacidad suficiente para los programas realizados. Además, deben tener entradas y salidas digitales o analógicas tanto para recibir los datos de los sensores, en el caso de que se empleen, como para controlar el encendido de los diferentes discos de los semáforos que constituyen las intersecciones. Si se emplean cámaras o dispositivos capaces de controlar los semáforos vía IP, los microprocesadores deberán poseer un módulo que permita la conexión a Ethernet o WiFi.

El tratamiento de imágenes procedentes de un vídeo requiere un coste computacional elevado, por lo que habrá que analizar la velocidad del procesador y la memoria RAM disponible.

- Arduino DUE<sup>[24]</sup>
  - Cantidad I/O: 54
  - Memoria Flash: 512 KB
  - Memoria RAM: 96 KB
  - Núcleos: 1
  - Velocidad procesador: 84 MHz
  - Conectividad: micro USB
  - Precio: 42,35€ IVA incluido



Fotografía 13. Arduino DUE

- Arduino GENUINO YÚN SHIELD<sup>[25]</sup>
  - Cantidad I/O: 32
  - Memoria Flash: 16 MB
  - Memoria RAM: 64 MB
  - Núcleos: 1
  - Velocidad procesador: 400 MHz
  - Conectividad: USB, Ethernet y WiFi
  - Precio: 53,12€ IVA incluido



Fotografía 14. Arduino GENUINO YÚN SHIELD



- Raspberry Pi 3 Model B+[<sup>26</sup>]
  - Cantidad I/O: 40
  - Memoria Flash: Micro SD
  - Memoria RAM: 1 GB
  - Núcleos: 4
  - Velocidad procesador: 1,4 GHz
  - Conectividad: USB, Ethernet y WiFi
  - Precio: 35,66€ IVA incluido



Fotografía 15. Raspberry Pi 3 Model B+

- Raspberry Pi Zero W[<sup>27</sup>]
  - Cantidad I/O: 40
  - Memoria Flash: Micro SD
  - Memoria RAM: 512 MB
  - Núcleos: 1
  - Velocidad procesador: 1 GHz
  - Conectividad: micro USB, Bluetooth y WiFi
  - Precio: 10,44€ IVA incluido



Fotografía 16. Raspberry Pi Zero W

- Raspberry Pi Zero[<sup>28</sup>]
  - Cantidad I/O: 40
  - Memoria Flash: Micro SD
  - Memoria RAM: 512 MB
  - Núcleos: 1
  - Velocidad procesador: 1 GHz
  - Conectividad: micro USB
  - Precio: 5,22€ IVA incluido



Fotografía 17. Raspberry Pi Zero

#### 2.3.4 SISTEMAS DE COMUNICACIÓN

La información captada por los sensores o por la cámara de vigilancia se envía al microprocesador para su análisis. Este dispositivo, a partir de los datos obtenidos, dirige las secuencias de los semáforos, por lo que envía una señal a los dispositivos de control instalados en cada uno de ellos. Estas señales se pueden enviar mediante cables eléctricos, cables de red o a partir de módulos WiFi.



En este proyecto se contempla la posibilidad de conectar cada intersección a un centro de control para conocer y modificar la fase activa en el caso de que el operario lo considere oportuno. Debido a ello, se analizan los dispositivos 4G que permitan la comunicación entre el dispositivo instalado en la intersección y el centro de control.

#### 2.3.4.1 CABLES ELÉCTRICOS

Las características del tráfico adquiridas por los sensores y la secuencia de los discos de los semáforos, se pueden transmitir mediante cables eléctricos. Estos cables deberán tener varios hilos y ser resistentes tanto a la humedad como a temperaturas extremas.

Para la instalación de los cables en el asfalto, se realizaría una hendidura con una radial. Dicha hendidura tendría una profundidad aproximada de 3 centímetros. El cable se introduciría en ella y se taparía con cemento.

- Cable RS PRO<sup>[29]</sup>
  - Sección: 0,5 mm<sup>2</sup>
  - Longitud: 25 metros
  - Mínima temperatura: -60°
  - Máxima temperatura: 180°
  - Resistencia a la temperatura: Sí
  - Precio: 15,35€ IVA incluido



Fotografía 18. Cable RS PRO

- Cable TE Connectivity<sup>[30]</sup>
  - Sección: 0,5 mm<sup>2</sup>
  - Longitud: 100 metros
  - Mínima temperatura: -55°
  - Máxima temperatura: 125°
  - Resistencia a la temperatura: Sí
  - Precio: 87,51€ IVA incluido



Fotografía 19. Cable TE Connectivity

#### 2.3.4.2 CABLES ETHERNET

El grado de ocupación de los carriles, se puede obtener mediante cámaras de videovigilancia. Las imágenes de video que captan las cámaras se pueden enviar al microprocesador mediante cable Ethernet.



Las cámaras se situarían sobre las columnas, en el caso de que se analice la presencia o no de peatones, y sobre los báculos si se analiza las características del tráfico de vehículos. Debido a ello, el cable, se introduciría por el interior del semáforo y se canalizaría por la tubería que posee los cables de control de los discos del semáforo hasta la cabina de regularización semafórica.

- Cable Ethernet OEM<sup>[31]</sup>
  - Longitud: 100 metros
  - Velocidad: Hasta 40 Gbit/s
  - Distancia máxima: 100 metros
  - Precio: 40€ IVA incluido



Fotografía 20. Cable Ethernet OEM

- Cable Ethernet Digitus<sup>[32]</sup>
  - Longitud: 100 metros
  - Velocidad: Hasta 40 Gbit/s
  - Distancia máxima: 100 metros
  - Precio: 51€ IVA incluido



Fotografía 21. Cable Ethernet Digitus

### 2.3.4.3 MÓDULOS WiFi

El envío de las características del tráfico y de la gestión de los discos de los semáforos, se puede realizar con módulos WiFi. Debido a ello, no sería necesario la instalación de cables eléctricos ni cables Ethernet.

La característica principal de los módulos WiFi es la compatibilidad TCP/IP por lo que cualquier microcontrolador conectado a él, tiene acceso a la red. Estos dispositivos pueden funcionar por si solos, sin acoplarse a un microcontrolador, si poseen una memoria flash en la que se introduzcan las acciones que debe realizar. Si por el contrario, no posee una memoria flash, el módulo WiFi deberá ir incorporado a un microcontrolador que realizará las acciones establecidas por el programador y enviará los datos a partir del módulo citado.

Estos módulos disponen de entradas y salidas que permitirían tanto la adquisición de datos de un sensor como el accionamiento de las luces que se ubican en las caras de los semáforos.



- ESP8266EX<sup>[33]</sup>
  - Tensión alimentación: 3,3V
  - Velocidad procesador: 160MHz
  - Memoria RAM: 96 KB
  - Memoria flash externa: 4 MB
  - Pines I/O: 16
  - Precio: 1,69€ IVA incluido



Fotografía 22. ESP8266EX

- ESP32<sup>[34]</sup>
  - Tensión alimentación: 3,3V
  - Velocidad procesador: 240MHz
  - Memoria RAM: 520 KB
  - Memoria flash externa: 16 MB
  - Pines I/O: 32
  - Precio: 3,96€ IVA incluido



Fotografía 23. ESP32

#### 2.3.4.4 MÓDULOS GSM

El microprocesador que controla la secuencia de los semáforos, se conectará a internet por lo que el operario desde una sala de control, podrá ver la fase activa y cambiarla en el caso de que sea necesario.

La conexión a internet se realiza mediante la red móvil, por lo que será necesario conectar un módulo al microprocesador capaz de permitir el envío y la recepción de información. El tipo de conexión podrá ser GPRS, si solo se envían datos, o 4G, en el caso de que, además de los datos, se envíen imágenes.

- Módulo SIM7600X 4G HAT<sup>[35]</sup>
  - Red: 4G
  - Tecnología: SIM7600X
  - Valido para Raspberry Pi
  - Precio: 85,99€ IVA incluido



Fotografía 24. Módulo 4G para Raspberry Pi



- Módulo GSM 2<sup>[36]</sup>
  - Red: GPRS
  - Tecnología: SIM900
  - Valido para Raspberry Pi
  - Precio: 17,89€ IVA incluido



Fotografía 25. Módulo GPRS para Raspberry Pi

- Módulo GPRS<sup>[36]</sup>
  - Red: GPRS
  - Tecnología: SIM900
  - Valido para Arduino
  - Precio: 40,33 € IVA incluido



Fotografía 26. Módulo GPRS para Arduino

## 2.4 SOFTWARE

El control de las secuencias de los semáforos en una intersección y la interpretación de las características del tráfico adquiridas por los sensores o por las cámaras de videovigilancia requiere de la implantación de uno o varios programas.

En el caso de emplear el microcontrolador Arduino para el control de la intersección, se desarrollará el programa en el entorno de programación que posee el propio Arduino. Este entorno se basa en una adaptación del lenguaje de programación C++. El microprocesador Raspberry Pi se caracteriza por poseer un funcionamiento equivalente al de un ordenador, por lo que se pueden instalar diversos sistemas operativos. Debido a ello, el programa se puede efectuar en distintos lenguajes de programación tal como Java o C++.

### 2.4.1 VISIÓN ARTIFICIAL

El objetivo de la visión artificial se basa en la extracción de características físicas del medio que nos rodea mediante un ordenador. Esta técnica, según se describe en el libro “Visión Por Computador”<sup>[37]</sup>, se encuentra en una etapa primitiva por lo que el desarrollo de esta tecnología con el tiempo, será de vital importancia en el ámbito cotidiano.

La visión artificial se caracteriza por ser capaz de tratar imágenes que recibe del medio físico a partir del color, brillo, formas, etcétera. Las imágenes pueden ser estáticas o dinámicas. Las imágenes estáticas no cambian con el tiempo, por lo que



son fotografías mientras que las imágenes dinámicas cambian con el transcurso del tiempo. Este último tipo de imágenes suele provenir de un vídeo.

Una imagen en color se transforma en visión artificial en una matriz tridimensional. En el caso de que la imagen se caracterice por ser RGB la primera matriz contendrá la intensidad de rojo puro, la segunda verde puro y la última azul puro. El color rojo, el verde y el azul son colores primarios. La mezcla de los tres colores da lugar a las diferentes tonalidades de píxeles que se observan en una imagen.

La cantidad de píxeles de una imagen depende de su resolución, por lo que si la resolución de esta es de 1280x720, la imagen contendrá 921.600 píxeles. En el caso de que la resolución fuese de 1920x1080 la imagen contendría 2.073.600 píxeles. El valor de cada píxel de cada plano se ubica en una posición de la matriz con unas coordenadas x e y.

Si la imagen está en escala de grises la matriz es unidimensional. En este caso, los píxeles indican el brillo.

Los píxeles, tanto como si la imagen está en color como en escala de grises, se encuentran en un rango comprendido desde 0 hasta 255.

Las imágenes se pueden agrupar formando secuencias animadas, vídeos. Normalmente se escogen 25 imágenes por segundo para formar la secuencia ya que es la frecuencia máxima que puede captar el ojo humano.

El tratamiento de imágenes en visión artificial se basa en las acciones que realiza el ser humano para detectar un objeto. Debido a ello, para la obtención de las características, se captura la imagen mediante una cámara, se eliminan las partes que no se desean o se resaltan las necesarias, se separan las partes de interés y por último se realiza la distinción del objeto deseado.

La secuenciación descrita anteriormente puede ir hacia atrás si falla alguna de las siguientes, como por ejemplo, si no se efectúa la distinción del objeto, el sistema retorna a la fase anterior y vuelve a separar las partes de interés.

#### 2.4.1.1 OBTENCIÓN IMAGEN Y TRANSFORMACIÓN DIGITAL

Las cámaras y los ordenadores son los componentes necesarios para un sistema de visión artificial. Las cámaras adquieren una imagen o un vídeo del mundo físico y el ordenador se encarga de realizar el tratamiento.



Las imágenes y vídeos, tal y como se ha descrito anteriormente, se adquieren mediante una cámara. Este dispositivo posee un orificio que permite el paso de la luz por lo que se proyecta la imagen en la cara opuesta. Debido a ello, la imagen obtenida por la cámara está invertida. En el orificio se coloca una lente biconvexa para adquirir la luz y proyectarla sobre el plano de formación de la imagen. En las cámaras digitales, la imagen se proyecta en un dispositivo de carga acoplada formado por una matriz de elementos fotosensibles. Debido a ello, se obtiene una carga eléctrica proporcional a la intensidad de la luz que permite la distinción de los colores que constituyen la imagen.

La lente se caracteriza por poseer un grosor despreciable y dos ejes de simetría, el eje axial que es el mayor y el eje óptico que es el menor. El punto en el que se corta el eje axial y el eje óptico se denomina centro óptico. La luz que pasa por dicho punto no sufre desviación al atravesar la lente, mientras que los demás haces de luces se desvían al atravesarla. La desviación disminuye a medida que los haces de luces se acercan al centro óptico por ambos extremos. Todos estos haces, se juntan en un punto llamado foco ubicado en el eje óptico. La distancia entre el foco y el centro óptico se denomina distancia focal.

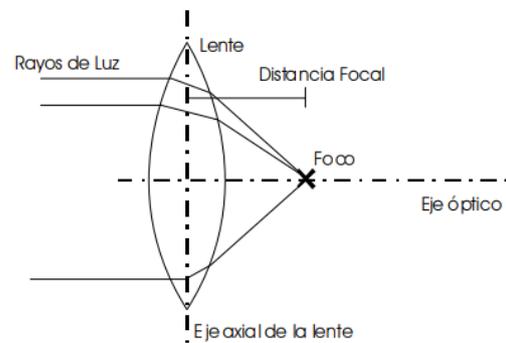


Figura 1. Trayectoria que recorre el haz de luz al atravesar la lente biconvexa

Las lentes están formadas por vidrio óptico. Este material se compone de óxido de bario, lantano y tántalo. La geometría de las lentes se diseñan de tal forma que se modifique la dirección de los haces de luces en el momento en que atraviesan la lente. Este cambio de dirección se obtiene aplicando la propiedad de la luz que consiste en el cambio de su dirección si pasan de un medio a otro. El ángulo de desviación depende del medio que atravesase y el ángulo con respecto a la normal de incidencia del haz de luz en la lente. El grado de dependencia entre los factores anteriores se denomina índice de refracción.

El proceso de transformación de las imágenes del mundo físico al mundo digital se denomina digitalización. En este paso, las imágenes se transforman en una matriz tridimensional, en el caso de que la imagen esté en color o en una matriz unidimensional si la imagen está en escala de grises. La matriz se compone de  $N \times M$  píxeles.

Si la imagen está en color se evalúa el valor del píxel a partir de la intensidad de los colores primarios en un rango comprendido entre 0 y 255. Si la imagen está en escala de grises, se evalúa el brillo de la imagen en el mismo intervalo que el



expuesto para la imagen en color. En el caso de que la imagen esté en blanco y negro, imagen binaria, el valor de los píxeles se establece en 0 o 1.

Las imágenes o vídeos se almacenan en el ordenador en forma de ficheros. En determinadas ocasiones, los archivos ocupan demasiado espacio y es necesario comprimirlos. Este proceso se realiza disminuyendo la cantidad de datos necesarios para reproducir la imagen en el ordenador ya que, para obtener las características del mundo físico, no es necesario que la imagen tenga una elevada definición ni resolución.

En las imágenes en blanco y negro y en escala de grises se pueden establecer relaciones tanto de proximidad como de distancia entre los píxeles que forman los objetos.

Las relaciones de proximidad se basan en el número de píxeles que existen alrededor del píxel de estudio y si están o no conectados entre ellos. Debido a ello, un píxel tiene 8 vecinos a su alrededor si los píxeles se ubican en las coordenadas  $(x-1, y)$ ,  $(x, y-1)$ ,  $(x+1, y)$ ,  $(x, y+1)$ ,  $(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$  y  $(x+1, y+1)$ . En la siguiente imagen, se muestra como lo píxeles marcados con una A y con una B son los vecinos del píxel marcado con una p.

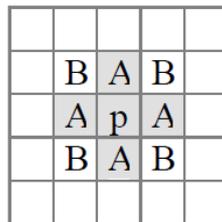


Figura 2. Los 8 vecinos del píxel p

Los píxeles están conectados entre sí si son vecinos y si se puede establecer una relación entre ellos. Dicha relación se basa en el valor del brillo del píxel, en el caso de imágenes en escala de grises o en el valor 0, negro, o blanco, 255, si la imagen es binaria. Como por ejemplo, un grupo de píxeles conectados son los que forman una recta ya que esta se compone de un grupo de píxeles que son vecinos entre sí unidos a otro grupo de píxeles.

Las relaciones de distancia entre píxeles se fundamentan en la distancia euclídea y en la distancia Manhattan que existe entre ellos. La distancia euclídea se caracteriza por ser el camino más corto en llegar de un punto a otro mientras que la distancia Manhattan es el camino que se debe seguir hasta llegar a dicho punto.

$$\text{Distancia euclídea} \rightarrow d(p, q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Ecuación 7. Distancia euclídea



$$\text{Distancia Manhattan} \rightarrow d(p, q) = |x_2 - x_1| + |y_2 - y_1|$$

Ecuación 8. Distancia Manhattan

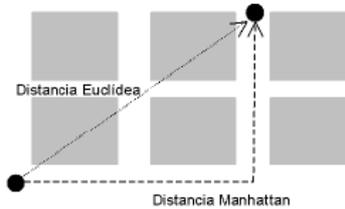


Figura 3. Distancia euclídea y distancia Manhattan

#### 2.4.1.2 PROCESAMIENTO IMÁGENES: FILTRO Y TRANSFORMACIONES GEOMÉTRICAS

Una vez que se ha captado y se ha digitalizado la imagen, se procede al procesamiento de la imagen realizando sobre ella operaciones y transformaciones geométricas para así eliminar partes de la imagen que sean innecesarias o realzar las que sean imprescindibles.

Para conseguir las partes necesarias de la imagen, se realizan las siguientes acciones:

- Operaciones lógicas (solo para imágenes binarias): AND, OR, NOT, XOR
- Operaciones aritméticas: suma, resta, multiplicación y división.
- Operaciones geométricas: escalado, traslación y rotación.
- Operaciones en el dominio de la frecuencia: filtro paso bajo, filtro paso alto y filtro paso banda.
- Operaciones morfológicas (solo en imágenes binarias): dilatación, erosión, apertura y cierre.

Las operaciones lógicas son las más usadas y se emplean en imágenes binarias. La conjunción se caracteriza por aplicar la operación lógica AND entre los píxeles de dos imágenes. La disyunción realiza la operación lógica OR entre los bits de dos imágenes. La negación efectúa la inversión de los píxeles que constituyen la imagen.

Las operaciones aritméticas realizan la suma, la resta, la multiplicación y la división de los píxeles únicamente en imágenes en escala de grises.

Tanto los objetos que forman la imagen como la imagen completa, se pueden trasladar, escalar o rotar, operaciones geométricas.

La traslación consiste en el movimiento de los píxeles de una imagen a partir de un vector de movimiento, es decir, se traslada un punto siguiendo el vector de movimiento hasta alcanzar el punto. En la siguiente ecuación se muestra el punto de destino, la matriz de desplazamiento con el vector de movimiento y el punto inicial.



$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

*Ecuación 9. Matriz de traslación*

El escalado se caracteriza por modificar el tamaño de la imagen. Este hecho es el resultado de la multiplicación de la matriz que contiene el factor de escalado por el punto original, tal y como se aprecia en la siguiente ecuación.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

*Ecuación 10. Matriz de escalado*

La rotación se basa en el giro en torno al origen de coordenadas de los píxeles que forman la imagen. La obtención del punto rotado se adquiere mediante la multiplicación del punto original por la matriz que contiene el ángulo de giro.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

*Ecuación 11. Matriz de rotación*

En el caso de que se requiera aplicar más de una operación geométrica no es necesario aplicar una a una, sino que se pueden aplicar de forma conjunta multiplicando las matrices de operación que se deseen efectuar. Este hecho, disminuye el coste computacional.

El grado de cuantificación en que aparecen los píxeles de una imagen se representa en un histograma. El histograma se caracteriza por poseer dos ejes, el eje de abscisas y el eje de ordenadas. En el eje de abscisas se establecen los distintos valores que pueden adquirir los píxeles y en el eje de ordenadas la cantidad de veces que aparecen. El histograma de una imagen no da información sobre la disposición espacial de los píxeles en ella ya que otra imagen puede tener un histograma parecido.

Si la imagen está en escala de grises, el histograma otorga información acerca de la cantidad de píxeles que hay para cada nivel de intensidad. En el caso de la imagen esté en color, se realiza un histograma para cada color primario, rojo, verde y azul, RGB, en el que indica cada uno de ellos, el brillo de los colores mencionados.

El intervalo de valores en que se representa una imagen en un histograma se denomina rango dinámico. Si el intervalo es pequeño, la imagen tiene poco contraste mientras que si en las zonas intermedias hay una cantidad pequeña de píxeles y en



los extremos hay una cantidad elevada de píxeles, el contraste es muy elevado y se dice que la imagen está saturada. Debido a ello, a partir del estudio del histograma, se puede conocer el grado de calidad de la imagen, por lo que si una imagen posee una de las características explicadas, se debería adquirir de nuevo la imagen ya que sino no se podría realizar la detección del objeto deseado.

En algunos casos, se requiere aumentar o disminuir el contraste para obtener mejor el objeto. Si se desea incrementar el contraste, se aplica una función cuadrática mientras que si se desea reducir, se emplea una función raíz cuadrada.

La calidad de la imagen se puede mejorar si se realiza un ecualizado del histograma. Este proceso consiste en obtener un nuevo histograma mediante una distribución uniforme de los píxeles, por lo que se aumenta la cantidad de información que contiene ya que se suavizan las zonas saturadas permitiendo distinguir el objeto de una forma más adecuada. Para ello, se establece una variable aleatoria entre 0 y 1 que representa la densidad de probabilidad. La transformación se basa en fijar la intensidad de los píxeles entre 0 y 1 y en dividir cada franja del histograma por la cantidad de píxeles que posee para que la suma de todas ellas sea igual a la unidad.

Las operaciones basadas en el dominio de la frecuencia se fundamentan en la repetición de los patrones ya que se pueden desprestigiar los elementos que no interesan como por ejemplo, el ruido que posee una imagen. Existen tres tipos de filtros, filtro paso bajo, filtro paso alto y filtro paso banda. El filtro paso bajo se caracteriza por eliminar los píxeles que poseen una frecuencia elevada, el filtro paso alto suprime los píxeles que tienen una frecuencia baja y el filtro paso banda establece un rango de valores en que si el píxel se ubica dentro de dicho rango se mantiene en la imagen y si no se elimina de ella.

El filtro paso bajo se basa en el promedio del valor de los píxeles vecinos. Este filtro se diseña mediante una matriz denominada kernel en la que el valor del centro corresponde al píxel de estudio y los demás valores a los píxeles vecinos. El filtro de la mediana es un filtro paso bajo. Este filtro modifica el valor del píxel de estudio por el valor de la mediana de los píxeles vecinos.

El filtro paso alto se obtiene mediante la derivada direccional. Esta derivada muestra los cambios que se producen en la dirección por lo que se obtienen los bordes de las siluetas de la imagen con la aplicación de este tipo de filtro.

Los bordes de las figuras de la imagen se pueden obtener, además de aplicando el filtro paso alto con el filtro de la laplaciana. Este filtro aplica la segunda derivada de la función obteniendo un valor nulo cuando la primera derivada se hace máxima. Si se produce la acción anterior, hay un cambio de intensidad o brillo brusco en la imagen. Este hecho conlleva a la detección de los bordes.



El último tipo de operación que se puede aplicar a una imagen son las operaciones morfológicas. Estas operaciones se basan en la dilatación, erosión, apertura y cierre. Este tipo de operaciones solo se puede realizar sobre imágenes binarias.

La dilatación consiste en el engrandecimiento de uno o varios píxeles. Para ello, se crea una matriz, kernel, que indica la estructura del elemento dilatador. Esta estructura recorre toda la imagen y en el instante en que se sitúa sobre un pixel que posee idéntica tonalidad que el kernel, se implanta en dicho punto el elemento dilatador.

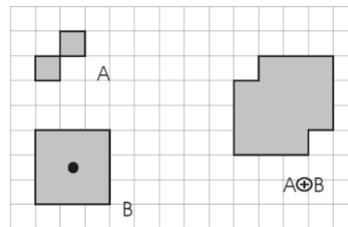


Figura 4. Elemento estructural A, elemento a dilatar B y resultado final

El funcionamiento de la erosión es idéntico al de la dilatación con la diferencia de que si se posiciona sobre un píxel que es de idéntico valor que el kernel, en vez de implementar el kernel en dicho punto, elimina los puntos que indica el kernel, de esta forma la figura se adelgaza.

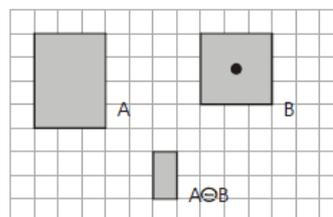


Figura 5. Elemento estructural A, elemento a erosionar B y resultado final

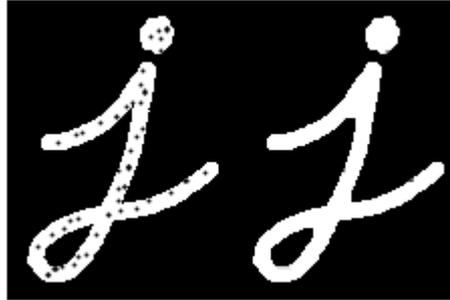
La apertura es la erosión seguida de una dilatación. Este proceso se realiza para eliminar el ruido de una imagen.



Figura 6. Imagen original. Imagen con apertura



El cierre es el proceso contrario el de apertura, es decir, se produce una dilatación seguida de una erosión. En este caso, se eliminan píxeles negros que se ubican en el interior de la imagen.



*Figura 7. Imagen original. Imagen con cierre*

### 2.4.1.3 SEGMENTACIÓN

La segmentación consiste en adquirir la región de interés de la imagen. Para ello, se establece una etiqueta a cada píxel que indica la figura a la que pertenece. En el caso de que todos los objetos tengan etiquetas, la segmentación estará completa sino, la segmentación será parcial. Esto depende de la complejidad de la imagen.

La segmentación no es un proceso que sea igual para todos los casos, ya que no se realiza de la misma forma si se desean adquirir las letras de un texto a si se desean obtener las líneas de delimitación de la carretera.

El estudio de la segmentación de una imagen se fundamenta en el análisis de su textura y de sus bordes.

Al grupo de siluetas que constituyen una imagen y poseen una cierta regularidad, se le denomina textura. La textura de las imágenes se puede analizar desde dos puntos de vista diferentes. El primer punto de vista, denominado top-down, se caracteriza por adquirir patrones de figuras regulares, como puede ser el caso de la detección de las líneas de delimitación de un carril de la carretera o la detección de puertas o mesas. El segundo punto de vista se le denomina bottom-up. Este enfoque se basa en aplicar métodos estadísticos para adquirir las propiedades de la textura de una imagen ya que el objeto a detectar lo compone un grupo de píxeles pequeño como puede ser un pájaro o una piedra.

La segmentación de una imagen se puede realizar mediante la detección de los bordes de las siluetas que la componen. La detección de los bordes se fundamenta en el cambio brusco del valor del píxel actual con el siguiente, es decir, se detecta un borde cuando se produce un cambio brusco de brillo.



Los objetos que se ubican en la imagen se caracterizan por tener los píxeles definidos en un rango de valores determinado por lo que se puede convertir la imagen de escala de grises a binaria estableciendo un píxel umbral para que se adquiera la silueta del objeto u objetos de interés.

El nivel de umbral se puede establecer de forma unitaria o múltiple. El modo unitario se emplea si el objeto u objetos a segmentar poseen un brillo muy diferente al resto de la imagen. Para ello, se debe conocer el brillo de los objetos y del fondo de la imagen. En el caso de que se escogiese un valor umbral erróneo, la segmentación de la imagen sería fallida y no se obtendrían los resultados requeridos.

El modo múltiple de selección de umbral se puede realizar de tres formas diferentes, umbralización de banda, multiumbralización, semiumbralización y umbralización adaptativa.

La multiumbralización se basa en la selección de diferentes niveles de umbral para segmentar distintos objetos que poseen valores de píxeles diferenciados entre ellos.

La semiumbralización se fundamenta en obtener los objetos que se requieren segmentar en escala de grises y el fondo de la imagen en negro.

La umbralización adaptativa analiza la imagen para adquirir un nivel de umbral adecuado. Este es el único método en el que no se establece un nivel fijo, sino que cambia según el brillo de los objetos y del fondo. De esta forma, objetos que tengan valores de píxeles muy diferenciados se pueden obtener mediante este método.

El etiquetado de los diferentes objetos que constituyen la imagen se basa en la asignación de un número a un grupo de píxeles unidos entre sí. Para ello, se recorre la imagen de izquierda a derecha y de arriba abajo y cuando se encuentra un píxel en negro se le asigna un número a dicho píxel y a todos los píxeles negros que estén a su alrededor. Este procedimiento se realiza de forma recursiva para los diferentes objetos que contiene la imagen.

La detección de rectas en una imagen es un método de segmentación ya que nos permite adquirir objetos con una forma cuadrada, rectangular, trapezoidal, etcétera. La obtención de las rectas se basa en el número de puntos que pasan por ellas, es decir, cada píxel posee una ecuación, por lo que si hay otros píxeles seguidos que poseen la misma ecuación, el programa entiende que hay una recta. El empleo de esta técnica implica establecer el número de píxeles seguidos necesarios que poseen la misma ecuación para considerarlos como una recta.

Otra técnica de segmentación es la basada en el color. La implementación de esta técnica implica la obtención de tonalidades comprendidas en un intervalo. Si la



imagen está en color, la intensidad de píxel que se encuentre en el rango establecido para cada plano se mantendrá en la imagen, en caso contrario, se establecerá el píxel en negro. Si la imagen está en escala de grises, se conservará el valor del píxel si su valor se ubica dentro del intervalo y si no, se pondrá dicho píxel en negro.

La segmentación en una secuencia de imágenes, vídeo, se emplea si se desea adquirir objetos en movimientos sobre un fondo estático. Estos objetos se obtienen a partir de la resta de dos imágenes consecutivas ya que el valor de los píxeles de los objetos que se mueven será diferente y el valor de los píxeles estáticos será idéntico, por lo que al realizar la resta dará un valor nulo.

#### 2.4.1.4 CLASIFICACIÓN

El proceso de clasificación se realiza después de la segmentación. En este proceso se distinguen los objetos que constituyen una imagen según sus características.

Para la clasificación se transforma el objeto en un vector cuyos componentes son las características o rasgos. Dicho vector se compara con un vector patrón que posee las características que debe cumplir el objeto.

Los valores que se introducen en el vector se suelen referir a su momento, a alguna transformación, Fourier o cosenos, a las propias imágenes o a características que se obtengan de los objetos mediante métodos algorítmicos.

Las características obtenidas del objeto pueden o no ser iguales a las características patrón, por lo que hay que determinar el grado de semejanza y asignar el objeto al patrón cuyo grado de semejanza sea mayor.

La obtención del vector patrón se realiza mediante un conjunto de patrones válidos. A este proceso se le llama conjunto de aprendizaje. Este conjunto reúne las características de diferentes objetos para obtener una función que sea capaz de saber si el objeto se asemeja o es igual al patrón. Si el conjunto de patrones tiene una elevada capacidad de ellos, se crea otro subconjunto denominado conjunto de test. Este subconjunto se emplea para realizar pruebas con el código y en el caso de que tanto el conjunto como el subconjunto de patrones detecten el objeto, se puede decir que el clasificador es generalizado. Si las pruebas realizadas con el test dan un resultado negativo, se debe rehacer el conjunto de patrones.

Si el conjunto de test tiene demasiados patrones, se efectúa otro subconjunto para así, asegurar más que las características del objeto se aproximan a la del patrón. A este nuevo subconjunto se le denomina conjunto de validación y se utiliza para no tener que emplear otra vez el conjunto de validación.



El número de muestras del conjunto de validación pueden ser insuficientes por lo que se realiza para certificar el correcto funcionamiento la validación cruzada. En este caso se emplea un clasificador del conjunto, obtenido al azar y se realiza la prueba de validación. Esta acción se vuelve a ejecutar descartando en conjunto empleado y seleccionando al azar otro conjunto. De esta forma, se garantiza la propiedad de generalización.

Las características que deben cumplir el conjunto de patrones se fundamentan en la economía, velocidad, fiabilidad, capacidad de discriminación e independencia.

La economía se refiere al coste computacional. Dicho coste debe ser lo más bajo posible ya que sino el sistema no funcionaría a tiempo real. El coste computacional, además, repercute en la velocidad, ya que la obtención de los objetos debe ser lo más rápido posible ya que sino el sistema sería inviable.

Las características que se emplean para validar un objeto no tienen que estar relacionadas entre ellas, independencia, ya que si una está relacionada con el resto, la clasificación del objeto podría ser errónea. La característica relacionada no contiene información relevante por lo que se debe eliminar. Para conocer si una característica depende de otra se utiliza la covarianza. Si el resultado de aplicar este método estadístico se aproxima a un valor nulo, las características son independientes entre sí. En el caso de que los resultados se aproximen a 1 o a -1, las características son dependientes entre ellas.

La independencia de las características se adquiere mediante la matriz de covarianzas. Cada posición de la matriz corresponde con la covarianza de un par de características. Debido a ello, el rango de la matriz es  $N \times N$ .

La fiabilidad se muestra a partir del número de veces que se repite un valor que corresponde a una característica en el vector que constituye el conjunto de patrones. Los valores no tienen por qué ser iguales sino que pueden ser muy próximos entre ellos. La fiabilidad se obtiene mediante la diagonal de la matriz de covarianzas. Si los valores son elevados, la dispersión es mayor por lo que la fiabilidad disminuye.

El grado de diferencia entre los valores que forman las características es la capacidad discriminante. Esta capacidad se mide mediante el cociente de Fisher. Una característica se puede discriminar si el coeficiente de Fisher es elevado.

Si dos características son dependientes entre sí, se debe eliminar una de ellas mediante el empleo de uno de los siguientes procedimientos: fuerza bruta, eliminación e incorporación.



La fuerza bruta consiste en la selección del conjunto que mejor cumple el criterio de selección. Este tipo de procedimiento garantiza que los objetos encontrados en la imagen van a ser los que mejor cumplan el conjunto de características.

La eliminación se fundamenta en suprimir uno de ellos al azar y observar el rendimiento del sistema. A continuación, se suprime el otro y se analiza de nuevo el rendimiento. Se elimina la característica cuyo rendimiento ha sido menor.

Por último, la incorporación se basa en añadir características que provocan un incremento en el rendimiento del sistema. En este caso, no se suprime ninguna característica. Este método es el más rápido pero el menos óptimo.

Existen multitud de clasificadores algunos basados en modelos probabilísticos, en redes neuronales artificiales, en distancia, en ángulos, etcétera. Si se implementa el sistema inteligente de tráfico mediante cámaras de videovigilancia será imprescindible la selección tanto de las líneas de delimitación del carril como la línea de detención, en el caso de los vehículos y la selección de las líneas que constituyen el paso de cebra, en el caso de los peatones.

El clasificador según la distancia se basa en la distancia euclídea. Para ello, se introduce en el vector características la distancia que debe tener las rectas de estudio. La distancia de las rectas que más se aproximen a la distancia introducida en el patrón, se podrá considerar como válida.

El clasificador según el ángulo se caracteriza por obtener el ángulo de cada recta y compararlo con el ángulo patrón establecido. Se seleccionaran las rectas que más se aproximen al ángulo patrón. En el caso de las rectas verticales, el ángulo patrón será de  $90^\circ$  y en el caso de las rectas horizontales de  $0^\circ$ .

#### 2.4.2 COMUNICACIÓN ENTRE PROCESOS

La comunicación entre procesos es el envío de información de una entidad a otra. Para que exista comunicación se deben compartir recursos del sistema en el mismo instante de tiempo. Por ello, si dos programas se ejecutan en el mismo ordenador y su ejecución está entrelazada en el tiempo, dichos programas comparten un recurso del ordenador. A este hecho se le denomina concurrencia<sup>[38]</sup>.

Los sistemas operativos gobiernan el acceso a los recursos del sistema, procesadores, memoria y dispositivos de E/S. Debido a ello, si se intercambia información entre dos programas, el sistema operativo debe asignar un espacio de memoria para la lectura/escritura de los datos. Los programas que se encuentran en ejecución reciben el nombre de procesos.



Los sistemas operativos de hoy en día permiten ejecutar varios procesos al mismo tiempo, multiprogramación. La multiprogramación se caracteriza por ejecutar un proceso cuando este está listo para ello. En verdad, el sistema operativo ejecuta únicamente un proceso a la vez, pero de forma muy rápida por lo que cambia de un proceso a otro sin que el ser humano se dé cuenta de ello.

La concurrencia emplea tanto el hardware como el software. Los procesos se ejecutan en un mismo ordenador y comparten recursos por lo que pueden utilizar el mismo procesador y la misma memoria. Con respecto al software, la concurrencia se presenta en el manejo de señales, en comunicaciones y en compartir recursos entre procesos.

El cambio de proceso a ejecutar en un procesador se realiza mediante una interrupción. La interrupción en un programa es la consecuencia del ciclo de instrucción del procesador. El ciclo de instrucción se puede interrumpir por la ejecución de un programa con prioridad o por la introducción de una señal eléctrica en el procesador originada por un dispositivo periférico como por ejemplo al pulsar una tecla del teclado. En el momento en que se produce la interrupción de un programa, se almacena en memoria la ejecución efectuada hasta el momento de su detención para así continuar cuando el procesador le dé acceso a sus recursos. Las interrupciones realizadas por dispositivos periféricos son asíncronas ya que el procesador no planifica su ejecución.

Una señal es un aviso al programa para que realice una interrupción y se ejecuten determinadas acciones impuestas por el programador. El aviso lo puede enviar el procesador para notificarle el término de una operación de E/S o un error. Además, el aviso se puede recibir por teclado, como por ejemplo si se pulsa ctrl-c.

Las señales se pueden enviar de forma síncrona si la genera el mismo proceso o asíncrona en el caso de que la señal la genere un dispositivo periférico a partir de la acción del ser humano.

Los procesos generalmente requieren de un proceso que los coordine de manera correcta, por lo que los procesos se comunican con el proceso principal mediante colas, memorias compartidas o semáforos. El acceso a los datos por parte de los procesos, debe estar regido por un sistema de control ya que si no podrían interferir uno con el otro.

#### 2.4.2.1 COMUNICACIÓN CLIENTE-SERVIDOR

La comunicación entre procesos se basa en el modelo cliente-servidor<sup>[39]</sup>. El servidor se caracteriza por administrar los recursos y los clientes por acceder a ellos.



Los clientes comparten un mismo recurso y su orden de acceso, lo establece el servidor.

El servidor puede estar en el mismo o en distinto ordenador que los procesos clientes. Si está en distinto ordenador, la comunicación se realizará mediante la red y se implantará un protocolo orientado a la conexión mientras que si está en el mismo ordenador se empleará un protocolo sin conexión.

Los protocolos orientados a la conexión se caracterizan por poseer un servidor que espera a un cliente que solicita conexión. En el momento en que se establece dicha conexión, la comunicación se realiza mediante descriptores de archivo y se envían los datos entre el cliente y el servidor sin incluir la dirección IP.

Los protocolos orientados tanto a conexión como a sin conexión se estiman como si fuesen de bajo nivel ya que el programador conoce la existencia y la ubicación del servidor y este se conecta a ellos a través de la identificación que posee cada uno. Dicha identificación si se emplean colas de mensaje o memorias compartidas, se adquiere a partir de una o varias palabras y un identificador y si se utilizan sockets, se obtiene mediante la dirección IP y el puerto de conexión.

Si los clientes y el servidor se ubican en el mismo ordenador, el puerto único es FIFO, primero que entra, primero que sale. En el momento en que se conecta el servidor, los clientes que requieran intercambiar información con él, abren su FIFO y procesan la solicitud.

Los clientes pueden solicitar diferentes datos del servidor, por lo que se deberán abrir otros puertos FIFO ya que si un cliente se conecta al mismo puerto que se ha conectado otro, la información que le otorgue el servidor podría ser incorrecta. Este hecho se produce si la comunicación es privada o si la transferencia de datos es elevada. Si la comunicación con un proceso no es privada y se emplea únicamente un puerto FIFO, se atiende la solicitud de dicho proceso antes de aceptar solicitudes de otros procesos.

#### 2.4.2.2 MEMORIA COMPARTIDA EN SYSTEM V

La manera más óptima de intercambiar datos entre procesos es empleando una zona de memoria compartida<sup>[40][41]</sup>. El envío de un dato se efectúa escribiendo dicho dato en la memoria y cuando el proceso que debe recibir el dato lo requiera, accede al segmento de memoria reservado y lo lee.

El acceso del proceso a la memoria compartida se realiza a partir de direcciones virtuales. El encargo de traducir las direcciones virtuales en direcciones físicas es el núcleo. El espacio de direccionamiento virtual se divide en tres regiones texto, datos y pila. El texto constituye el código del programa, los datos son las variables que



emplea el proceso y la pila almacena los parámetros de la función, las variables locales así como el contador que indica en punto debe continuar el programa principal después de ejecutar las acciones de la función.

La memoria física se fracciona en páginas de tamaño variable. Este tamaño lo escoge el núcleo y lo puede variar según lo crea conveniente. El tamaño de la pila cambia según la función que se esté ejecutando y el tamaño de la zona de datos se modifica en función de la reserva de memoria que realicen los procesos. La transformación de direcciones virtuales a direcciones físicas se efectúa mediante tablas de páginas.

El sistema System V permite conectar varios procesos a un mismo segmento de memoria compartida. La dirección de acceso a la memoria es virtual por lo que su dirección física puede variar con el tiempo. Este cambio de dirección no produce fallos en la ejecución del proceso ya que este posee una dirección virtual que no se modifica.

Las funciones que se emplean para el control de una memoria compartida son: `shmget`, asigna una zona de memoria compartida o permite el acceso a una zona ya creada, `shmctl`, establece y modifica los parámetros de control que el núcleo asocia a cada segmento de memoria compartida, `shmat`, vincula un proceso con una zona de memoria compartida y `shmdt`, separa al proceso del segmento de memoria compartida.

La instrucción `shmget` obtiene el identificador de memoria con el que se controla el segmento de memoria compartida. Esta función se compone de tres argumentos, `key`, `size` y `shmflg`. El primer argumento `key`, es la clave que se obtiene mediante la función `ftok` a partir de un archivo y un número entero. El siguiente argumento, `size`, establece el tamaño que debe poseer la memoria para el almacenamiento de los datos. Por último, `shmflg` se caracteriza por indicar el modo de obtención del identificador. Esta bandera permite crear el segmento (`IPC_CREAT`), en el caso de que no lo esté y asignar los permisos de lectura/escritura de cada proceso.

El comando `shmctl` realiza acciones de control sobre el segmento de memoria compartida. Dicho comando emplea la identificación de memoria obtenida con la instrucción `shmget`, el tipo de operación a efectuar y la estructura que contiene la información sobre la memoria compartida. Las operaciones que se pueden realizar son: `IPC_STAT`, adquiere el estado de la memoria compartida que se establece en la estructura, `IPC_SET`, inicializa las variables de control de la estructura, `IPC_RMID`, suprime el segmento de memoria compartida que posee el identificador expuesto en el primer argumento, `SHM_LOCK`, bloquea el segmento de memoria, y `SHM_UNLOCK`, desbloquea el segmento de memoria.



La asignación de direcciones virtuales se implementa con la llamada a la función `shmat`. En el caso de que se requiera desvincular el proceso de la dirección de memoria, se emplea la instrucción `shmdt`.

El comando `shmat` utiliza la identificación de la zona de memoria compartida, la dirección virtual donde se desea que comience dicha zona y una máscara de bits que fijan la forma de acceso a la memoria. En el caso de que el dato que se ubica en el segmento de memoria compartida sea solo de lectura, la bandera se establecerá como `SHM_RDONLY`. Esta función retorna la dirección virtual de la zona de memoria compartida.

En la instrucción `shmdt`, el único argumento que contiene hace referencia a la dirección virtual de la zona de memoria compartida que se desea desvincular del proceso.

La lectura o la escritura del dato en el segmento de memoria compartida se realiza a través de punteros una vez obtenida la dirección virtual a la que está unida.

#### 2.4.2.3 SECCIONES CRÍTICAS Y SEMÁFOROS

Los procesos pueden compartir un mismo recurso del sistema, por lo que el orden de acceso a dicho recurso debe estar controlado para que un proceso no adquiera un dato que no le corresponde<sup>[42]</sup>. La parte del código que se ejecuta en la que un solo proceso accede a un recurso del sistema, se denomina sección crítica. En la sección crítica se adquiere o se modifica el dato que se ubica en el recurso compartido.

El acceso a la sección crítica se caracteriza por poseer completa independencia, progreso y límites de espera. La independencia se fundamenta en que un solo proceso puede estar en su sección crítica. El progreso se basa en que si ningún proceso se encuentra en la sección crítica y uno quiere entrar, puede hacerlo. Todos los procesos deben acceder a la sección crítica por lo que ningún proceso debe estar de manera indefinida esperando la orden de acceso. A este hecho se le conoce como límite de espera.

El acceso a la sección crítica de un proceso se realiza mediante un semáforo basado en un número entero variable capaz de realizar las operaciones de acceso o de espera. Si la variable es negativa, el semáforo está en espera y el proceso no puede acceder a su sección crítica hasta que dicha variable no pasa a un valor positivo. Esta variable la modifica el proceso que se encarga de controlar el acceso a la sección crítica de los diferentes procesos que constituyen el programa de forma que no exista exclusión mutua.



El control de acceso a la sección crítica se realiza mediante una llamada al hardware del sistema. Esta llamada se denomina TestAndSet. Si un proceso solicita el acceso a su sección crítica pide dicha orden mediante el TestAndSet. Este sistema solicita el acceso al kernel y si este observa que ningún proceso está ejecutando su sección crítica, le permite el acceso. En el caso de que si esté ejecutando su sección crítica, el kernel le pone en espera hasta que el proceso que está escribiendo o leyendo el dato que se ubica en el recurso compartido finalice.

El sistema operativo UNIX posee semáforos propios para garantizar la intercomunicación de procesos denominados System V. El semáforo lo abre un proceso al realizar una llamada al sistema mediante la función semget. Esta llamada origina una estructura que contiene la información del semáforo y la envía al núcleo. Este retorna una identificación para tratar el semáforo en la totalidad de los procesos en que se emplee.

El sistema de semáforo implantado en System V se caracteriza por controlar un grupo de semáforos como si fuera un único semáforo. El control del grupo de semáforos, se realiza estableciendo las características de cada uno de ellos en los parámetros de configuración.

La creación de un semáforo se realiza mediante la función semget. Dicha función se compone de tres argumentos: la identificación del grupo de semáforos, el número de semáforos que constituyen el grupo y la bandera. La identificación del semáforo se efectúa mediante la función ftok. Para ello, se introduce en los argumentos de la función un nombre de archivo y un número cualquiera. Si el semáforo no existe y lo crea el proceso se implementa IPC\_CREAT junto con el número de permiso en la bandera. En el caso de que el semáforo ya este creado, solo se implementa el número de permiso. El número de permiso se fija según sea solo lectura, solo escritura o ambos y los propietarios que pueden tener acceso.

La configuración inicial de cada semáforo se efectúa con la función semctl. En el primer argumento se introduce la identificación del grupo de semáforos, en el segundo el número de semáforo que se requiere inicializar, el tercer argumento especifica los comandos por ejecutar y el último argumento establece la posición inicial del semáforo.

Los comandos que se emplean en el tercer argumento pueden ser: GETVAL, obtiene el valor inicial de un semáforo específico, GETPID, retorna el identificador del último proceso que empleo el semáforo, GETNCNT, adquiere el número de procesos que están esperando para acceder a la sección crítica, SETVAL, inicializa el semáforo con el valor introducido en el último argumento, IPC\_RMID, suprime el grupo de semáforos que poseen la identificación del primer argumento e IPC\_SET, inicializa los permisos del semáforo.



El control de un semáforo que se encuentra dentro de un grupo de semáforos, se efectúa mediante la identificación del grupo de semáforos, la estructura de elementos que contiene los parámetros de configuración de cada semáforo y la cantidad de semáforos que se requiere tratar en la operación.

La estructura que posee los parámetros de los semáforos contiene tres variables, `sem_num`, número del semáforo que se desea tratar, `sem_op`, acción que debe realizar el semáforo, espera o paso, y `sem_flg`, bandera que determina las opciones de operación.

La variable `sem_op` puede adquirir un valor positivo, negativo o nulo. Si dicha variable adquiere un valor negativo, el proceso está bloqueado hasta que otro proceso le desbloquee modificando el -1 por el 1. En caso contrario, si la variable posee un valor positivo, permite el paso del proceso a la sección crítica hasta que otro proceso lo bloquee cambia el 1 por el -1. Si el valor de la variable `sem_op` es nulo, y el semáforo no tiene un valor nulo, se bloquean todos los procesos que están pendiente de acceder a su sección crítica hasta que el valor de dicho semáforo sea nulo.

#### 2.4.2.4 SEÑALES

Los eventos asíncronos que se realizan durante la ejecución de un programa que funciona de forma cíclica se efectúan mediante señales<sup>[43]</sup>. Una señal es un aviso efectuado por software para la ejecución de un determinado evento.

La señal es depositada si el proceso inicia una acción en el momento en que se recibe. El tiempo de vida de una señal es la duración entre el instante de tiempo en que se recibe y el fin de la ejecución del evento que produce. Una señal está pendiente si esta se ha generado pero todavía no se ha ejecutado la acción que efectúa la interrupción.

Un proceso se puede interrumpir por más de un señal, por lo que debe existir un manejador de señal para cada señal de interrupción. Si el proceso recibe la señal y esta está bloqueada, el proceso no la ignora sino que la deposita hasta que el proceso la libere.

Los nombres de las señales que se emplean en la interrupción de un proceso comienzan por SIG y se definen en la librería `<signal.h>`. Dichos nombres simbolizan números enteros mayores que cero. A continuación, se muestra una tabla con las señales más utilizadas.



Nombre de la señal	Nº	Significado
SIGINT	2	Interrupción si se pulsa Ctrl-C
SIGABRT	6	Terminación anormal como la iniciada por abort
SIGFPE	8	Error en operación aritmética como la división por 0
SIGPIPE	13	Fallo en la escritura/lectura por conexión rota en memoria compartida, sockets, etcétera.
SIGALRM	14	Señal de espera como la iniciada por alarm
SIGKILL	9	Cierre de un proceso
SIGUSR1	10	Señal definida por el usuario 1
SIGUSR2	12	Señal definida por el usuario 2

*Tabla 1. Señales de control de trabajo*

Determinadas señales como SIGFPE, se activan si se produce un error en el sistema y otras señales se originan de forma cíclica o por acciones externas. Las señales se envían a los procesos que posee el usuario. Para ello, se debe conocer el identificador de los procesos entre los que se envían y se reciben las señales.

Las señales tal y como se ha descrito anteriormente se pueden bloquear hasta que el proceso las desbloquee. Si la señal está bloqueada, el proceso continúa con la ejecución del programa. Las señales bloqueadas se agrupan en la máscara de señal. Las señales ignoradas se depositan pero el proceso se deshace de ella por lo que no se realiza ningún evento. La instrucción sigaction con el manejador de señal SIG\_IGN se emplean para ignorar una señal.

Las señales se pueden agrupar formando un conjunto de señales mediante la función sigemptyset, si no se declara ninguna señal en el instante en que se inicializa el grupo, y la función sigfillset, si se declaran todas las señales en el momento de inicializar el grupo. En el caso de que se requiera añadir o eliminar una señal se utiliza la función sigaddset y la función sigdelset, respectivamente.

La utilización de señales reduce el coste de cómputo de la CPU ya que un proceso se puede detener hasta que reciba una señal sin tener que comprobar de forma continua la modificación de una variable de control. Esto se realiza con las instrucciones pause y sigsuspend.

EL comando pause se caracteriza por pausar el proceso hasta que una señal procedente del mismo o de otro proceso le permita continuar. En el momento en que



se atrapa la señal, el proceso volverá a su ejecución hasta que llegue de nuevo al pause y este se suspenda.

Si la señal que reactiva el proceso está suspendida, el programa se mantendrá pausado. Para que este problema no ocurra, se debe desbloquear la señal al mismo tiempo que se inicia el pause. Esto se realiza con la función `sigsuspend`.

#### 2.4.2.5 SOCKETS

El sistema UNIX permite el intercambio de información entre dos procesos que se ubican en el mismo ordenador o en ordenadores diferentes mediante el sistema de comunicación basado en sockets<sup>[44][45]</sup>. Un socket es un punto de comunicación por el que un proceso puede enviar o recibir información por lo que la transmisión de datos puede ser unidireccional o bidireccional. Los procesos establecen conexión entre ellos si poseen el mismo dominio. Debido a ello, un proceso que posea un dominio diferente no establecerá conexión con los procesos anteriores.

El socket se identifica con un descriptor. Este permite la redirección de los datos a los sockets y por tanto el empleo de procesos sobre la red. El descriptor se adquiere con la función `socket` que permite realizar las operaciones de escritura/lectura una vez que se ha adquirido la dirección del dominio del proceso que crea el socket mediante la instrucción `bind`. Las características de los sockets se ubican en la librería `<sys/socket.h>`.

El intercambio de información entre dos procesos a partir de sockets se basa en la familia o dominio de la conexión y en el tipo de conexión.

La familia o dominio de la conexión es el grupo de todos los conectores sockets que poseen las mismas características de protocolo, la manera en que se constituye la dirección de la red, el puerto, etcétera.

La transmisión de información entre procesos, tal y como se ha explicado anteriormente, se puede realizar en el mismo ordenador o en diferentes ordenadores. Si no está orientado a la conexión, procesos en el mismo ordenador, se denomina socket de datagrama y si está orientado a la conexión se denomina socket de flujo. Los datagramas no poseen conexiones permanentes por lo que cada paquete de información puede seguir una ruta diferente y no se garantiza que lleguen en el mismo orden en que se enviaron. El protocolo orientado a conexión se fundamenta en la búsqueda de enlaces libres entre los ordenadores que poseen los procesos a conectar. Este tipo de conexión se caracteriza por enviar los datos por la misma ruta por lo que el orden de llegada es el mismo que el orden de salida.



La conexión mediante sockets se efectúa con un mecanismo de direccionamiento. De esta forma, los procesos saben a qué ordenador se deben conectar para transferir los datos desde el proceso origen hasta el proceso destino. El mecanismo de direccionamiento difiere si los procesos se conectan en el mismo ordenador o en diferentes ordenadores.

En el caso de que los procesos intercambien información entre diferentes ordenadores, se emplea el dominio AF\_INET y se implementan las características de la conexión en la estructura struct sockaddr\_in. Esta estructura se constituye de cuatro parámetros: short sin\_family, dominio UNIX, u\_short sin\_port, número de puerto (Max. 16 bits), struct in\_addr sin\_addr, dirección IP, y char sin\_zero[8], 8 bytes no usados. Si los datos entre procesos se envían/reciben en la misma máquina se usa el dominio AF\_UNIX y se establece la conexión mediante los dos datos introducidos en la estructura sockaddr\_un. Dicha estructura se compone de dos parámetros short sun\_family, dominio UNIX, y char sun\_path[108], ruta de ficheros.

La dirección IP que se establece en el parámetro s\_addr de la estructura sin\_addr debe estar en código binario, por lo que se emplea la función inet\_addr para traducir la dirección decimal del nodo a formato binario. Si se desea obtener la dirección IP a partir del código binario, se utiliza la función inet\_ntoa.

El sistema emplea puertos para los protocolos de internet TCP y UDP. Dichos puertos se ubican entre el intervalo comprendido entre 1 y 1.023 por lo que se debe escoger un puerto fuera de dicho intervalo ya que sino el sistema funcionaría de forma incorrecta. En el caso de que los procesos a comunicar se ubiquen en diferentes máquinas hay que establecer el orden con que se guardan en memoria los bytes que constituyen el número de puerto. Este problema se origina ya que hay ordenadores que almacenan el byte más significativo en la parte baja de la memoria y el byte menos significativo en la parte más alta y ordenadores que lo almacenan de forma inversa, es decir, el byte más significativo en la parte más alta y el menos significativo en la parte más baja.

Para que no se produzca un error en la posición de los bytes, se almacenan en la memoria de las máquinas de idéntica forma. Existen cuatro disposiciones de almacenamiento htonl y htons, convierten el formato del ordenador al formato que posee la red, y ntohl y ntohs, transforman el formato de la red al formato que posee el ordenador. La terminación de las disposiciones indica el tipo de dato, l se refiere al tipo unsigned long y s al tipo unsigned short.

La comunicación mediante sockets se basa en el modelo cliente-servidor. Tanto el cliente como el servidor realizan determinadas acciones antes de que se establezca comunicación entre ellos.



El proceso servidor se encarga de crear el punto de comunicación y de comunicar a la red su intención de aceptar peticiones de conexión con otros procesos u ordenadores. Hasta que el cliente no le solicita conexión, el servidor se mantiene a la espera. Cuando se realiza la conexión, puede comenzar el intercambio de información.

El programa cliente se caracteriza por abrir el punto de comunicación y unirse al nodo creado por el servidor. Una vez que este le solicita conexión al servidor y este la acepta, se puede iniciar la transmisión de datos de un proceso a otro.

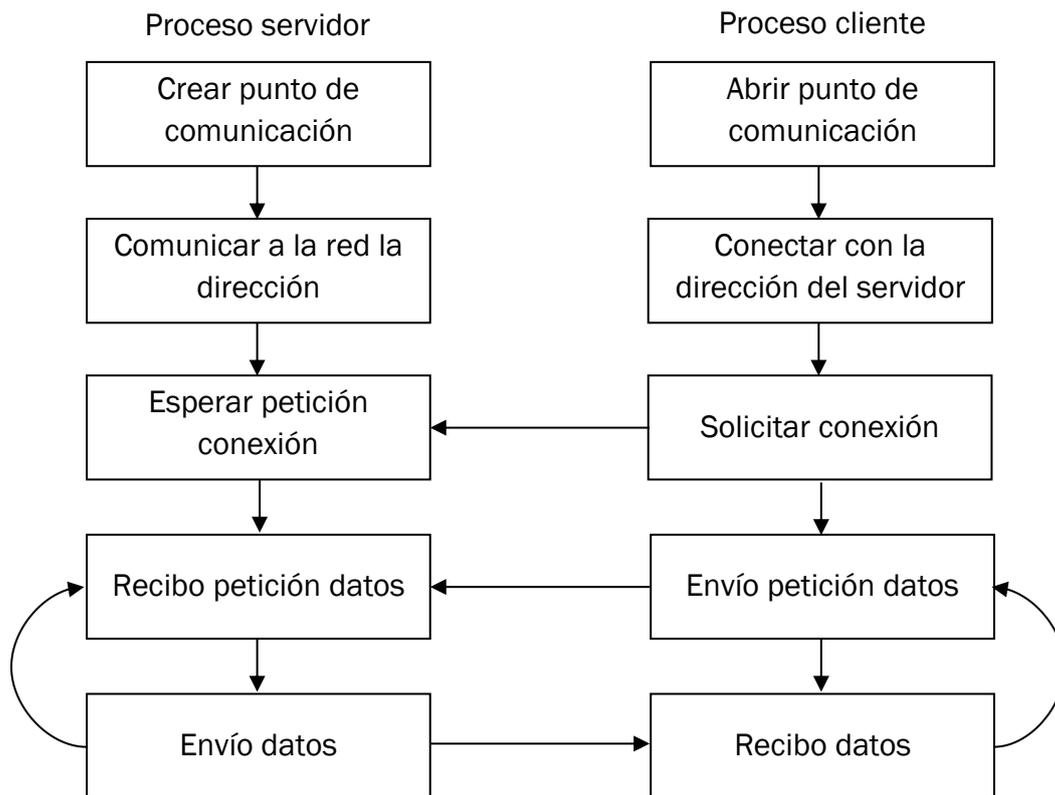


Figura 8. Diagrama flujo control sockets

La creación y la apertura de un punto de comunicación se realizan mediante la función socket. Esta función contiene un total de tres argumentos: af, establece la familia de direcciones, type, precisa el tipo de comunicación, si está o no orientado a la conexión, y protocol, detalla el tipo de protocolo que se emplea en la comunicación. Si este último argumento se establece con un valor nulo, el sistema selecciona de forma automática el protocolo a usar.

Existen dos familias de direcciones dependiendo del tipo de conexión, AF\_UNIX y AF\_INET. La familia AF\_UNIX se caracteriza por emplear protocolos internos de UNIX para el intercambio de información entre procesos que se ubican en el mismo



ordenador. La familia AF\_INET emplea protocolos de internet como por ejemplo TCP. Este tipo de protocolo se emplea para intercambiar datos entre procesos que se encuentran en diferentes ordenadores.

El tipo de comunicación entre procesos depende de la familia de direcciones. La familia AF\_UNIX usa SOCK\_DGRAM, conector con un protocolo no orientado a la conexión, y la familia AF\_INET utiliza un conector con un protocolo orientado a la conexión, SOCK\_STREAM.

Tanto la comunicación a la red de la dirección de conexión por parte del servidor como la conexión por parte del cliente a la dirección del servidor se efectúa mediante la función bind. Dicha función posee un total de tres argumentos, sfd, identificador devuelto por la instrucción socket, addr, estructura que contiene las características de la conexión y addrlen, indica el tamaño de la estructura anterior.

El servidor espera hasta un cliente solicite conexión a partir de la función listen. Esto sucede en comunicaciones orientadas a la conexión. Esta instrucción origina una cola de espera para los procesos que desean intercambiar datos con el servidor. Para ello, se introduce en los argumentos de la función, el identificador del socket y el número de peticiones que puede atender.

Para la solicitud de conexión por parte del cliente al servidor se emplea la función connect. Dicha solicitud se realiza mediante el identificador del socket obtenido en con la instrucción socket, la estructura que contiene los parámetros de conexión y el tamaño de dicha estructura. Si el tipo de comunicación es SOCK\_DGRAM, protocolo no orientado a la conexión, connect detalla la dirección de la conexión con el proceso remoto pero no se conecta con él. En el caso de que el tipo de comunicación sea SOCK\_STREAM, protocolo orientado a la conexión, connect enlaza los procesos entre los que se va a producir un intercambio de información.

La aceptación o no de la petición de conexión del proceso cliente, la realiza el proceso servidor a partir de la función accept mediante el identificador del socket, la estructura que posee los parámetros de conexión y el tamaño de dicha estructura. El servidor puede tener varias conexiones a la espera, por lo que dicha función acepta las conexiones en el orden que asigna la instrucción listen.

En el momento en que el servidor acepta la conexión de un cliente, se puede iniciar el intercambio de datos entre ellos. El envío de datos se realiza con la llamada a write. El uso de dicha función implica establecer en ella el descriptor del socket, la dirección de memoria del dato a enviar y su tamaño. La recepción de datos se efectúa con la llamada a read. Para la utilización de esta instrucción se incluyen en sus argumentos el descriptor del socket, la dirección de memoria donde se van a ubicar los datos y el tamaño de memoria que hay que asignar para almacenar la información recibida.



En el caso de que se requiera el cierre de la comunicación de algún proceso, se llama a shutdown. El cierre de la comunicación puede ser total, si se cierra tanto la comunicación del servidor como la de los clientes o parcial si solo se cierra la comunicación de algún cliente. La función shutdown suspende la comunicación del descriptor del socket expuesto en su primer argumento. Dicha suspensión puede ser solo de recepción, SHUT\_RD, de envío, SHUT\_WR, o de ambas, SHUT\_RDWR. El tipo de suspensión se establece en el segundo argumento de la función.

## 2.5 EL INTERNET DE LAS COSAS

En la actualidad millones de dispositivos se conectan a internet para intercambiar información. El Internet de las cosas (Internet of things, IOT) se establece como un mundo inteligente en el que todo se pueda controlar desde internet, viviendas, industria, ciudades, transporte, etcétera<sup>[46]</sup>.

La perspectiva del internet de las cosas se basa en el acceso a la información desde cualquier dispositivo, smartphone, tablet, ordenador a través de la red. Para ello, se instalan sensores que detectan y evalúan cualquier parámetro del mundo físico y transmiten dichos parámetros a la red. De esta forma, se convierte un aparato en inteligente. El control de estos dispositivos se puede realizar desde un servidor automático o de modo manual, es decir, a partir de la interacción del ser humano mediante un dispositivo conectado a la red.

Los dispositivos conectados a internet poseen inteligencia si las acciones que deben llevar a cabo las toma un ordenador. La solución adoptada por el ordenador será más óptima que la solución que pudiese aplicar el ser humano ya que la máquina puede adquirir información de varios dispositivos y tratar dicha información para que la solución obtenida sea la más eficaz.

El internet de las cosas se subdivide en cuatro fases: dispositivo, sensor capaz de adquirir información del mundo físico, red, permite el transporte de información, soporte, software que permite ejecutar aplicaciones en un sistema operativo, y aplicación, programa que controla el dispositivo conectado a internet.

La información adquirida por los sensores y el envío de dicha información a la red, se realiza mediante redes de sensores inalámbricos denominados WSN. Este tipo de redes disponen de microprocesadores constituidos por un pequeño procesador, una memoria muy acotada, un microchip capaz de conectarse a redes WiFi y uno a varios sensores.

Las características principales de este tipo de redes WSN son la conexión inalámbrica, la auto-organización y su bajo consumo.



Las redes WSN pueden estar conectados entre sí y una de ellas enviar la información a la red. A este tipo de red se le denomina multi-hop. En el caso de que una única red se conecte directamente a internet, el tipo de conexión se denomina single-hop. El radio de la red inalámbrica alcanza una distancia inferior a 100 metros.

Si la conexión es de tipo multi-hop, las redes de sensores inalámbricas se auto-organizan. Esta propiedad se fundamenta en el hecho de que cada red conoce el nodo del que va a recibir la información y el nodo al que debe enviar datos. Debido a ello, se pueden colocar varias redes WSN de tal manera que solo una necesita conexión a la red.

Los aparatos que conforman dichas redes tienen un bajo consumo por lo que no es necesario la instalación de una fuente de alimentación. La energía la puede obtener mediante pilas o paneles solares. Otra forma de ahorrar energía es mantener el dispositivo en modo dormido el mayor tiempo posible.

El gran número de dispositivos conectados a la red implica emplear el protocolo de internet IPv6. Este tipo de protocolo se emplea por todos los desarrolladores de softwares ya que está estandarizado por la Internet Engineering Task Force, IETF.

Los dispositivos conectados a internet establecen conexiones de persona a persona P2P, de persona a máquina P2M y de máquina a máquina M2M. La conexión P2P se emplea para el envío de mensajes o la transferencia de archivos entre personas, la conexión P2M se utiliza en el ámbito doméstico o en el control de una máquina por un operario ubicado en otro lugar y la conexión M2M se usa en la sincronización entre máquinas.

El internet de las cosas aplicado a la industria produce un incremento de la productividad ya que aumenta la eficacia de las tareas realizadas al tener datos más precisos, un control de stock y logística por lo que se disminuyen costes y un análisis de datos, por lo que se mejora el funcionamiento de cualquier sistema.

El internet de las cosas se aplica a seis grandes grupos, smart cities, smart home, smart energy, smart health, industry 4.0 y connected vehicle.

Las smart cities o ciudades inteligentes se caracterizan por optimizar las infraestructuras a partir de la información obtenida por las redes WSN. Los recursos que posee una ciudad se optimizan mediante una mejora en la organización del transporte y en un ahorro energético.

Las smart home o casas inteligentes son viviendas automatizadas. Dichas viviendas se caracterizan por poseer sensores capaces de ahorrar energía, de dotar de seguridad a la casa, de permitir el acceso mediante tarjeta o reconocimiento facial y de proporcionar comodidad a sus habitantes.



Tanto en el ámbito empresarial como en el ámbito doméstico se puede ahorrar energía mediante dispositivos conectados a internet, smart energy. El ahorro de energía conlleva una disminución de gasto en facturas y un menor impacto en el medio ambiente. El ahorro se establece mediante aparatos que analicen patrones de consumo y creen un aviso en el caso de que se produzca un gasto de energía que se pudiese suprimir.

Los smart health son sensores que se instalan en el cuerpo humano y miden el riesgo de fallo de los órganos, la tendencia a tener algún problema de salud y la trayectoria que tienen los organismos. La trayectoria se emplea para conocer el comportamiento que ha tenido el cuerpo antes de un problema así como para avisar al usuario de que debe dejar la actividad que esté realizando y acudir al médico de inmediato.

La industria 4.0 o cuarta revolución industrial es el análisis de los datos captados por los sensores, big data, para optimizar el funcionamiento de una planta mediante la inteligencia artificial. Debido a ello, se mejora la producción, por lo que se diseña y se vende el producto en un menor tiempo, y se incrementa la relación con el cliente ya que cualquier fallo se puede corregir en el momento desde cualquier lugar. Este último hecho disminuye el tiempo en que la fábrica está detenida.

El último grupo en que se divide el internet de las cosas es el denominado connected vehicle. En la actualidad, los vehículos nuevos están equipados con acceso a internet por lo que aumenta el confort de los conductores. Las mejoras de los vehículos basados en esta tecnología se da en el aviso de accidentes, en mapas en línea, en la apertura o cierre de los vehículos, en conectar de forma remota el aire acondicionado antes de llegar a él, en la carga que poseen las baterías de un vehículo eléctrico, etcétera.

Un proyecto IOT requiere de elementos hardware, de lenguajes de programación capaces de interpretar los datos de los sensores y enviar dichos datos a la red, de plataformas, de APIs y de almacenamiento de datos en la nube.

### 2.5.1 HARDWARE

Un sistema fundamentado en el internet de las cosas, puede adquirir información de internet para establecer las condiciones de funcionamiento de un aparato, como por ejemplo un sistema que elige la hora adecuada para regar los cultivos, o puede obtener información de sensores y enviarla a internet para su análisis, como por ejemplo, un sistema inteligente de gestión de tráfico.

En el caso de que el sistema se caracterice por adquirir información de internet para realizar una acción, se emplearan actuadores conectados al microprocesador,



mientras que si se envían datos a internet para su estudio, se utilizarán sensores que se conectarán a un microprocesador o a un microcontrolador. Los microprocesadores o microcontroladores deben poseer un módulo WiFi para su conexión a internet.

Los microcontroladores se caracterizan por poseer un funcionamiento autónomo, por lo que se emplean en sistemas embebidos para el control de máquinas, en la robótica, en la domótica, etcétera. Estos dispositivos se constituyen de una unidad de procesamiento, CPU, de una memoria ROM y RAM y de periféricos de entrada y salida<sup>[47]</sup>. La actividad que realiza el microcontrolador depende del programa que esté compilado en su memoria, por lo que no se pueden realizar otras acciones que no estén programadas. El número de bits de instrucciones que admite la CPU depende del microcontrolador. La mayoría de ellos, son de 8 bits pero también los hay de 4, 32 y 64 bits. La memoria interna puede ser de tipo ROM, EPROM, EEPROM o flash que permite almacenar programas desde 1 Kb hasta varios Mb. La capacidad de la memoria RAM, se encuentra en el intervalo comprendido entre 1 y 32 Kb y la velocidad del procesador varía según el tipo de microcontrolador desde el 1 MHz hasta los 40 MHz. Las partes mencionadas, CPU, ROM, RAM y I/O se integran dentro del mismo circuito, por lo que no emplean pines para el intercambio de información entre ellos.

Los microprocesadores ejecutan programas a partir de operaciones a bajo nivel, sumas, restas, multiplicaciones y divisiones, y disponen los datos que se obtienen de dichas operaciones en la memoria o en los dispositivos de entrada o salida a partir de los buses de direcciones que se conectan a las patillas del microprocesador. Estos se componen de una memoria caché, de una unidad aritmético-lógica, de registros, de una memoria RAM y de puertos. En la memoria caché se ubican los datos que podría utilizar el microprocesador para las acciones que se están realizando. De este modo se reduce el tiempo de procesamiento ya que los datos no se adquieren de la memoria RAM. La unidad aritmético-lógica realiza las operaciones a bajo nivel mencionadas anteriormente. Esta se encarga de obtener u ubicar un dato en la memoria RAM o de escribir un dato en el registro. Los registros se caracterizan por ser memorias de pequeño tamaño en el que se introducen los datos de la instrucción antes de realizar la operación en la unidad aritmético-lógica. La comunicación del microprocesador con los dispositivos externos a él se realiza a través de los puertos. Cada dispositivo externo tiene asignado un número de puerto y de esta forma, el microprocesador remite los datos al dispositivo externo correspondiente a través de un bus de datos. Los microprocesadores no disponen de memoria RAM ni de memoria ROM, ni tampoco de pines de I/O. Estos poseen un bus de datos que transmite la información entre el microprocesador y los dispositivos mencionados que se ubican en circuitos externos al microprocesador. Un microprocesador puede ejecutar varios programas simultáneamente en diferencia con el microcontrolador que únicamente puede ejecutar un programa a la vez.



Existen una multitud de microprocesadores o microcontroladores capaces de realizar las funciones que requiere un sistema IOT. En este apartado se estudiarán los más relevantes entre los que se encuentran, Raspberry Pi, BeagleBone, Arduino y ESP8266<sup>[48]</sup>. Los dos primeros se caracterizan por ser microprocesadores mientras que los dos últimos son microcontroladores.

La Raspberry Pi es un microprocesador con unas características inferiores a un ordenador pero capaz de tratar secuencias de imágenes procedentes de cámaras de videovigilancia. Este dispositivo posee un módulo WiFi y tarjeta Ethernet, por lo que puede recibir información de otras redes WSN y enviar dicha información a través del puerto Ethernet a internet o adquirir la información mediante las entradas digitales y remitirla a internet a través de su módulo WiFi. Se compone de un total de 40 pines que se comportan como entradas o salidas digitales dependiendo del uso que se las de. Este tipo de microprocesador no dispone de entradas ni salidas analógicas, por lo que este es su inconveniente ya que habría que realizar un circuito que transformase la señal analógica en señal digital o emplear un Arduino. En este último caso, se conectaría la señal analógica a dicho dispositivo y se transmitirían los datos a través del puerto serial. El precio de la Raspberry Pi es ligeramente superior al del Arduino, por lo que la elección de uno u otro depende de su uso.

El BeagleBone posee una velocidad de procesador y una memoria RAM inferior a la de la Raspberry Pi pero mayor número de entradas y salidas digitales. Además, dispone de entradas y salidas analógicas, por lo que no sería necesario la implantación de un circuito que transforme la señal analógica en digital o el empleo de un Arduino para ello. Este microprocesador posee módulo WiFi por lo que puede transmitir información sin necesidad de cables. La finalidad de este tipo de microprocesador es su empleo con un gran número de sensores ya que no posee tanta capacidad computacional como para tratar programas con un peso elevado como la Raspberry Pi. El precio del BeagleBone es bastante superior con respecto al coste conjunto de la Raspberry Pi y del Arduino.

El Arduino se caracteriza por ser un microcontrolador ya que posee la CPU, la memoria RAM, la memoria ROM y los pines de I/O en el mismo circuito. Debido a ello, su velocidad de procesamiento es menor con respecto a los microprocesadores por lo que su empleo se reduce al tratamiento de los datos de los sensores y a la activación o desactivación de un actuador. Los datos que se obtienen de los sensores se conectan a las entradas analógicas mientras que los actuadores a las entradas digitales. Este aparato no dispone de módulo WiFi, por lo que habría que conectar un módulo para transmitir información a la red o a otros dispositivos. El precio del Arduino es bajo, por lo que si se desea implementar un sistema IOT basado



únicamente en sensores o actuadores sin que el programa requiera un coste computacional elevado, se recomienda utilizar este dispositivo.

El dispositivo ESP8266 es un microcontrolador con un módulo WiFi incorporado, por lo que se puede conectar directamente a la red a partir de un punto de acceso o intercambiar información con otro dispositivo que crea una red WiFi, Ad-Hoc. Este aparato posee una velocidad de procesador y una memoria RAM demasiado baja como para realizar tareas que conllevan un elevado coste computacional, por lo que se suelen emplear para captar información a través de sus entradas y salidas de propósito general y enviar dicha información a otro dispositivo con mayor capacidad de procesamiento mediante la señal WiFi. También se emplean para activar o desactivar aparatos que se conectan en sus salidas digitales en el momento en que recibe la orden. Dicha orden la puede obtener directamente de internet o de un dispositivo que crea una red Ad-Hoc.

### 2.5.2 LENGUAJES DE DESARROLLO

El intercambio de información entre la plataforma que se ubica en la nube y los dispositivos mencionados en el apartado anterior, Raspberry Pi, BeagleBone, Arduino y ESP8266 se realiza mediante protocolos que se estipulan en el lenguaje de programación utilizado. Existen bastantes tipos de lenguajes que permiten establecer la comunicación entre la plataforma y el dispositivo entre los que destacan, C, C++, Java, Javascript y Python<sup>[49]</sup>.

Los lenguajes de programación C y C++ se pueden utilizar en todos los dispositivos descritos a excepción del Arduino y del ESP8266 que solo admiten C. Estos tipos de lenguajes se caracterizan por ser de medio nivel ya que poseen aspectos que permiten programar de la misma forma en que piensa el ser humano omitiendo detalles relativos al microprocesador, lenguaje de alto nivel, y además, permite controlar determinados parámetros del microprocesador, lenguaje de bajo nivel. Estos lenguajes requieren traducirse a lenguaje máquina, binario, para que el ordenador comprenda las acciones que debe realizar.

La Raspberry Pi y el BeagleBone admiten el lenguaje de programación Java. Este lenguaje se caracteriza por estar orientados a objetos. Las ventajas de Java respecto a C y C++ radican en la eliminación de las funciones menos usadas y más confusas. Además, libera la memoria que no se utiliza sin tener que aplicar una función para ello, como ocurre en C y C++. El lenguaje de programación Java es interpretado y compilado por lo que se reduce el tiempo ejecución ya que los programas se pasan, en primer lugar, a bytecode y en el momento de su ejecución se traducen a código máquina. Este hecho aumenta la seguridad ya que se controla el acceso a los



elementos hardware y a los ficheros, por lo que se impide la transmisión de código malicioso.

El lenguaje de programación JavaScript se emplea principalmente en páginas web. Dicha páginas contienen el programa por lo que se permite una mayor interacción con el usuario. Este tipo de lenguaje se utiliza para eventos asíncronos, con entradas y salidas que no se bloquean entre sí. El JavaScript se puede usar en la Raspberry Pi y en el BeagleBone.

Por último, el lenguaje de programación Python se caracteriza por ser de propósito general, por lo que permite desarrollar programas o scripts y orientado a objetos. Este lenguaje es interpretado, ya que no se traduce a código máquina sino que un programa, denominado interprete, ejecuta directamente el código. Debido a ello, se puede ejecutar un programa realizado en Python en cualquier sistema operativo, Unix, Windows, Mac OS, etcétera. El microcontrolador Arduino y ESP8266 no permiten el empleo de este lenguaje de programación.

### 2.5.3 PLATAFORMAS

Las plataformas son los alojamientos web que permiten observar y analizar los datos que envía el dispositivo a internet. Existen tres tipos de plataforma. La elección de una u otra depende de la función que realice el dispositivo conectado a internet y si se orienta a un ámbito empresarial, startups o doméstico.

En el ámbito empresarial, tanto Google como Microsoft o Apple, permiten conectar a internet un número elevado de dispositivos, por lo que si una empresa requiere que todos sus datos de fabricación estén en la red, debe contratar un alojamiento en una de las empresas mencionadas. En el caso de que un sistema IOT se implemente en una startup, las plataformas son de pago aunque existe la posibilidad de utilizarlas de forma gratuita con limitaciones en la cantidad de datos enviados/recibidos y con restricciones en el número de aparatos conectados. Si el sistema basado en IOT se emplea en el ámbito doméstico, existen plataformas gratuitas. Este tipo de plataformas también limita el número de mensajes enviados pero no cancela la cuenta una vez que se pase el periodo de prueba. A continuación, se analizan diferentes plataformas orientadas tanto a las startups como al ámbito doméstico.

La plataforma Samsung Artik Cloud posee su propio dispositivo para transmitir la información a internet aunque permite que los aparatos hardware estudiados anteriormente intercambien información con ella. En el caso de que se requiera conectar el dispositivo Arduino con esta plataforma, se realiza mediante el protocolo de comunicación RESTful empleando JSON ya que este no dispone de una librería



específica que realice la conexión con dicha plataforma<sup>[50]</sup>. Esta plataforma permite el envío de información a otros servidores en la nube, como Instagram o Google Drive, mediante el protocolo ITFFF. Existen diferentes contratos según el ámbito en que esté dirigido el sistema IOT<sup>[51]</sup>.

	Gratuito	Startups	Empresas
Mensajes máximos al mes	100000	1000000	Según contrato personalizado
Tiempo histórico en días	30	30	Según contrato personalizado
Número dispositivos	Ilimitado	Ilimitado	Según contrato personalizado
Precio/Mes	0€	13,38€	Según contrato personalizado

*Tabla 2. Tarifas plataforma Samsung Artik Cloud*

La plataforma aREST Framework se emplea con los microprocesadores Raspberry Pi y BeagleBone, y con los microcontroladores Arduino y ESP8266. Si se emplea uno de los microcontroladores anteriores, se utilizan las librerías propias de Arduino ya que el protocolo de comunicación de dicha plataforma, REST, es la que posee la librería del Arduino<sup>[52]</sup>. Esta plataforma posee 3 tipos de contrato y la elección de uno u otro depende de la función que realice el sistema IOT<sup>[53]</sup>.

	Gratuito	Aficionado	Desarrollador
Mensajes máximos al día	100	10000	100000
Tiempo histórico en días	No	No	No
Número dispositivos	5	25	100
Precio/Mes	0€	9€	19€

*Tabla 3. Tarifas plataforma aREST*



Otra plataforma que permite el intercambio de información con la nube es Thinger.io. Esta plataforma, además de ubicarse en la red, se puede instalar en un microprocesador para obtener los datos de forma local, es decir, en la misma máquina. Thinger.io permite la conexión de la Raspberry Pi, del BeagleBone, del Arduino y del ESP8266 y puede transmitir datos mediante el protocolo IFTTT a servidores en la nube como Google Drive, Dropbox, Facebook, Twitter, etcétera. Si se emplean los microcontroladores Arduino y/o ESP8266 se dispone de una librería oficial que permite la conexión entre los dispositivos y la plataforma<sup>[54]</sup>. En la siguiente tabla, se muestran las diferentes tarifas disponibles<sup>[55]</sup>.

	Gratuito	Aficionado	Empresas	Bajo demanda
Mensajes máximos al día	No	No	No	Según contrato personalizado
Tiempo histórico en años	1	2	3	Según contrato personalizado
Número dispositivos	2	20	100	Según contrato personalizado
Precio/Mes	0€	3,95€	19,95€	Según contrato personalizado

Tabla 4. Tarifas plataforma Thinger.io

La plataforma Arduino Cloud, únicamente es válida para los microcontroladores Arduino y ESP8266 y la librería que se emplea para la transmisión de información es oficial. Esta plataforma posee unas limitaciones con respecto a las demás. Dichas limitaciones se basan en la ausencia para consultar los datos desde otras aplicaciones, en no poseer un histórico y en no representar los datos en tiempo real. Este tipo de plataforma solo está en modo gratuito.

ThingSpeak se caracteriza por ser una plataforma de la empresa MathWorks que permite el la visualización de la información recibida por los dispositivos y el envío de dicha información a otras APIs. Esta plataforma se puede emplear con la totalidad de microprocesadores y microcontroladores descritos. Además de consultarse lo datos en internet, se puede instalar la aplicación en el propio ordenador para analizarlos de forma local<sup>[56]</sup>. A continuación, se muestran las diferentes tarifas que posee. El precio de la opción comercial depende del intervalo de tiempo en que se adquieren los datos. Para el precio mostrado en la siguiente tabla, se ha escogido 1 segundo<sup>[57]</sup>.



	Gratuito	Estándar
Mensajes máximos al día	8200	90000
Tiempo histórico en años	1	No
Número dispositivos	3	50
Precio/Mes	0€	579,76€

Tabla 5. Tarifas plataforma ThingSpeak

#### 2.5.4 APIs

Las APIs son los protocolos que emplean los programas que realiza el desarrollador para conectarles a las plataformas que se ubican en internet y así, garantizar el correcto intercambio de información entre ellos<sup>[48]</sup>.

El protocolo MQTT emplea el modo de conexión TCP/IP. Este protocolo es bastante ligero para el envío de mensajes y se basa en el modelo cliente – servidor. Se puede emplear tanto en la Raspberry Pi y en el BeagleBone, como en el Arduino y en el ESP8266. Los datos que se reciben, se pueden enviar a otras plataformas como Twitter o Facebook mediante este protocolo<sup>[58]</sup>. Existen tres subtipos de protocolos, QoS0, QoS1 y QoS2.

El subtipo QoS0, at most once, se caracteriza por ser el más rápido de los tres pero el menos fiable ya que se envían los datos pero no se recibe información acerca del correcto envío de ellos. El siguiente subtipo, QoS1, at least once, envía el mensaje y recibe confirmación de su recepción. El inconveniente de este subtipo radica en que pueden recibirse confirmaciones duplicadas. El último subtipo, QoS2, exactly once, es el más fiable de los tres subtipos descritos ya que la confirmación de que el mensaje ha llegado a su destino se recibe una sola vez, por lo que se consume un mayor ancho de banda.

El protocolo HTTP y REST se utiliza en bastantes aplicaciones web ya que es fácil de desarrollar y se admite en bastantes plataformas mediante JSON, JavaScript Object Notation, que permite el intercambio de información. Este protocolo se basa



en el modelo cliente – servidor y cualquier información que remita el cliente al servidor o viceversa se realiza de forma completa por lo que no se almacena ningún estado. Las acciones que realiza este protocolo son: GET, leer, PUT, editar, POST, crear, y DELETE, eliminar. El intercambio de información entre el cliente y el servidor o viceversa se realiza mediante un identificador denominado URI, por lo que las operaciones descritas anteriormente se realizan únicamente en el proceso que posee dicha identificación.

En último caso, se analiza el protocolo CoAP. Este protocolo se caracteriza por implementar el modelo HTTP a dispositivos y redes restrictivas. Se basa en el modelo cliente – servidor con un protocolo no orientado a la conexión, UDT. Las acciones que se pueden efectuar con CoAP son las mismas que se pueden realizar con el protocolo HTTP y REST, GET, PUT, POST y DELETE. Además, posee la propiedad observe. Dicha propiedad se fundamenta en modificar los datos del cliente a petición del servidor. El protocolo CoAP posee los mismos subtipos que el protocolo MQTT, por lo que si el sistema lo requiere, se puede solicitar o no la confirmación de que los datos han llegado de forma correcta.

#### 2.5.5 CLOUD COMPUTING

La computación en la nube o cloud computing, almacena en la nube tanto programas como archivos para agilizar tanto las acciones empresariales como las domésticas. De esta forma, se accede a los archivos desde cualquier lugar y se ejecutan programas sin necesidad de instalarlos en el ordenador. El software se compila en un servidor conectado a internet<sup>[59]</sup>.

Un servidor para ser considerado Cloud computing, según el Instituto Nacional de estándares y tecnología, (National Institute of Standards and Technology, NIST)<sup>[60]</sup> debe poseer las propiedades de autoservicio a demanda, acceso amplio a la red, pool de recursos, elasticidad y escalabilidad, y por último, servicio medible. El autoservicio a demanda se basa en el acceso de los usuarios al servidor sin necesidad de contactar con el proveedor del servicio. El acceso a la red se realiza por protocolos estándares. Este hecho permite que los usuarios accedan a sus datos o programas sin la necesidad de poseer un protocolo específico. Los proveedores poseen un grupo de servidores que proporcionan a los usuarios tanto la capacidad de almacenamiento como la velocidad de procesamiento que estos requieran, pool de recursos. La propiedad denominada elasticidad y escalabilidad se caracteriza por adaptar un servidor a las necesidades de los usuarios por lo que habilitan o deshabilitan recursos de forma automática sin que estos tengan que intervenir. Las acciones que ejecuten los usuarios en el servidor se almacenan en un registro. Debido a ello, en el caso de que ocurra algún problema, se garantiza la transparencia para el proveedor y para los usuarios.



Existen diferentes tipos de modalidades de computación en la nube, el software como servicio, SaaS, la plataforma como servicio, PaaS, la infraestructura como servicio (IaaS), la nube pública, la nube privada y la nube híbrida. El software como servicio, SaaS, se caracteriza por conectarse el usuario, en la mayoría de los casos a través del navegador web, a un programa que se encuentra compilado en la nube. La plataforma como servicio, PaaS, permite desarrollar y distribuir software fundamentado en la nube. La infraestructura como servicio, IaaS, mejora la funcionalidad de las empresas ya que permite que cualquier persona que requiera cierta información y tenga permiso para visualizarla, acceda a ellos sin necesidad de solicitarlos. La nube pública se emplea como almacenamiento de ficheros tanto en el ámbito empresarial como en el ámbito doméstico. En este caso, un grupo de empresas o de usuarios utilizan el mismo servidor pero únicamente tienen acceso a los archivos que ellos han creado. La nube privada consiste en un servidor para una única empresa u usuario, por lo que no tiene punto de acceso público. El funcionamiento de la nube híbrida es idéntico al de la nube privada con la diferencia de que este último dispone de un punto de acceso público.

La computación en la nube otorga a una empresa suprimir los costes de adquisición, en el caso de que no se tenga un servidor, y de mantenimiento, costes de energía o personal. Si la empresa contrata un servidor en la nube, estos costes los asume el proveedor del servidor, y además, en el caso de avería, el arreglo se realizaría en un tiempo menor ya que tienen las 24h un servicio de mantenimiento.

La seguridad en los servidores que almacenan archivos o programas posee una elevada importancia. Dicha seguridad es mayor si se emplea un servidor en la nube que un servidor propio ya que la experiencia del personal que trabaja para el proveedor del servicio es superior a los que trabajan en el mantenimiento de una empresa, cuya función no se dirige a este tipo de servicio, ya que están más acostumbrados a ataques maliciosos de hardware y de software.





# CAPÍTULO 3

## DISEÑO E IMPLEMENTACIÓN DEL SISTEMA





### 3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

El desarrollo del sistema inteligente de gestión de tráfico puede emplear diferentes tipos de sensores en la adquisición del flujo vehicular y en la detección de peatones por lo que en este apartado se realiza un estudio de los dispositivos a emplear. Además, se explica el funcionamiento general del sistema y las acciones que realizan los programas que lo constituyen.

#### 3.1 ARQUITECTURA DEL SISTEMA

En el apartado 2.3 se especificaron los dispositivos que se podían emplear para la adquisición de las características del tráfico, la detección de personas, el control de los discos de los semáforos de una intersección y los medios en los que se puede transferir la información entre los aparatos. En los siguientes apartados, se realiza un análisis de ellos para seleccionar los que otorguen mayor rentabilidad al proyecto y el lugar donde se deben situar.

##### 3.1.1 SELECCIÓN DE DISPOSITIVOS

El flujo vehicular y la presencia de peatones se pueden adquirir mediante sensores fotoeléctricos o electromagnéticos o a partir de cámaras de videovigilancia.

Los sensores fotoeléctricos detectan el paso de un vehículo y la presencia de personas. La detección se produce en el momento en que un vehículo o persona pasa por ellos y corta el haz de luz que emiten. Los sensores electromagnéticos detectan los vehículos a partir de la interacción de su campo magnético con el metal que contienen. Las cámaras de videovigilancia detectan tanto vehículos como personas mediante un programa instalado en un microprocesador que recibe la secuencia de imágenes.

El sistema a diseñar debe ser lo más automático posible. Debido a ello, los dispositivos que se utilicen deben ser capaces de adquirir los vehículos que circulan en la misma dirección y por diferentes carriles en caso de que haya más de uno, el tipo de vehículo que entra y sale del área de detección y la espera o no de un peatón antes de cruzar la carretera.

En el caso de que se utilizasen sensores fotoeléctricos, estos se ubicarían en los extremos de la carretera ya que si se colocan en su interior provocarían colisiones de los vehículos con los aparatos. Debido a ello, si la carretera es de doble sentido estos dispositivos adquirirían las características del tráfico tanto de los vehículos que circulan en una dirección como de los vehículos que circulan en dirección contraria.



La carretera de acceso a la intersección puede tener más de un carril por lo que si dos vehículos pasasen a la vez por los sensores fotoeléctricos, estos solo reconocerían un vehículo.

Los peatones pueden pasar por el cruce de peatones sin cruzar la calzada por lo que este tipo de tecnología no serviría para la detección de personas ya que una vez detectada, el sistema sería incapaz de conocer si la persona está o no esperando a que el semáforo de peatones le permita el paso.

Los problemas descritos para el empleo de sensores fotoeléctricos hacen imposible el desarrollo del sistema con esta tecnología, por lo que se descarta su implementación.

La detección de vehículos por sensores electromagnéticos se realiza a través de un cable enterrado entre 2 y 3 centímetros en el asfalto. Si el acceso a la intersección posee más de 1 carril, cada uno de ellos lleva un cable enterrado y un sensor electromagnético ubicado en el armario de regulador semafórico. El asfalto de las carreteras necesita cada cierto tiempo un mantenimiento. En el fresado se destruye el cable por lo que habría que reemplazarlo. Este tipo de sensores no se emplean en la detección de peatones ya que si estos no poseen elementos metálicos, dicho sensor no los reconoce.

El hecho de que se instalen varios sensores si el acceso posee más de un carril y que si se hace un mantenimiento habría que realizar de nuevo la instalación del cable produciría un incremento tanto del coste de instalación como del coste de mantenimiento por lo que se desestima el empleo de sensores electromagnéticos.

Las cámaras de videovigilancia son capaces de detectar mediante un programa informático, la densidad del tráfico independientemente del número de carriles que constituya el acceso a la intersección, obtener de manera automática el área de control y detectar la presencia de peatones. La utilización de cámaras de videovigilancia no requiere mantenimiento y no las afecta las tareas de conservación del asfalto, por lo que el coste de instalación es inferior con respecto a la utilización de los sensores anteriores. Debido a ello, se selecciona el empleo de cámaras de videovigilancia para la adquisición de las características del tráfico y para la detección de los peatones.

En el apartado 2.3.2 se expusieron diferentes cámaras de videovigilancia que se podían emplear en la implantación de este proyecto. Tanto para la detección de vehículos como para la detección de personas la cámara debe tener el mayor ángulo de visión y el mayor alcance. La cámara más rentable y que posee las características anteriores es la Cámara IP Dahua X-Security con un ángulo de visión de 100°, un alcance nocturno de 30 metros y con posibilidad de enviar vídeo por señal WiFi.



El sistema inteligente de gestión de tráfico posee un microprocesador capaz de analizar los datos recibidos por las cámaras de videovigilancia y de controlar las secuencias de los semáforos. Este microprocesador debe poseer una memoria interna con un tamaño que permita el almacenaje del programa, una memoria RAM que tenga capacidad suficiente como para ejecutar los procesos en tiempo real, una velocidad de procesador que no ralentice la ejecución de los programas que controlan el sistema y un módulo WiFi que permita tanto la recepción de información por parte de las cámaras como el envío de las señales de control de los semáforos.

Los microcontroladores Arduinos descritos en el apartado 2.3.3 no poseen las características expuestas ya que el tratamiento de imágenes conlleva un coste computacional y una memoria RAM superior al que otorgan estos dispositivos, por lo que se excluye su instalación en el sistema.

Los microprocesadores Raspberry Pi expuestos en el mismo apartado que los microcontroladores Arduinos, se caracterizan por poseer una memoria flash y una memoria RAM elevada con lo que los programas de control se podrían introducir y ejecutar en tiempo real. Además, el procesador posee una velocidad que permitiría tanto el tratamiento de las secuencias de imágenes como el control de los semáforos sin que se produjesen retardos. Estos dispositivos poseen un módulo WiFi con el que transmitir información entre dispositivos.

El microprocesador más viable y rentable para control del tráfico es el Raspberry Pi 3 Model B+. Este dispositivo consta de 4 núcleos a 1,4 GHz y una memoria RAM de 1 GB capaz de controlar las secuencias de los semáforos en tiempo real. Además posee conexión WiFi por lo que no sería necesario la instalación de cables entre los dispositivos que intercambien información.

Los datos se transmiten entre los dispositivos mediante cables o módulos WiFi. La utilización de módulos WiFi es más rentable que el coste del material y la instalación de cables, por lo que se emplean cámaras con conexión WiFi y módulos WiFi capaces de recibir del microprocesador la luz del semáforo que debe estar activa. Los módulos WiFi contienen salidas analógicas por lo que a partir de tres de ellas, se controla el encendido de la luz roja, verde y ámbar. El coste computacional del control del semáforo es bajo por lo que no se requiere velocidad del microcontrolador. El módulo WiFi que posee las características mencionadas y otorga rentabilidad al proyecto es el ESP8266EX.

El sistema inteligente de gestión de tráfico se conectará a internet para poder observar la fase activa de una intersección desde un centro de control. Además, el operario podrá modificar la fase activa si fuese necesario. Debido a ello, cada intersección se conecta a internet a través de la red. El dispositivo seleccionado para ello es el módulo GSM 2 ya que únicamente se envían datos por lo que no se necesita



una velocidad de transferencia elevada. Este aparato es válido para el microprocesador el Raspberry Pi 3 Model B+.

Como conclusión de este apartado, los dispositivos a emplear en este sistema son: cámaras de videovigilancia IP Dahua X-Security, microprocesador Raspberry Pi 3 Model B+ y módulos WiFi 8266EX y módulo GSM 2.

### 3.1.2 UBICACIÓN DE DISPOSITIVOS

La ubicación y el posicionamiento de las cámaras de seguridad es de vital importancia tanto para la puesta en marcha como para el funcionamiento continuo del sistema ya que si se colocan de forma incorrecta, los datos adquiridos por las cámaras podrían ser erróneos.

Cada acceso a la intersección se compone de tres cámaras, una enfocando a los vehículos y las otras dos enfocando cada una a cada lado del paso de peatones.

La secuencia de imágenes de los vehículos se capta desde una cámara situada en el báculo del semáforo lo más próximo posible a la mitad del carril o de los carriles, si el acceso contiene más de uno. La cámara se debe posicionar de tal forma que en la parte inferior de la imagen se obtenga la línea de detección.

Las secuencias de imágenes de los peatones se captan desde dos cámaras situadas en la parte superior de cada columna del semáforo que se ubican en el lateral de la calzada. El posicionamiento de las cámaras depende del lado que este enfocando, es decir, si se realiza el enfoque de la cámara izquierda, esta se posiciona de tal forma que se vea el rectángulo más próximo a la acera en la parte derecha de la imagen, mientras que si se efectúa el enfoque de la cámara derecha, esta se posiciona de tal forma que se vea el rectángulo en la parte izquierda de la imagen.

El microprocesador Raspberry Pi 3 Model B+ se ubica en la caja del regulador semafórico y los módulos WiFi, se posicionan en cada cara del semáforo para así controlar el estado de las luces. Si el semáforo posee semáforos de repetición solo se conecta uno por módulo WiFi ya que la información que recibe es idéntica para todos ellos.

## 3.2 FUNCIONAMIENTO GENERAL DEL SISTEMA

El sistema inteligente de gestión de tráfico se divide en dos fases, la puesta en marcha y la ejecución continua. En la puesta en marcha, el programa obtiene los tiempos iniciales de cada fase y adquiere la región de la que se obtendrán las



características tanto del tráfico como de peatones, y en la ejecución continua se adquirirá el flujo de vehículos y la detección de personas.

Las secuencias de imágenes obtenidas de cada cámara se tratan en un programa diferente, por lo que si el acceso a una intersección se produce por cuatro vías diferentes, el número de procesos que adquieren las características del tráfico y de los peatones se tasan en doce, cuatro cámaras para vehículos, enfocando cada una a su vía de acceso, y ocho cámaras para peatones, enfocando cada una a un lado de la calzada.

La información obtenida por las cámaras anteriores se envía a un programa que interpreta los datos y a partir de él, se controla el paso de vehículos a la intersección y el cruce de peatones. La fase que está activa en cada momento se envía a otro programa vía red 4G para que desde una sala de control poder modificar, si se desea, la fase activa. El flujo de vehículos, la presencia o no de peatones y la fase activa, se remite a una plataforma web para su análisis. El número de programas que se ejecutan en total para una intersección con cuatro accesos se evalúa en quince, doce para la adquisición de las secuencias de imágenes de cada cámara, uno para la interpretación de los datos de las cámaras y el control de los semáforos, otro para controlar el acceso a la intersección de forma manual, y el último, para enviar las características de la intersección referentes al flujo de vehículos, a la presencia de peatones y la fase activa a un servidor web.

### 3.2.1 PUESTA EN MARCHA

En el primer arranque del sistema, el programa solicita información acerca de las características tanto del tráfico como de la intersección. Las características del tráfico se emplean para el cálculo del tiempo inicial de las diferentes fases que tiene la intersección. Para dicho cálculo, el programa solicita el número de fases, divisiones de acceso de los vehículos a la intersección, y la numeración de carriles que pertenecen a las distintas fases. Esta numeración no debe contener un cero. Si se introdujese dicho valor por error, el programa le volvería a solicitar el número de movimiento. La fase principal es la número uno. A continuación se introducen las características de cada movimiento ya que, un vehículo puede seguir recto o si la intersección lo permite realizar un cambio de dirección. En el caso de que se produzca un cambio de dirección se debe introducir el número de carril del que se produce la desviación. Por último, el sistema solicita para cada movimiento, la intensidad de vehículos, la cantidad de carriles y la longitud del paso de cebra. La intensidad de vehículos que acceden por cada vía a una intersección es un parámetro que un operario ha analizado con anterioridad.



A partir de los datos introducidos, el programa calcula de forma automática los tiempos iniciales óptimos y máximos de cada fase. En el caso de que el tiempo de ciclo sea inferior a 35 segundos o superior a 120 segundos, el sistema ajusta los tiempos de cada fase para que el tiempo de ciclo sea de 35 segundos si el obtenido es menor y de 120 segundos si es mayor. Este hecho solo ocurre en el cálculo de tiempo inicial óptimo. Si el tiempo en rojo del semáforo es inferior al tiempo que tardan los peatones en cruzar, se modifica el tiempo de ciclo para aumentar el tiempo en rojo de la fase o fases que no cumplan la condición descrita.

A continuación, se ejecutan automáticamente los programas que tratan la secuencia de imágenes de las cámaras. El tratamiento de las imágenes de los vehículos que acceden a la intersección por distinto carril se realiza con el mismo programa pero con diferente identificador. Lo mismo ocurre para el análisis de la secuencia de imágenes de los peatones, se emplea el mismo programa pero se identifica de forma diferente. La identificación de los procesos correspondientes a las cámaras de los peatones, se establece a partir del identificador asignado al semáforo y a la posición de la cámara, es decir, si está a la derecha o a la izquierda de la calzada.

En el instante en que finaliza la identificación de los procesos, se obtiene la región de interés tanto de los vehículos como de los peatones. En el caso de los vehículos, la región que se desea analizar es el polígono que forma la línea de detención con las líneas de delimitación de carril y con la línea que une los extremos de las líneas de delimitación.

La región de interés de los peatones se obtiene a partir del rectángulo del paso de cebra que está más cerca de la acera. Una vez obtenida dicha figura, se adquiere la línea vertical que se ubica más al exterior y se implementa un polígono que esté a una distancia de 0,5 metros con respecto a la recta anterior. La primera recta que forma el polígono es paralela a la recta exterior del rectángulo del paso de cebra. Otras dos rectas horizontales surgen de los extremos de la recta anterior con una distancia de 1,5 metros y la recta que cierra el polígono es la que une los extremos de las rectas anteriores.

El grado de ocupación vehicular y la detección de personas se obtienen a partir de la densidad de ocupación, por lo que en la puesta en marcha se adquiere el área total de las distintas regiones de interés. El área de análisis correspondiente a la carretera posee perspectiva por lo que se establece una matriz que otorga diferentes pesos a los píxeles de tal forma que un vehículo que se encuentre en la zona de detención posea la misma área que un vehículo que acaba de cruzar la línea de detección. En el caso de los peatones no es necesario realizar esta corrección ya que el tiempo que tardan en cruzar la calzada se evalúa en función de la longitud de la calzada y la velocidad media de los peatones, 1,5 m/s. El área total es el sumatorio



de los pesos asignados a los píxeles en el caso de los vehículos y el sumatorio de los píxeles que conforman la región de interés en el caso de los peatones.

El área total obtenida por cada proceso que controla cada cámara de videovigilancia se envía al programa de control para que éste, junto con el área ocupada, evalúe los tiempos de fase y ciclo para el flujo actual de vehículos y de peatones.

### 3.2.2 EJECUCIÓN CONTINUA

Una vez que la puesta en marcha del sistema inteligente de gestión de tráfico ha obtenido el tiempo de ciclo óptimo para la intersección y ha evaluado las áreas iniciales, se adquiere el área ocupada de cada región de interés. Éste área, en el caso de los vehículos, es la sumatoria de los píxeles que se ubican en la matriz de corrección de perspectiva y cuyo valor en la matriz que contiene los objetos en movimiento posee el valor 255, es decir, el color blanco, y en el caso de las personas, es el sumatorio de los píxeles que se obtienen mediante el método de sustracción de fondo. Las áreas ocupadas se envían al programa de control que junto con el área total obtiene la densidad de ocupación.

En el caso de los vehículos, la densidad de ocupación obtenida se emplea para conocer la cantidad de vehículos que se encuentran en la región de interés y el tiempo que tardarían en pasar y la densidad de ocupación de los peatones se usa como un detector. Para este último caso si el número de píxeles que se obtienen con el método de sustracción de fondo supera el 2% del área total, margen que se estipula para la eliminación del ruido residual, el programa adapta los tiempos de fase calculados según el flujo de vehículos para que los peatones puedan y tengan tiempo suficiente para cruzar la calzada. La densidad de ocupación se calcula mediante la siguiente expresión.

$$\rho_{ocupación} = \frac{A_{ocupada}}{A_{total}}$$

*Ecuación 12. Densidad ocupación*

El programa de control adquiere el flujo de vehículos y de peatones en tiempo real por lo que puede modificar la secuencia de los semáforos según los datos que reciba de los procesos que tratan las imágenes captadas por las cámaras de videovigilancia.

En el momento en que se inicia la ejecución continua, la fase activa es la introducida por el operario como fase 1. Las intersecciones tienen un acceso de preferencia que en el programa corresponde con la fase mencionada. Este acceso



se caracteriza por poseer la mayor intensidad de vehículos. A continuación, el programa de control analiza los datos de tráfico y de peatones y si cumple las condiciones para el cambio de fase, este lo ejecuta.

Los factores que determinan el cambio de fase son el tiempo, el flujo de vehículos y la presencia de peatones. En el instante en que se produce el cambio de secuencia de los semáforos, se calcula el tiempo óptimo de cada fase según las características del tráfico y la presencia de peatones. Una vez obtenido dicho tiempo, se pone en funcionamiento un cronometro para que no se supere el tiempo máximo asignado a cada fase. A continuación, se establecen las condiciones que se deben producir para que se modifique la fase activa sin haber alcanzado el cronometro el tiempo óptimo o máximo.

En el caso de que la totalidad de vehículos que estuviesen detenidos atravesasen la intersección en un tiempo inferior al óptimo y no hubiese más en el área de análisis, la fase actual se desactivaría y la siguiente fase con vehículos se activaría. Si se alcanza el tiempo óptimo y la densidad de ocupación es superior al 10%, se calcula de nuevo el tiempo óptimo según el flujo de vehículos y la fase actual se mantiene activa. El tiempo en que una fase está activa nunca puede sobrepasar el tiempo máximo establecido al principio del ciclo. Se produce un cambio de ciclo si se alcanza el tiempo óptimo y la densidad de ocupación es inferior al 10% ya que, en este caso, los vehículos que estaban detenidos ya han pasado y los que detecta el sistema son vehículos que entran en el área de análisis.

La detección de peatones es tan importante como el análisis del flujo vehicular ya que hay que garantizar que tengan tiempo suficiente para cruzar la calzada y que el tiempo de espera sea limitado. Si únicamente se tuviera en cuenta el flujo de vehículos y solo habría movimiento por un acceso a la intersección, la espera de los peatones sería ilimitada ya que dicho semáforo estaría siempre permitiendo el paso de los vehículos. Para que no se produzca este problema se analiza también el área de peatones por lo que en el momento en que se detectan, se modifican los tiempos de fase óptimos para que puedan cruzar la calzada. Este hecho origina que un semáforo se pueda poner en verde sin que haya vehículos para permitir el paso de peatones por otros cruces de la intersección.

En algunos casos, la zona de espera para cruzar la calzada se ubica en la misma acera por lo que el sistema detectaría a los peatones. También, se puede dar el caso de que un peatón pase de largo por dicha zona sin detenerse. En el momento en que el sistema detecta a un peatón, se modifica el ciclo para que tenga tiempo suficiente para cruzar. Si el peatón no se detiene en el área de análisis, el sistema vuelve a calcular los tiempos de fase para que estos vuelvan a ser óptimos sin tener en cuenta a los peatones que no se han detenido.



En determinados momentos del día, en una intersección, puede que no haya ningún vehículo ni ningún peatón. En este caso, se activa la fase principal, es decir, la número 1. Si se detecta algún vehículo o peatón en esa misma fase o en otra se vuelven a calcular los tiempos de cada fase. Si se detecta un vehículo en otra fase, el sistema además de calcular los tiempos, cambiaría a la fase en la que se ha detectado el automóvil.

Mientras una fase está activa, el sistema no modifica los tiempos a no ser que se detecten peatones. Por ello, si se produce un cambio de fase el sistema obtiene de nuevo la duración de cada fase para adaptarla a los vehículos que han llegado mientras los semáforos estaban en rojo.

El cambio de secuencia de verde a rojo de un semáforo pasa por el encendido del disco amarillo. Este disco informa a los conductores de que el semáforo se pondrá en rojo en un tiempo muy breve, por lo que deben disminuir la velocidad para así detenerse al alcanzar el semáforo. El tiempo en que el semáforo permanece en verde se establece en 3 segundos. Dicho tiempo se mantiene igual para todas las intersecciones ya que de esta forma, los conductores reaccionan siempre de la misma forma y la decisión para empezar a disminuir la velocidad depende únicamente de la distancia a la que se encuentren.

Las áreas ocupadas de los vehículos, la detección de los peatones y la fase activa se remite a una plataforma IOT que permite consultar el histórico desde cualquier dispositivo que tenga acceso a la red.

El sistema inteligente de gestión de tráfico permite tanto la visión como la modificación de la fase activa desde un centro de control a través de internet. En el caso de que un operario modifique la fase, esta no se volverá a cambiar hasta que este vuelva a poner el sistema en modo automático. En dicho instante, el programa de control establece el tiempo para cada fase según el flujo de vehículos y la presencia de peatones. La fase que había accionado el operario permanecerá activa hasta que se cumpla una de las condiciones de cambio de fase.

### 3.3 DIAGRAMA DE FUNCIONAMIENTO DEL SISTEMA

El desarrollo del sistema inteligente de tráfico se realiza con un total de cinco programas. El primer programa se denomina “Sistema inteligente de vehículos”, SIV, y se encarga del tratamiento de las imágenes de los vehículos. El segundo proceso se llama “Sistema inteligente de peatones”, SIP, y se emplea para la detección de los peatones que se ubican en la zona de espera. El tercer programa se nombra “Sistema de control de tráfico”, SCT, y se utiliza para el control de la secuencia de los semáforos. El cuarto proceso se denomina “Centro de control de tráfico”, CCT, y se caracteriza por mostrar y modificar, si el operario lo cree conveniente, la fase activa



de una intersección. El último programa, “Internet of Things”, IOT, recibe las características de la intersección y las envía a la plataforma web.

Los programas SIV y SIP se pueden emplear para cualquier acceso y zona de espera, respectivamente, ya que ambos son capaces de adquirir de forma automática la región de interés. Debido a ello, el número de veces que se ejecutan dichos procesos depende de la cantidad de accesos que contenga la intersección. Estos programas junto con el SCT y el IOT se ejecutan en el microprocesador Raspberry Pi que se ubica en el armario de regulación semafórica mientras que el programa CCT, se ejecuta en el centro de control de tráfico. En la siguiente figura se muestra el diagrama de flujo del funcionamiento general del sistema en la puesta en marcha y en la ejecución continua para una intersección con tres accesos. Para este caso, la intersección tendrá un total de 9 cámaras, 3 para los vehículos y 6 para los peatones, por lo que el número total de procesos serán 11, 3 procesos SIV, 6 procesos SIP, 1 proceso SCT, 1 proceso IOT, y 1 proceso CCT.

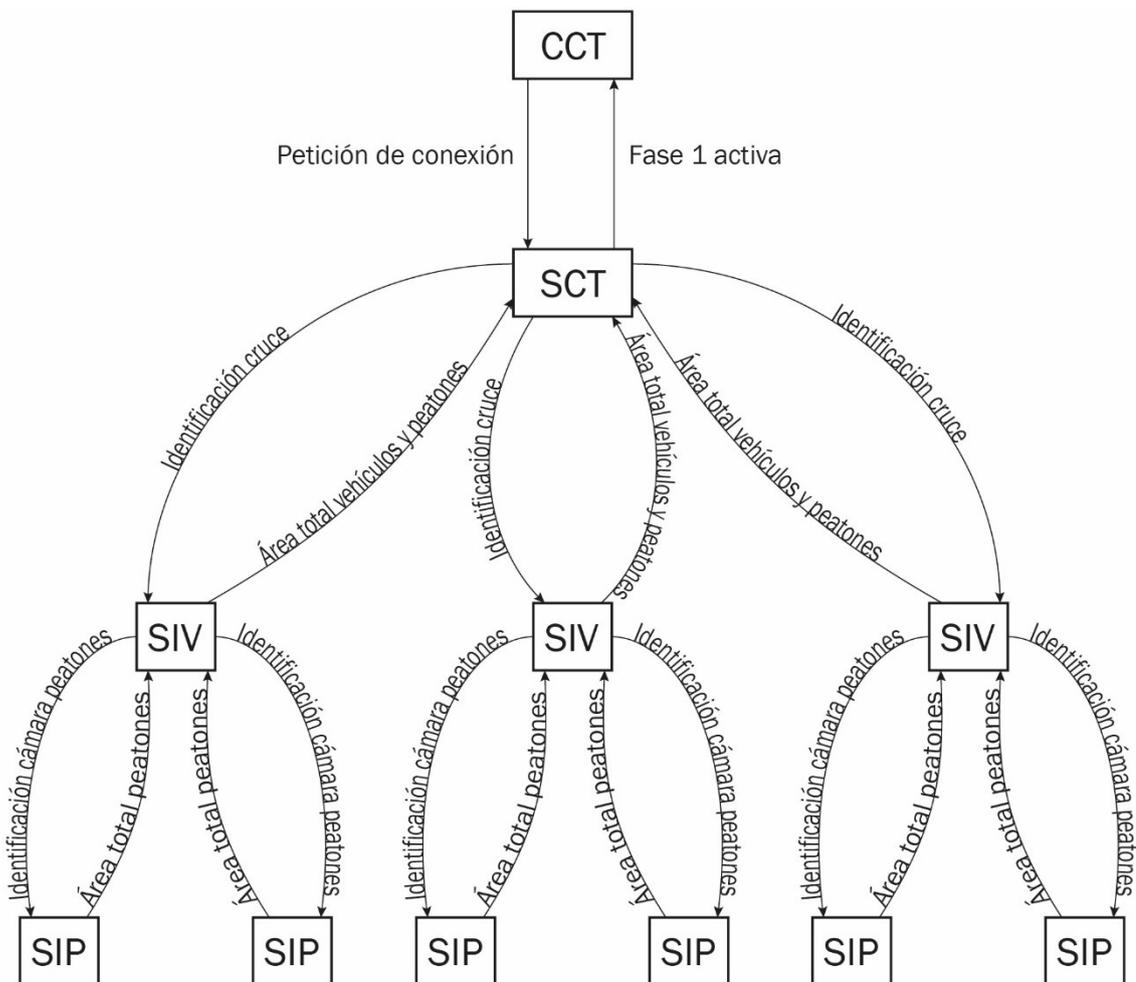


Figura 9. Diagrama puesta en marcha



En la puesta en marcha del sistema, se identifican los procesos que interpretan las cámaras de videovigilancia y se obtiene el área total de la región de interés tanto de los vehículos como de los peatones.

En primer lugar, se realiza la conexión a internet entre el proceso CCT y el proceso SCT. Para ello, el programa CCT envía a la red una solicitud para aceptar las conexiones de los procesos SCT. En el instante en que se establece conexión entre ambos programas, se solicitan los datos referentes al tráfico vehicular y al paso de peatones. A continuación, se produce la apertura de los procesos que tratan las imágenes que reciben de las cámaras de videovigilancia. El programa SCT realiza la apertura de los programas SIV de uno en uno para así, enviarles el número de identificación. Dicho número corresponde al introducido como número de movimiento que solicita el proceso SCT en su inicio. Una vez identificado el proceso SIV, se abre de nuevo otro programa SIV y se le asigna la identificación que le corresponda. Este hecho se repite hasta que se abren todos los procesos que controlan las cámaras que enfocan a los vehículos.

En el momento que se asigna un identificador al programa SIV, este ordena la apertura de dos programas SIP, uno para el análisis de los peatones que se ubican en el lado derecho de la calzada y otro para los que se posicionan en el lado izquierdo. La iniciación de dichos programas se realiza de forma secuencial para así identificar a la cámara que enfoca al lado derecho con el número 0 y a la cámara que enfoca al lado izquierdo con el número 1. Esta acción se repite para cada programa SIV que contenga el sistema inteligente de gestión de tráfico. La apertura de los programas SIP entre distintos procesos SIV no se realiza de forma secuencial ya que su identificador procede del proceso SIV que les crea.

Un vez que se identifican los procesos, se adquiere el área total de cada región de interés que obtienen los programas SIV y SIP. Los procesos SIP envían el área de peatones a través de memoria compartida a cada proceso SIT que le ha abierto. Y estos, envían el área que han recibido de los procesos SIP junto con el área de vehículos al proceso de control SCT mediante memoria compartida. El envío de las áreas totales se produce de idéntica forma para los procesos SIV que ha abierto el programa SCT.

Para que haya sincronización entre los datos que se transmiten y los programas SCT y SIV sepan cuál es el proceso que está enviando o recibiendo información, se emplean semáforos. Los semáforos se caracterizan por permitir el acceso a la lectura o escritura de la memoria compartida de forma secuencial.

El programa de control SCT, en el instante en que recibe el área total de todas las regiones de interés, envía a través de internet la fase activa inicial y comienza con la ejecución continua. Dicha fase, tal y como se ha expuesto anteriormente es la número 1 que corresponde con la principal.

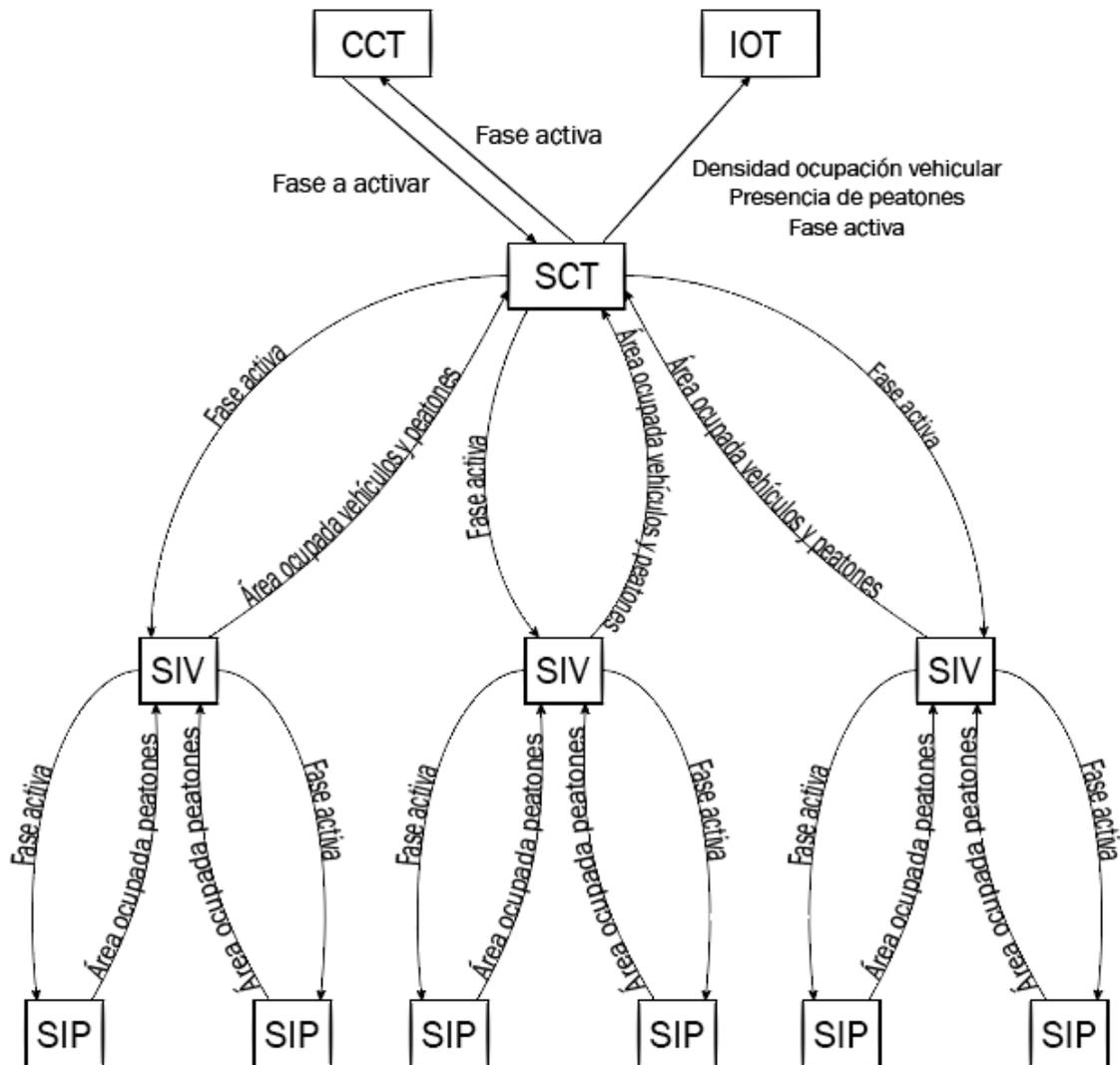


Figura 10. Diagrama ejecución continua

Los datos que se transmiten entre los programas en puesta en marcha son diferentes de los que intercambian en ejecución continua. La información que se envía corresponde a la fase que se encuentra activa y al área ocupada de vehículos y de peatones.

En la ejecución continua el programa de control SCT analiza las áreas enviadas por los procesos que manejan las cámaras para el cálculo del tiempo de fases y para el control de la secuencia de los semáforos.

El programa CCT envía la fase que se encuentra activa o la que se establece por la interacción de un operario al proceso SCT a través de internet. El programa de control, SCT, remite dicha fase a los programas SIV que ha creado y estos a su vez la envían a los procesos SIP que han abierto. El envío de la fase activa de SCT a SIV y a



SIP se realiza mediante memoria interna. Los programas que manejan las cámaras necesitan saber la fase que está activa ya que el método de reconocimiento de vehículos y peatones depende del disco del semáforo que este encendido.

Una vez que los procesos SIP reciben la fase activa, estos evalúan el área ocupada por los peatones y lo remiten al proceso SIV. Los datos de ocupación recogidos por el programa SIV, se remiten junto con el área ocupada por los vehículos al programa de control SCT. El envío de dicha información de SIP a SIV y de SIV a SCT se efectúa mediante memoria compartida. Estas acciones se realizan del mismo modo para los diferentes programas SIV que ha creado el proceso SCT.

La información transmitida entre los programas SCT, SIV y SIP se sincroniza mediante semáforos, igual que ocurre en la puesta en marcha del sistema. Debido a ello, los procesos conocen el remitente de la información y su destinatario. Además, se garantiza que los datos que reciba un proceso sean los que este requiere y no los datos de otros procesos.

Por último, el programa de control SCT analiza la información recibida de los procesos SIV y controla la secuencia de los semáforos que constituye la intersección a partir de dicha información. El control de los semáforos se establece mediante un algoritmo que decide si se mantiene o si se cambia la fase activa. Esta fase, se envía al programa que se ubica en el centro de control de tráfico, CCT y al proceso IOT. El proceso IOT, además de recibir la fase activa, obtiene el área ocupada por los vehículos y la presencia de peatones y remite dichos datos a la aplicación web. A partir de este punto, el tipo de datos que se envían entre los distintos programas que constituyen el sistema es invariante y el envío y la recepción de ellos se produce de forma continua.

El encendido de los diferentes discos de los semáforos que posee una intersección se realiza mediante el módulo WiFi ESP8266EX. En la puesta en marcha del sistema, el proceso SCT asigna a cada módulo un número que corresponde con el número de movimiento principal asignado y pone los semáforos para vehículos en verde que correspondan con la fase 1. En el momento en que se produce un cambio de fase en la ejecución continua, el programa de control SCT envía a los módulos WiFi una señal para que cambie los discos de los semáforos que pertenecen a la fase activa de verde a ámbar y a continuación envía otra señal para que los discos que estén en ámbar pasen a rojo y los semáforos que constituyen la fase a activar pasasen de rojo a verde. Solo se necesita un módulo WiFi por cada acceso a la intersección, ya que si el semáforo para vehículos está en rojo, el semáforo para los peatones estará en verde. Esta operación se realiza mediante un circuito inversor que se conecta a la señal que establece el encendido del disco verde del semáforo que regula el acceso de los vehículos a la intersección. Además, el semáforo de los peatones se pondrá intermitente para así, avisar a los peatones de que en un tiempo



breve dicho semáforo se pondrá en rojo. La duración en intermitente equivale al tiempo en que el disco amarillo está activo.

### 3.4 IMPLEMENTACIÓN DEL SISTEMA

El sistema inteligente de gestión de tráfico se desarrolla, tal y como se ha expuesto anteriormente a partir de 5 programas, CCT, proceso que controla la fase activa desde un centro de control, IOT, programa que envía los datos al servidor web para su análisis, SCT, proceso que analiza los datos recibidos de los procesos que tratan las imágenes, SIV, programa que adquiere el flujo vehicular, y SIP, proceso que detecta los peatones. En este apartado se describe el algoritmo implantado para su puesta en marcha y para su ejecución continua. En la puesta en marcha del sistema adquiere el área total de la región de análisis de los vehículos y de los peatones y en la ejecución continua el área de ocupación de ellos. El área de ocupación lo trata el programa de control, SCT, para el cálculo del tiempo de fase y para el control de la secuencia de los semáforos.

#### 3.4.1 PROGRAMA CCT

El programa CCT muestra al operario la fase que se encuentra activa en una intersección y permite su modificación. Dicho programa se ubica en un centro de control de tráfico por lo que el intercambio de información entre el programa que controla de forma automática la intersección y el proceso CCT se realiza a través de internet.

En la puesta en marcha, el proceso CCT establece una solicitud de conexión mediante sockets a la red y se mantiene a la espera hasta que el proceso SCT realiza la petición de enlace. Para dicho enlace se emplea un protocolo orientado a conexión ya que los programas se ubican en diferentes máquinas. Este hecho conlleva a establecer una IP pública fija y un puerto de acceso que no utilice el sistema operativo.

La ejecución del programa no se debe interrumpir si el sistema controla el tráfico de forma automática ya que este no funcionaría en tiempo real. Este hecho se produce por la pérdida de tiempo en que el operario introduce por teclado si desea o no modificar la fase. Además, esta acción es inviable si se requiere controlar de forma automática y continua el tráfico. Para que no se produzca este inconveniente, se emplean señales. Una señal se caracteriza por detener la ejecución continua del sistema para realizar una determinada tarea en cualquier momento. En este caso, la tarea consiste en el cambio de fase. La detención de la ejecución del proceso se realiza pulsando las teclas ctrl-c.



En la ejecución continua del programa CCT se muestra la fase activa de la intersección y en el caso de que el operario cambie la fase, se solicitará por pantalla la fase que se desea activar. Además, este indicará si el sistema está funcionando en modo manual o en modo automático para conocimiento del operario.

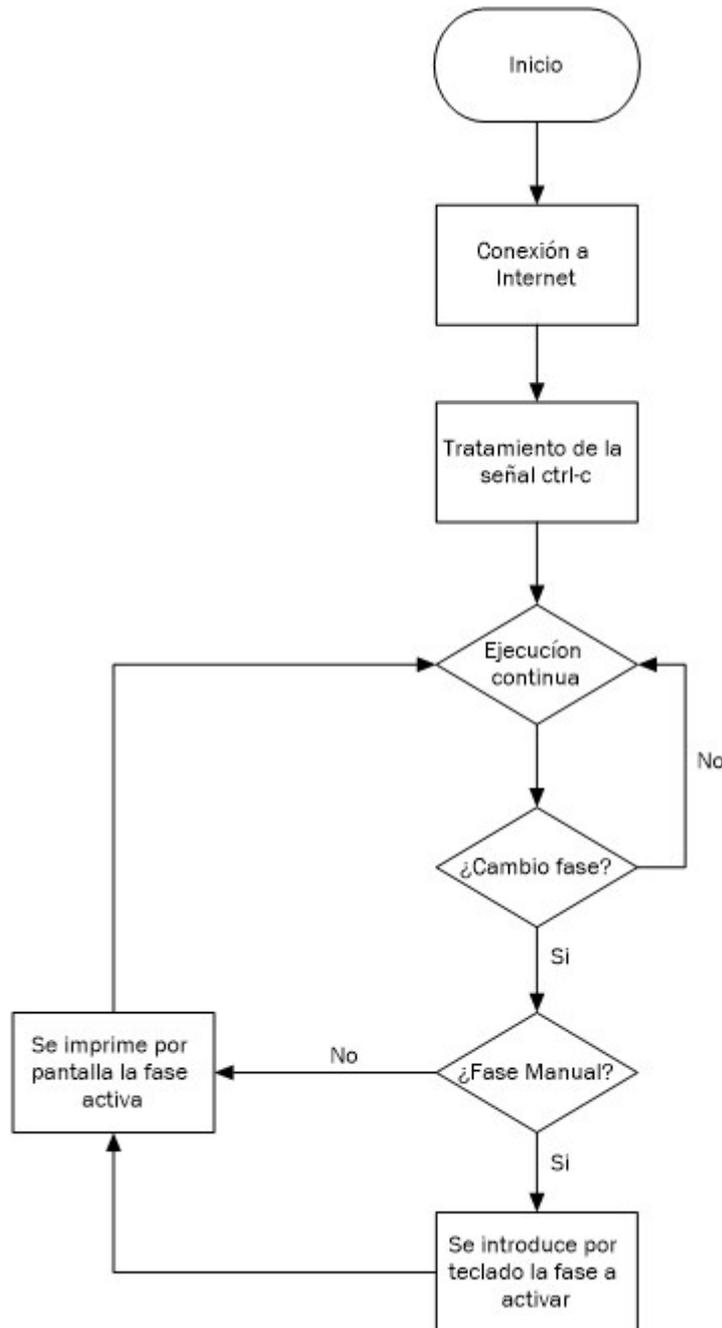


Figura 11. Diagrama flujo programa CCT

Si el sistema o el operario no cambian la fase activa, las características expuestas en la pantalla del operario no se modifican y el programa queda a la espera de recibir de nuevo la fase activa del proceso SCT o de un cambio de fase producido por la persona que se ubica en el centro de control de tráfico. La fase activa se puede



modificar de forma manual o de forma automática. En el caso de que se cambie de forma manual, se introducirá la fase que se desea activar por teclado y está se imprimirá por pantalla. Si la modificación de la fase se realiza de forma automática, se imprimirá directamente por pantalla la fase activa. Tanto si se produce la modificación de forma manual o de forma automática, el programa se queda a la espera de un nuevo cambio de fase.

### 3.4.2 PROGRAMA IOT

La fase activa, la densidad de ocupación de las regiones de interés de los vehículos y la presencia de peatones en sus respectivas áreas de análisis, se envía a una plataforma web para conocer desde cualquier dispositivo y lugar que disponga conexión a internet los datos mencionados. La comunicación entre el programa y la plataforma deben poseer el mismo protocolo de comunicación. Para este sistema se emplea el protocolo HTTP ya que es compatible con la mayoría de dispositivos y de plataformas web. El desarrollo de este programa se efectúa en Python ya que admite el protocolo descrito. El envío de la información y el momento en que debe ejecutarse se establece en una función desarrollada en C++ implementada en el programa SCT.

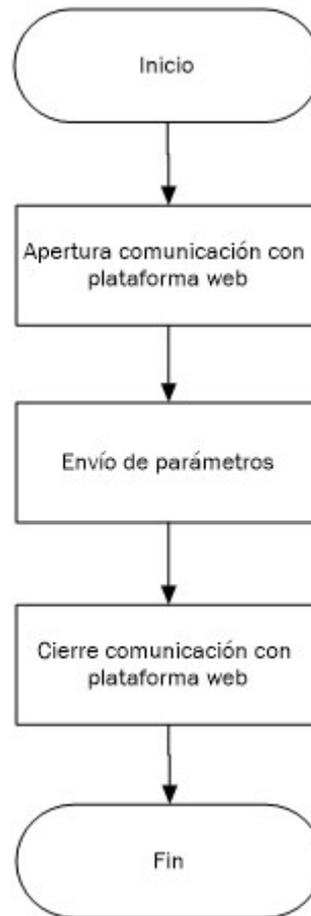


Figura 12. Diagrama flujo programa IOT

El proceso desarrollado en Python posee una única función que recibe los parámetros del programa SCT a través de sus argumentos. Dichos parámetros, se envían a la plataforma ThingSpeak a partir de la dirección de conexión y una contraseña. La contraseña indica la gráfica en la que se deben situar los valores, por lo que si se dispone de un cruce con 4 accesos a la intersección, se abren un total de 5 canales con sus respectivas contraseñas y se remite a un canal la fase activa y a cada uno de los otros 4, la densidad de ocupación de los vehículos y la detección de los peatones de cada acceso a la intersección. Una vez que se remiten los datos, se cierra la comunicación y se vuelve a abrir cuando el proceso SCT vuelve a ordenar su envío a la plataforma web.

### 3.4.3 PROGRAMA SCT

El programa que analiza la información recibida de los procesos que tratan la secuencia de imágenes captadas por las cámaras de videovigilancia y controla la secuencia de los semáforos que constituye una intersección se denomina SCT,



sistema de control de tráfico. El funcionamiento general de dicho programa, se muestra en el siguiente diagrama de flujo.

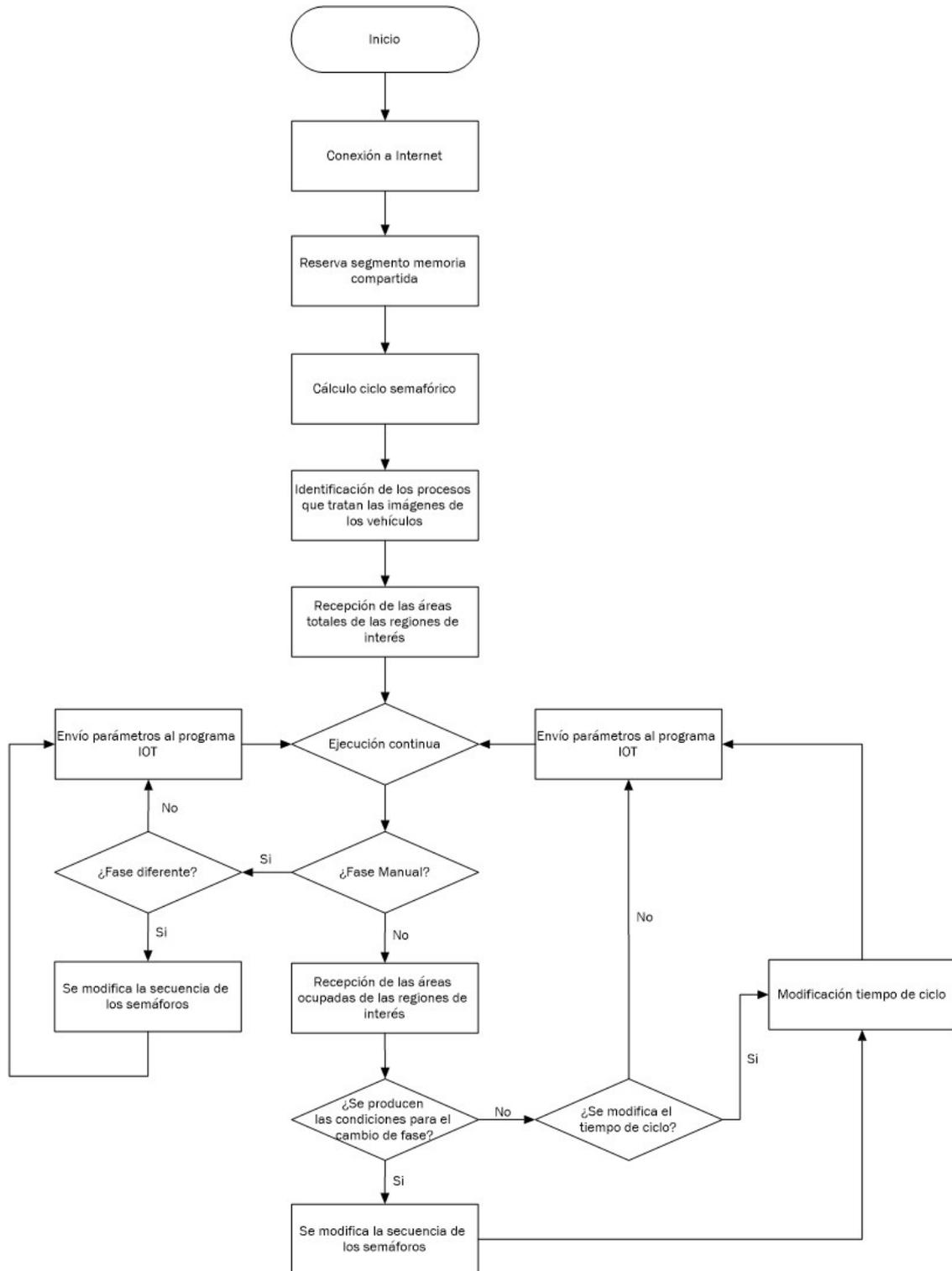


Figura 13. Diagrama flujo programa SCT



En la puesta en marcha del programa SCT, en primer lugar se realiza la conexión con el proceso CCT. Estos programas no se ubican en la misma máquina por lo que se emplea un protocolo orientado a la conexión. Para conectar los programas entre sí, hay que establecer en el proceso de estudio, la IP pública fija y el puerto de acceso que se han asignado al programa CCT.

El proceso SCT intercambia datos con los programas SIV que abre de forma automática. La transmisión de información se produce mediante memoria compartida ya que es la comunicación más rápida entre procesos, por lo que se realiza una reserva de un segmento de memoria con un tamaño suficiente como para que quepan los datos. El acceso a los datos de la memoria compartida se realiza mediante punteros ya que estos apuntan a la dirección de memoria.

A continuación, se adquieren los tiempos de las distintas fases que forman el ciclo semafórico. Para ello, el programa solicita en su puesta en marcha el número de fases, el número que corresponde a cada movimiento de la intersección, si el vehículo se desvía hacia otra dirección y de qué número de movimiento procede, y si el vehículo realiza un giro en la incorporación a la vía de salida. Una vez establecidas las características de la intersección, se introducen los datos del flujo vehicular, intensidad de vehículos y cantidad de carriles de acceso, y longitud del paso de peatones.

A partir de los datos introducidos, se obtiene el ciclo semafórico. Si dichos datos otorgan un ciclo inferior a 35 segundos o superior a 120 segundos, el sistema ajusta de forma automática los tiempos de fase para adecuarse a los tiempos de ciclo mínimo o máximo a los que se acomoda mentalmente el usuario de la vía pública.

Los tiempos obtenidos para cada fase hacen referencia únicamente al flujo vehicular por lo que hay que analizar si la duración de los semáforos en rojo es suficiente para que los peatones dispongan de suficiente tiempo como para cruzar la calzada. La velocidad de los peatones se establece entre 1,1 y 1,5 m/s. La elección de la velocidad entre el rango dado depende de las ciudades. Para este proyecto se ha empleado una velocidad de 1,1 m/s. De esta forma, todas las personas tendrán tiempo para cruzar la calzada sin que existan accidentes.

En el caso de que se modifique la duración de las fases por falta de tiempo para cruzar la calzada, se obtiene un nuevo tiempo de ciclo. Este tiempo es la suma del tiempo de ciclo actual más el incremento entre el tiempo que tardan los peatones en cruzar y el tiempo del semáforo en rojo de la fase que no garantiza el tiempo para el paso de los peatones. Una vez obtenido el nuevo tiempo de ciclo, se calcula el flujo crítico para obtener la nueva duración en verde de las otras fases. El tiempo en verde de la fase a modificar no cambia ya que el flujo de vehículos no varía. Lo que se altera es el tiempo en rojo. Debido a ello, se obtiene el nuevo flujo crítico, despejando la



variable  $Y$  de la ecuación (4) y el nuevo flujo de saturación para el tiempo en verde asignado según el flujo de vehículos despejando la variable  $y_i$  de la ecuación (5).

Los tiempos en verde de las demás fases se aumentan para compensar el tiempo de ampliación del semáforo en rojo. El cálculo de la nueva duración en verde de los semáforos se calcula incrementando sus flujos de saturación. Los nuevos flujos de saturación se obtienen a partir de la diferencia entre el nuevo flujo crítico y el flujo de saturación de la fase que se amplía el tiempo en rojo, obtenidos anteriormente, y del sumatorio de los flujos de saturación de las fases en que se incrementan la duración en verde de los semáforos. Para cada fase, su nuevo flujo de saturación es directamente proporcional al flujo de saturación evaluado según el flujo vehicular y a la diferencia entre el nuevo flujo crítico y el flujo de saturación de la fase que no garantiza suficiente tiempo como para que crucen los peatones e inversamente proporcional a la suma de los flujos de saturación de las demás fases. Una vez obtenidos los flujos de saturación para cada fase, se evalúan los nuevos tiempos de fase.

$$y_i = \frac{y_i \text{ flujo vehicular} * (Y_{actual} - y_{ff})}{\sum y_{fnf}}$$

*Ecuación 13. Flujo de saturación para garantizar el tiempo de cruce de los peatones*

$Y_{actual}$ : Nuevo flujo crítico

$y_{ff}$ : Nuevo flujo de saturación de la fase que no garantiza tiempo para cruzar

$y_{fnf}$ : Flujos de saturación de las demás fases

El cálculo del tiempo de las fases de los semáforos de una intersección se realiza a partir del flujo de saturación más elevado que posee el conjunto de los movimientos para cada fase. En el caso de que haya una única fase, la ecuación para calcular el tiempo en verde según el flujo de saturación no es válida ya que dicho tiempo sería igual a la diferencia entre el tiempo de ciclo y el tiempo de pérdidas y el tiempo en rojo sería nulo. Para solucionar este problema, se evalúa el tiempo en verde de la fase como tiempo de ciclo. El nuevo tiempo de ciclo se establece a partir del valor mínimo o máximo para los que se acomoda la mentalidad del conductor y del peatón. Por ello, si el tiempo de ciclo es inferior a 35 segundos, el sistema selecciona los 35 segundos como tiempo de ciclo y si es superior a dicho tiempo, se emplea el máximo tiempo de ciclo, 120 segundos. A continuación, el programa obtiene el flujo crítico para el tiempo de ciclo seleccionado y el flujo de saturación a partir del nuevo flujo crítico. Este hecho, permite el empleo de la ecuación para el cálculo del tiempo de fase otorgando así duración al disco rojo y de esta forma garantizar la asignación de tiempo para que los peatones puedan cruzar la calzada.



El sistema inteligente de gestión de tráfico permite la ampliación del tiempo de fase si el flujo vehicular no disminuye del 10% una vez alcanzado el tiempo óptimo. El tiempo en que una fase está activa es limitado ya que el tiempo de ciclo no puede sobrepasar los 120 segundos, por lo que se establecen tiempos máximos para cada fase. El tiempo máximo de cada fase se evalúa de forma proporcional a los flujos de saturación. La adquisición de los máximos flujos de saturación se realiza a partir del flujo crítico evaluado para los 120 segundos. El máximo flujo de saturación para cada fase es directamente proporcional al flujo crítico anterior y al flujo óptimo de saturación e inversamente proporcional al flujo óptimo crítico. Por último, se obtienen los tiempos máximos de fase aplicando la ecuación que calcula los tiempos en verde.

$$y_{i \text{ máx}} = \frac{y_i * Y_{\text{máx}}}{Y}$$

*Ecuación 14. Flujo de saturación máximo*

Una vez evaluados los tiempos iniciales y el número de movimientos principales que constituye la intersección, el programa SCT crea semáforos para que la información que se intercambia con los procesos SIV y que este abre de forma automática, se produzca de forma secuencial. Este hecho garantiza la sincronización entre los programas y se evitan problemas de exclusión mutua. El proceso SCT crea dos grupos de semáforos, uno para el envío de la fase activa y otro para la recepción del área de ocupación de los vehículos y de los peatones. La cantidad de semáforos que posee cada grupo es idéntico y depende del número de movimientos principales que posea la intersección que es igual al número de programas SIV que abre SCT. Los movimientos principales en una intersección corresponden a la dirección que sigue la carretera antes del semáforo.

A continuación, se efectúa la apertura de los programas SIV de forma ordenada para asignar a cada uno de ellos su identificación y la dirección IP de la cámara a la que se conectará para la adquisición de las imágenes de los vehículos. Dicha identificación corresponde con el número de movimiento principal. El número total de movimientos principales coincide con la cantidad de accesos a la intersección. Una vez que se han enviado dichos datos, el programa SCT se mantiene a la espera hasta que el programa SIV que se acaba de abrir, reciba los datos y desbloquee el semáforo que no permite al programa SCT continuar con su ejecución.

El cálculo del tiempo del ciclo semafórico se efectúa mediante la densidad de ocupación, por lo que el programa recibe el área total de las regiones de interés de los vehículos y de los peatones en primer lugar. La adquisición de las áreas totales que envían los procesos SIV se obtienen de forma ordenada ya que la duración en captar las regiones es diferente para cada situación y el programa debe identificar el proceso que le envía los datos. Para ello, el programa SCT desbloquea los programas SIV en orden ascendente de número de movimiento y este se bloquea hasta que el



programa SIV desbloqueado por el SCT escribe los datos en el segmento de memoria. Una vez que el programa ha escrito las áreas totales, este proceso desbloquea el programa de control SCT y este accede a la memoria para adquirir dichas áreas.

En el primer arranque del sistema inteligente de gestión de tráfico, se inicializan los semáforos que constituyen la intersección. La primera fase activa corresponde a la número 1, fase principal, por lo que los semáforos que se ubican en los movimientos principales que pertenecen a la fase estarán en verde. Los semáforos de dichos movimientos que no pertenezcan a la primera fase estarán en rojo para los vehículos y en verde para los peatones. En el momento en que se produce el primer análisis de la densidad de ocupación, el programa decide si continua con la fase activa o realiza un cambio de fase.

A continuación, comienza la ejecución continua del programa. El sistema permite la modificación de la fase activa de modo manual a través del programa CCT por lo que, en primer lugar, se efectúa la lectura de los datos enviados por el proceso CCT. Si el operario ha activado el modo manual, el sistema no analiza la información recibida por los procesos que controlan las cámaras de videovigilancia ni cambia de fase cuando se cumplan las condiciones establecidas para ello. El programa SCT únicamente comprueba si la fase recibida es la misma que la fase activa. En el caso de que sea así, ese proceso no realiza ninguna acción y si es diferente, el proceso realiza un cambio de fase. Una vez efectuado el cambio de fase, el proceso se mantiene en bucle sin realizar ninguna acción hasta que el operario realice el cambio a modo automático. Si está en modo manual y el operario desea cambiar otra vez la fase activa, se debe volver a modo automático, retornar de nuevo a modo manual e introducir la fase que se desea activar.

Después de adquirir si el programa va a trabajar en modo manual o en modo automático, el programa SCT envía a los procesos SIV el número de fase que está activa. Los procesos necesitan saber la fase activa ya que el modo de detección tanto de los vehículos como de las personas es diferente si estos están detenidos o en movimiento. El envío de este dato, se realiza por memoria compartida y el acceso a él se regula mediante semáforos. De esta forma, varios procesos que requieran dicho dato no acceden a la vez al segmento asignado sino de forma secuencial.

Este programa modifica los tiempos de fase según el flujo de vehículos y la presencia de peatones. Debido a ello, en primer lugar, se adquieren las densidades de ocupación de los vehículos y de los peatones a partir del área total recibida y del área ocupada que envían los procesos SIV a través de memoria compartida. Al igual que los otros casos en que se emplea la memoria compartida para el intercambio de información entre procesos, se emplean semáforos para evitar problemas de exclusión mutua. En el instante en que se reciben los datos de un movimiento principal a través del programa SIV, se analiza la densidad de ocupación de los



peatones. Dicha densidad se utiliza como un detector. Esto quiere decir que si la densidad de ocupación es superior al 2%, hay una persona o un grupo de personas esperando para cruzar y si es menor o igual al 2%, el sistema comprende que no hay ninguna persona y que la densidad de ocupación que adquiere, la produce el ruido de las imágenes de las cámaras de videovigilancia.

La densidad de ocupación evaluada por el programa se analiza para que el sistema realice el cambio de fase en el momento óptimo. El cambio de fase depende del tiempo que haya transcurrido desde que se inició la fase activa, del flujo de vehículos del movimiento del que se permite el paso y de la presencia de peatones. El cambio de una fase a otra se efectúa si se cumplen alguna de las siguientes condiciones.

- Si no hay vehículos en ningún acceso a la intersección ni tampoco peatones se produce un cambio de fase, por lo que se desactiva la fase actual y se activa la fase principal que corresponde con la fase 1, si la fase actual es diferente a la principal. Esta fase no se desactiva hasta que aparece algún vehículo por otra vía de acceso o algún peatón en la fase principal. En este caso, se desactivará la fase actual y se activaría la siguiente fase, es decir, la número 2. En ambos casos, se desactivaría la fase principal, el sistema volvería a funcionar de modo secuencial y el tiempo de ciclo se iniciaría de nuevo.
- Se produce un cambio de fase si en el acceso a la intersección no se detecta ningún vehículo en la región de interés.
- En el caso de que se amplíe el tiempo óptimo por poseer la fase una densidad de ocupación superior al 10%, este no puede superar el tiempo máximo asignado a dicha fase, por lo que una vez cumplido el tiempo máximo se produce el cambio de fase. De esta forma se garantiza que el tiempo de ciclo no supere los 120 segundos.

El tiempo óptimo de cada fase se recalcula si se produce un cambio de fase, si se detectan peatones en el área de espera o si el flujo de vehículos es superior al esperado. En el caso de que el sistema detecte peatones, este analiza si tienen tiempo suficiente como para cruzar y si no amplía el tiempo en rojo de la fase que posee el movimiento con peatones esperando.

Se puede dar el caso de que los peatones pasen por la zona de espera sin detenerse. Este hecho se produce por un comportamiento involuntario del peatón o por estar la zona de espera en la misma acera. Si ocurre este caso, el sistema vuelve a calcular los tiempos de fase sin tener en cuenta a los peatones.

Si se termina el tiempo óptimo asignado a la fase y la densidad de ocupación de los vehículos es superior al 10%, se amplía el tiempo óptimo y se recalculan los



tiempos de fase. En el caso de que se termine dicho tiempo y se siga cumpliendo la condición anterior, se vuelven a calcular dichos tiempos. El tiempo en que la fase está activa no sobrepasa el tiempo máximo asignado al inicio del ciclo para cada fase. De esta forma se garantiza que ningún usuario de la vía pública esté esperando más de 120 segundos.

En el caso de que únicamente haya vehículos por un acceso a la intersección y esta se componga de más de una fase, no se realizará el cambio de fase aunque se haya cumplido el tiempo óptimo asignado a no ser que haya peatones esperando para cruzar la calzada.

Inmediatamente después de que el sistema decida el cambio de fase o dicho cambio se realice de forma manual desde una sala de control de tráfico, se produce la modificación de las luces activas de los discos de los semáforos. En primer lugar, se desactiva la luz verde de los semáforos de los vehículos y se activa la luz ámbar durante 3 segundos. A continuación, se activa la fase que posee un tiempo en verde superior a 0. Debido a ello, si la siguiente fase a activar se caracteriza por poseer un tiempo en verde nulo, el sistema pasa automáticamente a la siguiente fase.

En el caso de que no haya vehículos ni personas se activa la fase principal, es decir, la fase 1, y si el cambio de fase lo ordena un operario desde el programa CCT, la fase a activar es la que el operario haya introducido por pantalla. Esta fase se recibe en el proceso SCT a través de la red. Este cambio de secuencia es válido para intersecciones con más de una fase. En el caso de que el sistema posea una única fase, éste establece la fase 0 si el semáforo que enfoca a los vehículos está en verde, y la fase 1 si dicho semáforo está en rojo.

En los instantes en que se realiza un cambio de fase, se detecta un peatón o se amplía el tiempo de fases por poseer la fase activa una densidad de ocupación vehicular superior al 10%, se efectúa una modificación de ciclo de tal forma que se adaptan los tiempos para adquirir los tiempos óptimos de las fases, para garantizar el cruce de los peatones o para disminuir el excesivo flujo vehicular de un movimiento. En las modificaciones de los tiempos de fases, no se establece un tiempo de ciclo mínimo ya que, como por ejemplo si no hay vehículos por una o varias fases ni tampoco peatones, dicho tiempo podrá ser inferior a 35 segundos.

El cálculo de los nuevos tiempos óptimos de fase se realiza, en primer lugar, a partir de la densidad de ocupación de los vehículos. Los datos referentes a la intensidad vehicular introducidos por el operario en la puesta en marcha del sistema se adquieren en un momento determinado. Debido a ello, el tiempo de las fases no se adapta al flujo vehicular en tiempo real. Dichos datos, se emplean en este proyecto como datos iniciales que establecen la proporción de vehículos que circula por cada carril, por lo que se utilizan para calcular los flujos de saturación en tiempo



real según la densidad de ocupación. En la siguiente ilustración, se muestra el diagrama de flujo para la modificación del tiempo de fases.

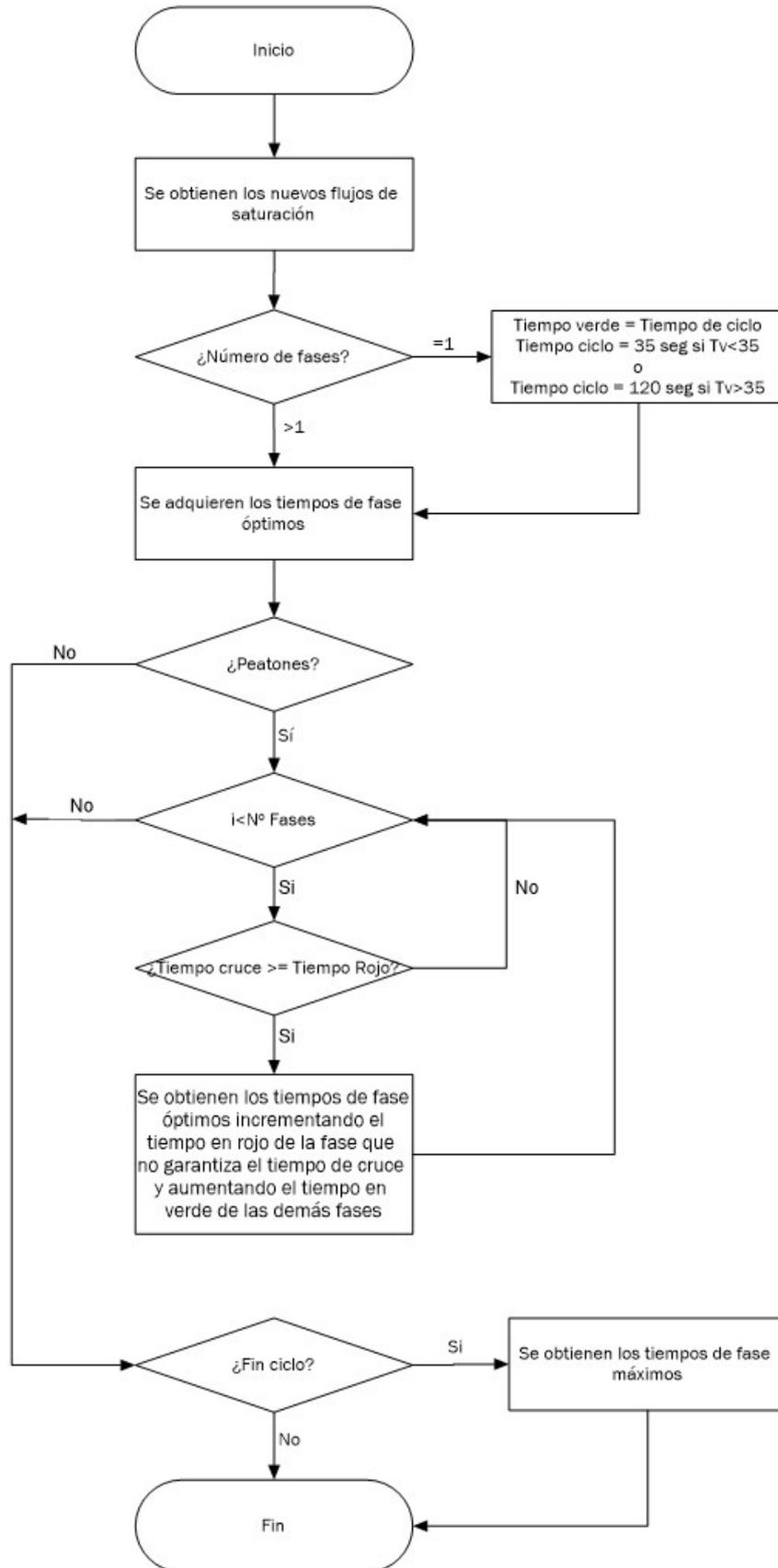


Figura 14. Diagrama de flujo modificación tiempos de fases



En primer lugar, se obtiene el nuevo flujo de saturación. Éste es directamente proporcional a la densidad de ocupación y al flujo de saturación inicial, por lo que el flujo de saturación para los vehículos que se encuentran actualmente en la región de interés corresponde al tanto por ciento de la densidad de ocupación del flujo de saturación inicial. Una vez obtenidos los flujos de saturación para cada movimiento, se adquiere el flujo crítico mediante la suma de los mayores flujos de saturación de las diferentes fases. A continuación, se calculan los tiempos de las fases a partir de los flujos de saturación obtenidos. Si el sistema solo posee una fase, el tiempo en verde será igual al tiempo de ciclo y dicho tiempo se evaluará en 35 segundos si la duración en verde es inferior o en 120 segundos si es superior.

En el caso de que no haya vehículos en ningún acceso a la intersección pero si peatones, se recalcula el tiempo de fases ya que el obtenido según el flujo de vehículos daría un valor nulo para todos los semáforos y activaría la fase principal por lo que no se permitiría el paso de los peatones. Para que esto no ocurra, se asigna a cada fase un tiempo en verde mínimo. Para la obtención de dichos tiempos, se incrementa en una unidad el tiempo de ciclo y se calcula el nuevo flujo crítico. A continuación, se adquiere los nuevos flujos de saturación. Dichos flujos se caracterizan por ser directamente proporcional al flujo de saturación inicial y al nuevo flujo crítico evaluado e inversamente proporcional al flujo crítico inicial. A partir de dichos flujos de saturación, se obtienen los tiempos de fase del ciclo.

El sistema garantiza que el tiempo en que el semáforo de los vehículos está en rojo es suficiente para que los peatones crucen la calzada. Para ello se verifica si el tiempo en que el semáforo está en rojo es igual o superior al tiempo de cruce. En el caso de que no se produzca la condición descrita, el programa modifica los tiempos de fase para que se cumpla dicha condición. Esta modificación se produce de igual forma que la realizada en el cálculo inicial del ciclo semafórico, por lo que se incrementa el tiempo de ciclo la diferencia entre el tiempo de cruce y el tiempo que el semáforo está en rojo. Acto seguido, se evalúa el nuevo flujo crítico para nuevo tiempo de ciclo y se calcula el flujo de saturación de la fase que no posee tiempo suficiente como para permitir con seguridad el paso de los peatones. El cálculo del nuevo flujo de saturación se obtiene a partir de la diferencia entre el nuevo flujo crítico y el flujo de saturación de la fase expuesta y mediante el sumatorio del flujo de saturación de las demás fases. El nuevo flujo de saturación para cada fase es directamente proporcional al flujo de saturación inicial y a la diferencia evaluada e inversamente proporcional a la sumatoria calculada. Los tiempos de fase se adquieren a partir de los nuevos flujos de saturación. Este sistema es válido si el sistema posee más de una fase, en el caso de que no sea así, el sistema tasa el tiempo en verde del semáforo a partir del nuevo flujo de saturación.

El tiempo en que una fase está activa es limitado, por lo que si las condiciones que muestra el flujo de vehículos requieren un aumento de tiempo, este se puede



incrementar hasta que se alcanza el máximo tiempo establecido para dicha fase. Los tiempos máximos se obtienen a partir de las saturaciones máximas que se adquieren para un tiempo de ciclo de 120 segundos. Las saturaciones máximas se caracterizan por ser directamente proporcional al flujo de saturación óptimo y al flujo crítico máximo e inversamente proporcional al flujo crítico óptimo. El flujo crítico máximo se evalúa despejando la variable que corresponde al flujo crítico de la ecuación que tasa el tiempo de ciclo para una duración máxima de 120 segundos. A partir de los flujos de saturación calculados, se obtienen los tiempos máximos de las fases.

A continuación, se remite la información relativa a la fase activa, a la densidad de ocupación de los vehículos y a la existencia o no de peatones esperando para cruzar la calzada al programa IOT desarrollado en Python. El lenguaje de programación C++ permite la ejecución de un programa escrito en Python y el paso de parámetros por argumentos entre C++ y Python. Para ello, en primer lugar, se inicializan los módulos y se declaran las variables que contendrán los datos que se enviarán por argumento. Una vez declaradas, se indica el nombre del programa que se debe ejecutar, IOT, y la función a la que se enviarán los datos por argumento. Acto seguido, se realiza la transformación del tipo de variable int de C++ al tipo de variable int de Python y se establece un argumento de 4 parámetros. Cada variable transformada, se introduce en una posición. La ubicación de las variables en el argumento coincide con las posiciones que ocupan en la función del programa IOT. Por último, se envían los parámetros del argumento a la función y se eliminan los objetos creados y la memoria empleada.

Después de remitir los parámetros al programa IOT, se emite la fase activa al programa que se ubica en el centro de control de tráfico CCT para que el operario sepa dicha fase y la modifique si lo desea. La transmisión de la fase activa se realiza a través de la red.

#### 3.4.4 PROGRAMA SIV

El proceso que interpreta la secuencia de imágenes de la cámara de videovigilancia se denomina SIV, sistema inteligente de vehículos. Este programa se caracteriza por adquirir la región de interés de los vehículos y obtener de ella tanto el área total como el área ocupada. Estos datos junto con los recibidos de los programas que tratan las imágenes de los peatones los envía al proceso de control, SCT, para su análisis. A continuación, se expone el diagrama de flujo del programa SIV.

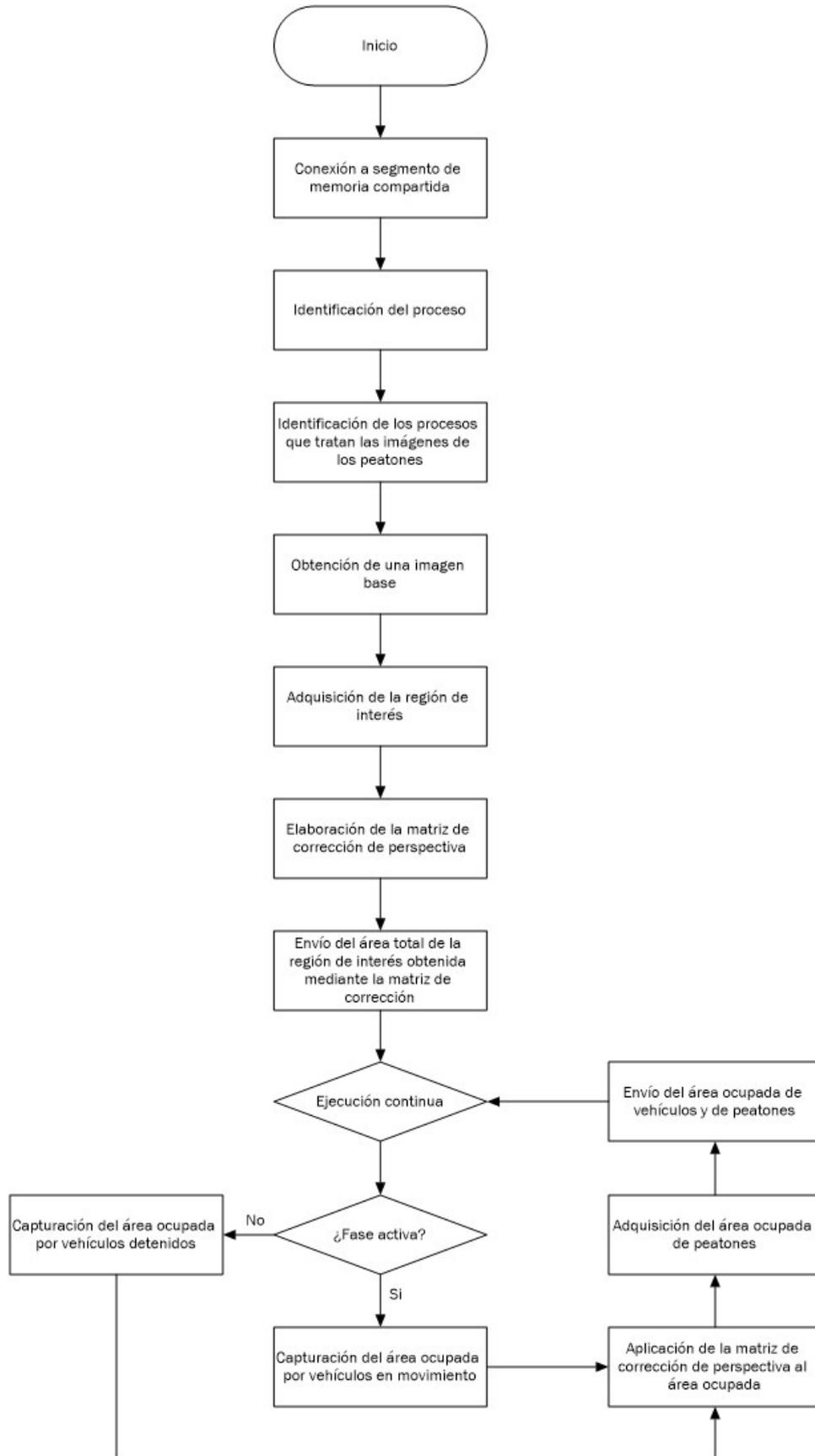


Figura 15. Diagrama flujo programa SIV



En la puesta en marcha del sistema, el programa SIV, en primer lugar, se conecta al segmento de memoria compartida reservado por el proceso SCT. El programa de control SCT establece un identificador para el segmento reservado, por lo que para que el programa SIV se conecte a dicho segmento se implementa el mismo identificador. Este se adquiere mediante una ruta de un fichero y un número entero. Una vez que se realiza la conexión, se adquiere la dirección de memoria en la que se ubica la variable que se desea leer o escribir. La información intercambiada entre el proceso SIV y el proceso SCT se caracteriza por englobarse en cuatro estructuras, la primera para la identificación del proceso, la segunda para los datos de las áreas iniciales del cruce, la tercera para las áreas ocupadas de dicho cruce, y la última para recibir la fase activa. Estas estructuras se introducen dentro del mismo segmento de memoria pero en diferentes direcciones, por lo que el programa SCT reserva suficiente memoria como para almacenar las estructuras descritas.

Si la intersección posee más de un movimiento principal, el programa SCT abre tantos programas SIV como movimientos principales haya. Debido a ello, el acceso o la modificación de los datos que se ubican en la memoria compartida debe estar controlado ya que si dos o más procesos acceden a la vez, se producen problemas en la sección crítica por lo que, tanto los datos escritos como leídos podrían ser incorrectos. Para solucionar este problema se implantan semáforos. La conexión de los semáforos con los creados por el programa SCT se realiza de idéntica forma que para el enlace del proceso SIV con el segmento de memoria compartida.

A continuación, se efectúa la identificación del proceso. El programa SCT inmediatamente después de realizar la apertura del proceso SIV, almacena en la dirección de memoria en la que se posiciona la estructura de identificación, el número de movimiento principal (identificación), la cantidad de procesos SIV que abre y la dirección IP de la cámara de videovigilancia a la que se debe conectar. Estos datos los recoge el proceso SIV. Además, se asigna el número de semáforo que corresponde al grupo de semáforos que controlan el acceso a la memoria compartida para el envío del área total y del área ocupada y el número de semáforo que corresponde al grupo de semáforos que controlan el envío de la fase activa a los programas que tratan la secuencia de imágenes de las cámaras de los peatones.

El programa SIV se encarga de abrir automáticamente los procesos que interpretan las imágenes de los peatones. Los cruces de calzada se caracterizan por poseer dos zonas de espera. Debido a ello, cada proceso SIV inicia dos programas SIP con los que establecerá comunicación, por lo que se crean dos grupos de semáforos para controlar el acceso a la sección crítica de la memoria, uno para el envío del área total y ocupada de los peatones y otro para la recepción de la fase activa. En el momento en que se realiza la apertura de un semáforo se le asigna a este una clave para que otros procesos los puedan emplear. En el caso de que la intersección se caracterice por poseer más de un cruce, el proceso SCT abrirá tantos



programas SIV como accesos haya y estos, a su vez, abrirán cada uno de ellos dos procesos SIP. Debido a ello, la identificación que posea cada par de procesos SIP debe ser diferente. Esto se consigue incrementando el número que se emplea para adquirir la identificación del semáforo, el número de proceso que lo crea, es decir, el número de proceso que se le asigne al programa SIV. En el momento en que se abre un proceso SIP, el programa SIV introduce el número de proceso, la cantidad de procesos y la identificación del programa SIV en la memoria compartida y este se bloquea hasta que el proceso SIP adquiere dichos datos.

Los procesos SIP intercambian información con el programa SIV, por lo que cada proceso SIV reserva un segmento de memoria compartida. Este segmento requiere de una identificación y esta debe ser diferente para cada par de procesos SIP, por lo que se incrementa el número entero que se requiere para dicha identificación, el número de identificación que asigna el proceso SCT a ese proceso SIV. La información que se transfiere entre ambos programas corresponde al área total de la zona de espera de los peatones, el grado de ocupación de dicha zona y la fase que se encuentra activa.

Una vez que se han creado los procesos SIP y se ha establecido comunicación entre dichos procesos y el programa SIV, se adquiere la región de interés de los vehículos. Para ello, se obtiene una imagen base, es decir, sin vehículos ni peatones para así, disminuir los falsos positivos que se podrían dar en la selección de la región del área de análisis.

La imagen base se obtiene a partir del análisis de una secuencia de imágenes que se almacenan en un vector durante 10 segundos, por lo que el total de imágenes que se analizan ascienden a un total de 250. Estas imágenes se guardan en escala de grises ya que para la detección de las líneas de delimitación de carril y la línea de detención no es necesario que estén en color. Además, el análisis de las imágenes en color aumenta el coste computacional y en el caso de que la intersección posea un número elevado de accesos, la puesta en marcha del sistema podría llevar bastante tiempo.

Las imágenes en escala de grises se caracterizan por mostrar el brillo de cada píxel en una matriz unidimensional. El número de píxeles a analizar depende de la dimensión que posean las imágenes que se almacenan en el vector. En este proyecto se establece para todas las imágenes una dimensión de 600 píxeles de ancho por 350 píxeles de alto, por lo que el total de píxeles a analizar asciende a 210.000. La imagen base se fundamenta en obtener de las imágenes, los objetos que están estáticos, por lo que no se adquieren objetos en movimiento. Para saber que objetos están estáticos y cuales en dinámico, se comparan los píxeles que se ubican en la misma posición de la matriz de todas las imágenes que se almacenan en el vector y se asigna en la misma posición de la matriz que constituye la imagen base, el píxel



que su valor se repita más veces. Debido a ello, lo que se implementa en la imagen base es la moda de los píxeles.

A continuación, se realiza la segmentación de la imagen, es decir, se procede a la detección de las líneas de delimitación de carril y a la línea de detención. Tanto las líneas de delimitación como la línea de detención se caracterizan por poseer tonalidades blancas, por lo que en primer lugar se adquieren los objetos de la imagen que poseen dicha tonalidad. El programa se configura inicialmente para que adquiera los objetos que poseen un 60% de tonalidad. En el caso de que no detecte el área de análisis, se disminuye de forma automática el tanto por ciento de la tonalidad de blanco que poseen los objetos. La disminución de dicha tonalidad podría dar lugar a falsos positivos, por lo que si el blanco de las líneas de delimitación y de detención ha perdido bastante intensidad, se recomienda volver a pintarlas o poner sobre ellas una lona blanca del mismo ancho que la línea de detección y tan largo como se quiera. Además, si la pintura de las líneas está desgastada, el sistema podría tardar bastante tiempo en adquirir la región de análisis.

Una vez obtenidos los objetos con tonalidad blanca, se adquieren sus contornos mediante el filtro de Canny. Si la pintura de las líneas está desgastada, los píxeles que forman su contorno pueden estar separados o introducirse en el interior del rectángulo. En ambos casos, no se detectarían los segmentos de líneas por lo que se dilatan los contornos de los objetos para solventar este problema.



*Ilustración 1. Objetos con tonalidad blanca (izquierda). Contornos objetos blancos (derecha)*

En el momento en que se produce dicha dilatación, se adquieren las líneas rectas que poseen los contornos de los objetos. En primer lugar, se realiza la selección de la línea de detención. Esta línea se caracteriza por ser muy próxima a la horizontal y por poseer una longitud que no debe ser superior al 60% de la anchura de la imagen base. Debido a ello, se calcula el ángulo de todas las líneas que detecta la función HoughLinesP y la distancia euclídea de cada una de ellas, por lo que las líneas que posean un ángulo inferior a  $10^\circ$  y una distancia inferior a la descrita pasan a la siguiente fase de análisis.



*Ilustración 2. Líneas detectadas por la función HoughLinesP*

La instrucción HoughLinesP puede detectar varias líneas de una única línea que posee el contorno de la imagen. Estas líneas tienen diferentes dimensiones y se ubican muy próximas unas con otras. Para la detección de la región de interés únicamente se necesita una línea por cada línea de delimitación y de detención, por lo que se selecciona de cada conjunto de líneas que están separadas entre ellas una distancia inferior a 10 píxeles, la línea que posea una distancia euclídea mayor.



*Ilustración 3. Selección líneas verticales carretera*



En segundo lugar, se efectúa la selección de las líneas de delimitación de carril. Las líneas de delimitación se aproximan a la verticalidad, por lo que se evalúa el ángulo de todos los segmentos que se obtienen con la función HoughLinesP y se descartan los que posean un ángulo inferior a  $15^\circ$ . En este caso, no importa la distancia euclídea de las rectas, como ocurre para el análisis de las líneas horizontales, ya que interesa que sean lo más largas posibles para así analizar una mayor región.

En la selección de las líneas horizontales sucede lo mismo que en el análisis de las líneas verticales, es decir, la función HoughLinesP obtiene varias líneas para una única línea de contorno. La solución para este problema es idéntica que para el realizado en la selección de las líneas horizontales, por lo que se escoge la mayor recta del conjunto de rectas que estén separadas entre sí un máximo de 9 píxeles.



*Ilustración 4. Selección líneas horizontales carretera*

En la siguiente fase, se procede a la clasificación de las rectas. Las condiciones que se deben producir para dicha clasificación se basan en la proximidad del extremo inferior de las líneas de delimitación con la línea de detención y en el ángulo que forma cada línea de delimitación con la línea de detención. La clasificación se realiza comprobando para cada recta horizontal, las rectas verticales. Si la distancia entre uno de los extremos de la recta horizontal está a una distancia inferior a 10 píxeles de un punto de la recta vertical y el ángulo que forman ambas rectas se encuentra en un intervalo comprendido entre  $20^\circ$  y  $90^\circ$ , las rectas analizadas pueden ser la de detención y la recta que delimita el carril por un extremo. Se continúa analizando las rectas verticales para la recta horizontal anterior y si se encuentra una que cumpla los parámetros establecidos, el sistema adquiere la línea de detección y las líneas



de delimitación del carril. En el caso de que no se detecte otra o ninguna línea vertical que cumpla con dichos parámetros, se analizará la siguiente línea horizontal con todas las verticales.

Si se analizan todas las rectas y no se cumplen las condiciones, se disminuye el número de votos. Los votos es el número de puntos que debe poseer un grupo de píxeles para ser considerados como una línea. Para ello, se calcula la ecuación de la recta que pasaría por dicho píxel y si dicha ecuación coincide con la de otros píxeles, se incrementa el número de votos. Este parámetro se establece en la función HoughLinesP. Si el número de votos alcanza el valor nulo, el programa SIV no detecta la línea de detención ni las líneas de delimitación para la tonalidad de blanco establecida, por lo que se disminuye dicha tonalidad un 2% y se vuelve a establecer el número de votos en 100. Interesa un número de votos elevado ya que se adquieren las líneas de forma más efectiva. Estas acciones se repiten hasta que el sistema adquiere las líneas mencionadas.



*Ilustración 5. Líneas verticales y horizontales que cumplen las condiciones descritas*

La zona de análisis en que deben empezar a detectarse las características del tráfico se establece a partir de la unión de los extremos superiores de las líneas de delimitación. Una vez implementada dicha línea, ya se tiene la región de interés para el tráfico vehicular.

El análisis del flujo de tráfico se efectúa mediante la densidad de ocupación, relación entre el área ocupada y el área total, por lo que se debe corregir la perspectiva que posee la región de interés para que el área que ocupa un vehículo sea igual en cualquier posición de dicha región. Si no se corrigiese la perspectiva, la



adquisición de los tiempos de fase por el programa SCT a partir del análisis del flujo de vehículos sería erróneo ya que estos ocupan menos espacio a medida que se ubican más lejos de la cámara de videovigilancia y el tiempo evaluado por el proceso mencionado sería inferior al que realmente debería tener.

La corrección de perspectiva se realiza implementando diferente peso a los píxeles que constituyen el área de análisis. Para ello, se establece una línea vertical entre la línea de detención y la línea de detección y se fijan diferentes puntos de control sobre dicha línea. El número total de puntos se evalúa en 10, por lo que la distancia entre dichos puntos corresponde a la relación entre la distancia de la línea vertical y el número total de puntos. Una vez que se obtiene la distancia entre puntos, los puntos de control se tasan incrementando el punto de control actual con dicha distancia.

A continuación, se obtienen las áreas que ocupa el mismo vehículo en los diferentes puntos de control, por lo que se analizan secuencias de imágenes de las cámaras de videovigilancia hasta que se hayan adquirido todas las áreas. Si el acceso a la intersección se compone de más de un carril, se obtiene el área de los vehículos que circulan por cada carril en cada punto de control. Los carriles de acceso poseen la misma distancia entre ellos tanto en la línea de detención como en la línea de detección. Debido a ello, se divide cada línea anterior entre el número total de carriles y se implementan las líneas de delimitación entre los puntos que corresponden por orden a la línea de detención y a la línea de delimitación.

La detección de los vehículos se realiza mediante el método de sustracción de fondo. Si los píxeles que componen los vehículos se caracterizan por poseer un valor parecido, el sistema detectará que dichos píxeles son estáticos y los establecerá con el valor 0, negro. Este problema se solventa aplicando operaciones morfológicas. En primer lugar, se realiza una erosión de la imagen, para eliminar el ruido, y a continuación un cierre para establecer como píxeles dinámicos, las regiones que el método detecta como estáticas. La matriz que se aplica al cierre no debe poseer unas dimensiones elevadas ya que produciría la unión de los vehículos, por lo que si hay áreas elevadas, se adquieren los contornos de dichas áreas y se rellenan de color blanco, es decir, se les asigna el valor 255.

La distinción entre los vehículos que circulan por diferentes carriles se realiza a partir del rectángulo circunscrito de cada vehículo y las rectas de delimitación de carril. Si la distancia de cada vértice del rectángulo con cada una de las rectas es inferior a 15 píxeles, el vehículo circula por el carril que delimitan dichas rectas.

Una vez que se clasifican los vehículos según el carril por el que circulan que aparecen en un frame, se analiza si se ubican en los puntos de control en los que se adquiere el área. De un frame al siguiente, se puede dar el caso de que el vehículo no se posicione de forma exacta en dicho punto de control, por lo que se establece



un rango en el que se puede ubicar para pertenecer a dicho punto. El rango se establece entre la mitad del punto superior y la mitad del punto inferior. Si la coordenada y del centroide del vehículo posee un valor inferior a la mitad superior o superior o igual a la mitad inferior, se adquiere el área del vehículo. Dicho área se caracteriza por ser un paralelogramo, en el que su largo corresponde a la distancia entre las líneas de delimitación del carril y su ancho a la distancia entre los vértices inferiores y los puntos que forman la intersección entre la línea horizontal superior del rectángulo y las líneas de delimitación del carril. El área en los diferentes puntos de control se obtiene del mismo vehículo, por lo que para que no se obtenga el área de otro vehículo que este en el mismo carril, se establece que si no se ha adquirido el área del punto de control anterior, el área del vehículo que pasa por el punto de control no se tiene en cuenta.

La variación del peso de los píxeles de una región de interés en perspectiva, se evalúa según la ecuación de la recta que se obtiene para el conjunto de áreas adquiridas en un carril en cada punto de control. Dicha ecuación, se obtiene mediante el método de mínimos cuadrados para así, disminuir el error. A partir de la ecuación de la recta, se calculan las áreas para todos los puntos que componen la recta vertical que se posiciona entre la línea de detención y la línea de detección y se evalúa la relación que existe entre el área que ocupa el vehículo en la línea de detención con el área que ocupa en los diferentes puntos de la recta. La relación evaluada para cada punto, se disponen en cada fila del área de análisis en una matriz de corrección.

A continuación, se evalúa el área total a partir de la sumatoria de los diferentes pesos impuestos en la matriz de corrección. Una vez que se ha evaluado el área total, se adquieren las áreas totales de las dos regiones de interés de los peatones que abre cada proceso SIV a través de la estructura que envía el área total de los peatones, y este remite dichas áreas junto con el área total evaluada para los vehículos al programa SCT mediante la estructura que remite la información inicial. Esta información se transmite mediante memoria compartida y se emplea el grupo de semáforos que se encargan de regular el acceso a la sección crítica para el envío o recepción de las áreas totales y ocupadas de cada proceso para así evitar problemas de exclusión mutua.

En el momento en que el proceso SIV remite las áreas totales de los peatones y de los vehículos al programa SCT, comienza la ejecución continua del programa. El análisis de la secuencia de imágenes que se capturan de las cámaras de videovigilancia se realiza frame a frame, se escalan a un tamaño de 600 píxeles de ancho por 350 píxeles de largo y se transforma la imagen en color a imagen en escala de grises. Una vez que se obtiene la imagen en escala de grises, se recorta para solo analizar la región de interés. Dicho recorte, se realiza rellenando de color blanco el interior de la región de interés en una imagen auxiliar y efectuando la operación



lógica and entre el frame y la imagen anterior. De esta forma, si el píxel de la imagen auxiliar vale 0, éste no pertenece a la región de interés y si vale 1 si pertenece.

La forma de obtención del área ocupada por los vehículos es diferente si estos están detenidos o en movimiento, por lo que el programa debe saber si la fase a la que corresponde el movimiento principal del programa SIV que corresponde con la fase activa. Debido a ello, el programa recibe la fase activa y la remite a los procesos que interpretan las cámaras de los peatones mediante memoria compartida. La transmisión de dicha información entre el proceso SCT y el programa SIV se realiza a través del segmento de memoria reservado por el proceso SCT y el envío de la fase activa entre el proceso SIV y los programas SIP se efectúa mediante el segmento de memoria que ha reservado el proceso SIV para la transferencia de información con los procesos que él abre de forma automática. El acceso a los segmentos de memoria está controlado por el grupo de semáforos que se encargan de regular la adquisición de la fase activa entre procesos. El control entre el programa SCT y los procesos SIV lo realiza el grupo de semáforos creados por SCT y el control entre el programa SIV y los procesos SIP lo efectúa el grupo de semáforos establecidos por SIV.

El área ocupada de los vehículos se obtiene mediante el método de sustracción de fondo<sup>[61][62]</sup>. Este método se caracteriza por obtener objetos en movimiento a partir de un modelo de fondo. Los objetos capturados se representan en una imagen binaria. El método de sustracción de fondo, primer plano, se basa en la diferencia entre el frame actual y un modelo obtenido de una secuencia de imágenes. Este modelo muestra los objetos estáticos de la escena, fondo de imagen. El método de sustracción empleado es el denominado BackgroundSubtractorMOG2 ya que selecciona un número adecuado de distribución gaussiana para cada píxel pudiéndose así adaptar con mayor rapidez a los cambios bruscos de iluminación.

Este método, tal y como se ha dicho anteriormente, se adapta muy rápido a los cambios que haya en la imágenes. Debido a ello, un vehículo que se encuentre detenido en un semáforo desaparecerá de la imagen binaria que muestra los movimientos ya que la función introducirá al vehículo como un objeto estático. Este hecho conlleva que la adquisición del área ocupada sea diferente para vehículos detenidos que para vehículos en movimiento.

Si la fase está activa, los vehículos se encuentran en movimiento y el método de sustracción les reconocerá. En el caso de que la fase no este activa, los vehículos se encuentran detenidos y el método de sustracción les absorberá, por lo que hay que captar dichos vehículos antes de que esto ocurra. La adquisición del área ocupada de los vehículos que están estáticos se realiza disminuyendo la actualización del modelo de fondo.



El método de sustracción realiza una comparación entre los píxeles de la imagen que recibe de la cámara y los píxeles de la imagen modelo. En el caso de que los píxeles a comparar sean diferentes, la función establece que el objeto al que pertenece dicho píxel está en movimiento por lo que se muestra en la imagen binaria. Un objeto puede tener píxeles que posean un brillo muy parecido a los de la imagen modelo, por lo que puede haber zonas que no las detecte como que estuviesen en movimiento. Este hecho se soluciona aplicando las mismas operaciones morfológicas que las establecidas para la obtención de la matriz de corrección

En el caso de que la fase no esté activa, los vehículos están detenidos por lo que se adquieren estos antes de que el método de sustracción les absorba. Para ello, se emplea el método de detección de objetos estáticos “Motion Filtering + ACC”<sup>[63]</sup>. Esta técnica se basa en adquirir máscaras de frente en el intervalo de tiempo en el que el método no absorbe a los vehículos. Una vez transcurrido dicho tiempo, el sistema compara los píxeles que posee la primera imagen con los píxeles que se ubican en la misma posición de las demás imágenes. Si todas las imágenes poseen para dicha ubicación de píxel el mismo valor, la imagen resultante tendrá el valor de la primera. En el caso de que los valores de dicho píxel sean diferentes, ese píxel no corresponde a un objeto estático por lo que el valor que se implementa en la imagen estática será nulo.

El tiempo en que el método de sustracción de fondo absorbe al vehículo se establece en 4 segundos, por lo que cada dicho tiempo, se verifica si hay vehículos detenidos. La imagen a analizar, aunque se ha aplicado un filtro para eliminar el ruido, este puede quedar de forma residual, por lo que se establece que para que un objeto sea considerado como estático, su área debe ser superior al 2% del área total de la región de interés. Mientras unos vehículos están detenidos a la espera de que el semáforo se ponga en verde, pueden llegar otros vehículos. Dichos vehículos están en movimiento por lo que hasta que no se mantengan 4 segundos quieto el sistema no les reconocerá como estáticos. Para que el sistema los reconozca de forma instantánea se aplica la operación lógica *or* entre la imagen estática y la imagen que muestra el movimiento de los vehículos. En el momento en que se cambia de fase y el área de los vehículos en movimiento supera en un 50% el área adquirida de los vehículos en la imagen estática, se eliminan los vehículos que se ubican en la imagen estática y se analizan únicamente los que están en movimiento. Se establece la segunda condición por si hubiese algún vehículo en la intersección que impida el paso de los vehículos de la fase que se ha activado ya que sino el sistema adquiriría un área nula y volvería a cambiar de fase.

Una vez obtenida la imagen, tanto estática como dinámica, que muestra el área ocupada por los vehículos, se recorre la matriz que forma dicha imagen y si el píxel se evalúa en 255, color blanco, se adquiere el valor del píxel que se ubica en la misma posición de la matriz de corrección. El área total corresponde al sumatorio de



todos los pesos anteriores. A continuación, se adquieren las áreas ocupadas de los procesos SIP mediante el segmento de memoria compartida creada por el programa SIV para el intercambio de información con los procesos SIP que inicia. El acceso a la escritura y lectura de las áreas ocupadas entre los tres programas se coordina mediante el grupo de semáforos establecidos por el programa SIV.

Las áreas ocupadas correspondientes a los procesos que interpretan las imágenes de las cámaras que enfocan a la zona de espera de los peatones junto con el área ocupada obtenida por el programa SIV, se envían al proceso de control SCT para su análisis a partir de la estructura ocupación implementada en una región de la memoria compartida reservada por el programa SCT. El acceso a la sección crítica de cada proceso se maneja mediante el grupo de semáforo creado por SCT para ello.

En el instante en que se envían los datos correspondientes a las áreas ocupadas de los vehículos y de los peatones, se obtiene un nuevo frame y se analiza las características del tráfico de idéntica forma.

#### 3.4.5 PROGRAMA SIP

El programa SIP, sistema inteligente de peatones, se caracteriza por analizar la secuencia de imágenes de las cámaras de videovigilancia que enfocan a la zona de espera de los peatones. En la puesta en marcha del programa, se selecciona la región de interés de la zona de espera y el área total. Una vez obtenido dicho área, comienza la ejecución continua del sistema en la que se adquiere el área ocupada por los peatones. Tanto el área total como el área ocupada se envían al proceso SIV para que este los remita junto a las áreas evaluadas en el proceso mencionado al programa SCT para su estudio. Las tareas que realiza el programa SIP, se describen en el siguiente diagrama de flujo.

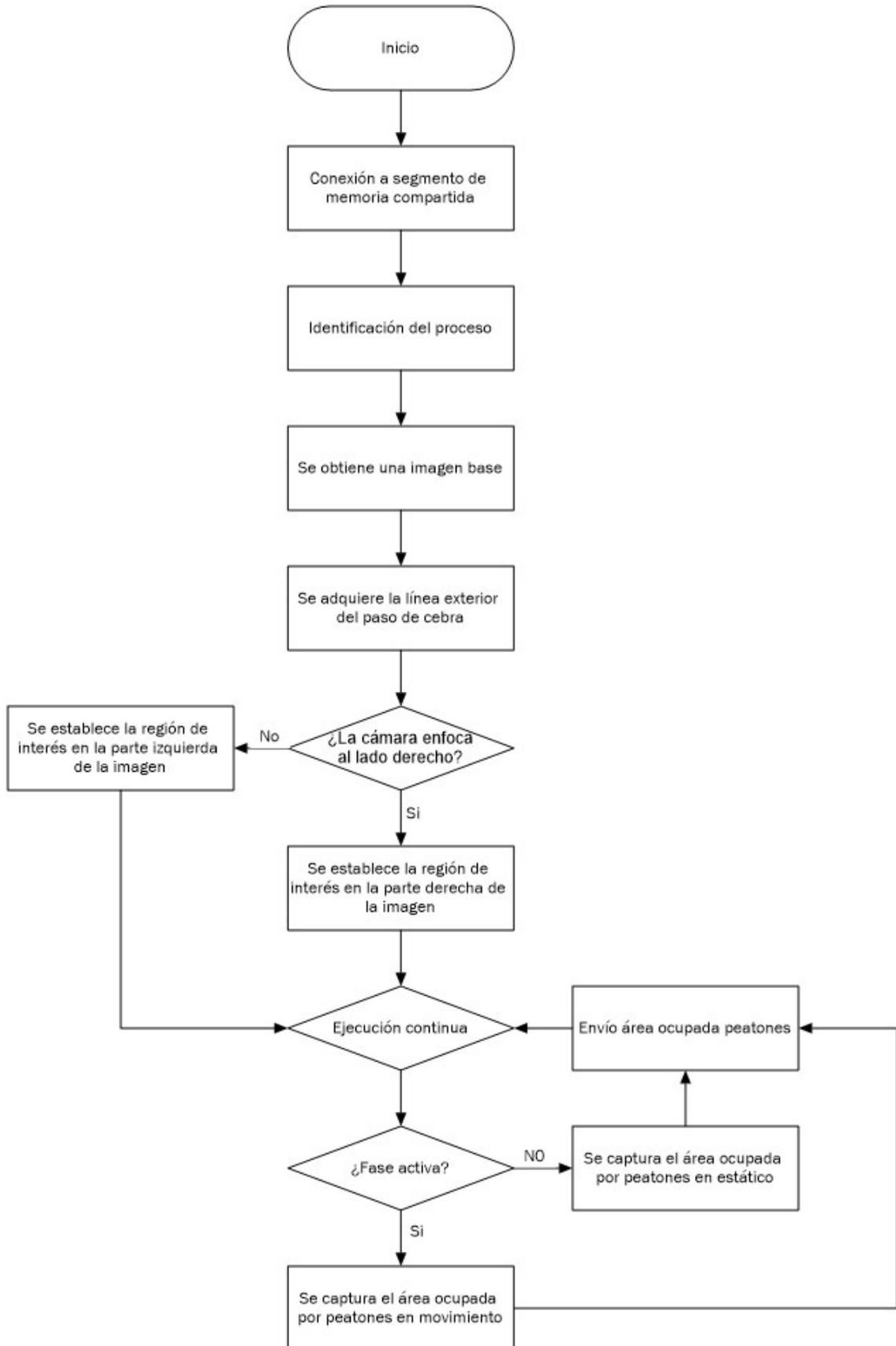


Figura 16. Diagrama flujo programa SIP



En la puesta en marcha del sistema, este programa establece conexión con el proceso SIV para adquirir la dirección IP de la cámara a la que se debe conectar para captar la secuencia de imágenes de los peatones, el número de identificación del programa que le crea, el número de identificación que se le asigna a dicho proceso y la cantidad de procesos que se conectan a la memoria compartida creada por SIV. El programa SIV se bloquea hasta que el programa SIP que ha creado, lee los datos descritos y le desbloquea para que realice la apertura del siguiente proceso SIP. El bloqueo del proceso SIV se debe a que los procesos que abre no conocen su identificación hasta que adquieren los datos de la memoria compartida y si dicho programa abriese otro SIP sin que el primero haya obtenido los datos, los procesos SIP podrían adquirir datos que no les corresponde o tener problemas con el acceso a la memoria.

En el instante en que el proceso SIP recibe los datos, los semáforos que permiten el acceso de los procesos a su sección crítica de forma controlada se conectan con el proceso SIV que le ha creado. La conexión de los semáforos de los procesos SIP con el programa SIV se realiza mediante un identificador. Los movimientos principales que constituyen las fases de la intersección se caracterizan por abrir un programa SIV, para el control de las cámaras de los vehículos, y dos procesos SIV, para el control de las cámaras de los peatones. Debido a ello, cada par de procesos SIV debe poseer un identificador diferente ya que si no habría errores en la comunicación entre procesos. La identificación se adquiere a partir de la ruta de acceso de un archivo y un número entero. Para que la identificación entre los procesos que crea el programa SCT sea diferente, se incrementa el número entero, el número de identificación del proceso SIV que le crea. En el momento en que se conectan los semáforos de ambos procesos, se desbloquea el programa SIV para que este abra el siguiente proceso SIP.

Una vez que se ha obtenido la identificación del proceso que crea el programa SIP, se realiza la conexión con el segmento de memoria compartida reservado para el intercambio de información entre los procesos SIP y el programa SIV. La identificación de dicho segmento se establece según el número de identificación del proceso SIV. El segmento de memoria creado para la recepción del número de proceso no se emplea para la transmisión de datos ya que si todos los procesos se conectan a la misma región de memoria compartida, habría problemas de exclusión mutua. Los datos que intercambian los procesos son las áreas totales, las áreas ocupadas y la fase activa.

A continuación, se adquiere una imagen sin peatones, imagen base, para así disminuir los falsos positivos que se pudiesen dar en la segmentación de la imagen. El proceso de adquisición de la imagen base es idéntico que el efectuado para el programa SIV. El proceso SIP adquiere durante 10 segundos las imágenes de la cámara de videovigilancia, las transforma a escala de grises y las almacena en un

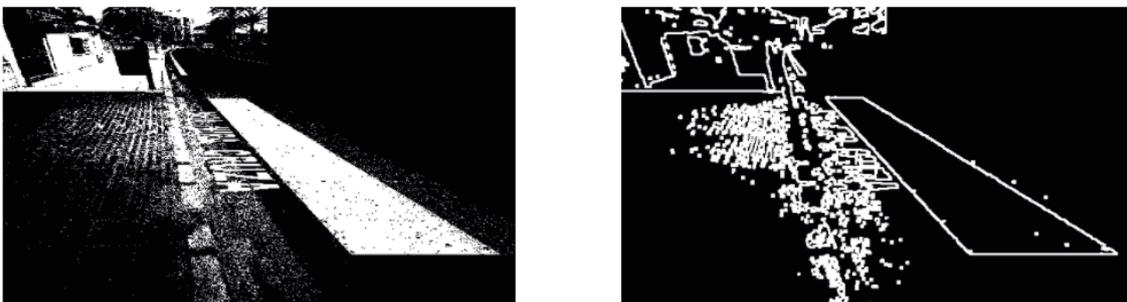


vector. Acto seguido, se analizan los píxeles que se ubican en la mismas posiciones para la totalidad de las imágenes almacenadas y se asigna en una nueva imagen, el píxel que más se repita, es decir, se calcula la moda de los píxeles que se posicionan en el mismo lugar.

La región de interés del área de espera de los peatones se obtiene a partir del rectángulo del paso de peatones que se ubica más cerca de la zona de espera. Para que este se adquiera correctamente, la cámara se debe posicionar de tal forma que se capture dicho rectángulo de forma completa pudiendo captar fragmentos de otros.

La segmentación de la imagen para la adquisición de la región de interés se basa en el color y en las líneas que se detectan de los objetos que poseen la tonalidad de color establecida. Los rectángulos que constituyen el paso de cebra se caracterizan por poseer tonalidades blancas, por lo que en primer lugar, se obtendrán los objetos que posean un 60% de blanco. Si el sistema no detecta las líneas del rectángulo para dicha tonalidad, esta se disminuye. Esta disminución puede provocar falsos positivos, por lo que si se ve que la pintura blanca del rectángulo está desgastada, se recomienda volver a pintarlo o poner una tela de color blanco y de las mismas dimensiones que el rectángulo. En el momento en que el sistema entre en ejecución continua, la tela se puede retirar ya que no realiza ninguna función.

Una vez obtenidos los objetos que poseen un 60% de tonalidad blanca, se obtienen los contornos de los objetos mencionados a partir del filtro de Canny. En el caso de que la pintura esté desgastada o se introduzca la línea exterior en el interior del rectángulo, las líneas que forman el contorno se separan y por tanto, la detección de las rectas será errónea. Para evitar este problema, se aplica la operación morfológica de dilatación, por lo que las líneas separadas se juntan.



*Ilustración 6. Ejemplo 1. Objetos con tonalidad blanca (izquierda). Contornos objetos blancos (derecha)*



*Ilustración 7. Ejemplo 2. Objetos con tonalidad blanca (izquierda). Contornos objetos blancos (derecha)*

La detección de las rectas que se ubican en la imagen se realiza mediante la función `HoughLinesP`. Esta función se caracteriza por obtener la ecuación de la recta que pasa por un píxel y analizar si hay otros píxeles que posean la misma ecuación. Al número de píxeles que poseen la misma ecuación se le denomina votos. Debido a ello, cuantos más votos tiene una recta, más probabilidad de que el conjunto de píxeles lo sea.



*Ilustración 8. Ejemplo 1. Líneas detectadas por la función `HoughLinesP`*

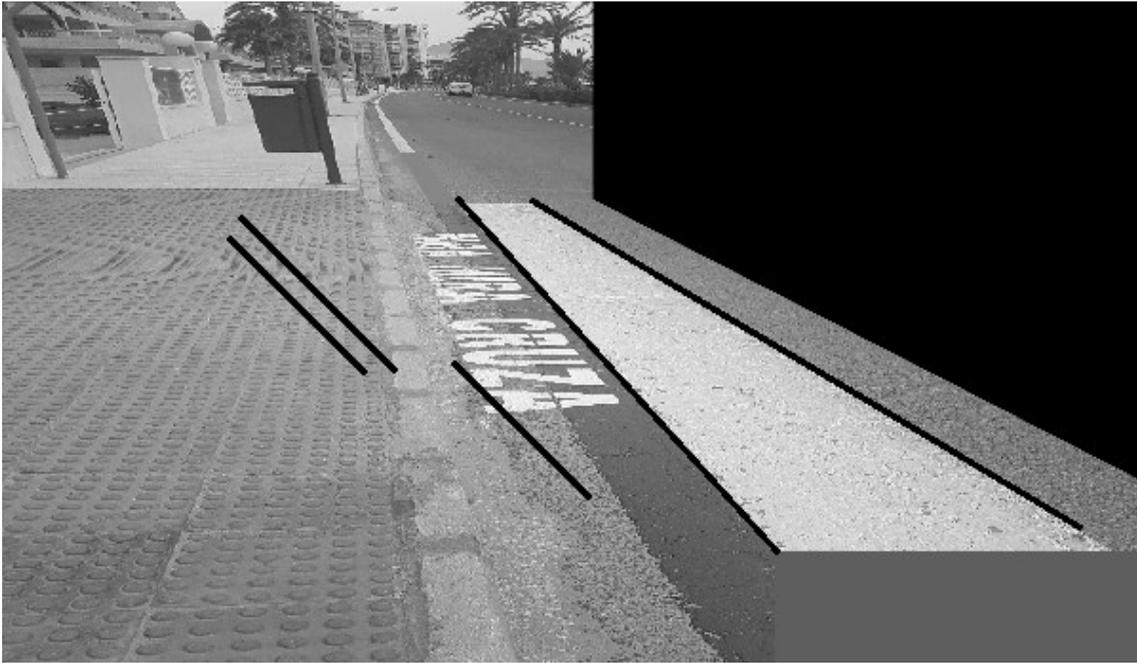


*Ilustración 9. Ejemplo 2. Líneas detectadas por la función HoughLinesP*

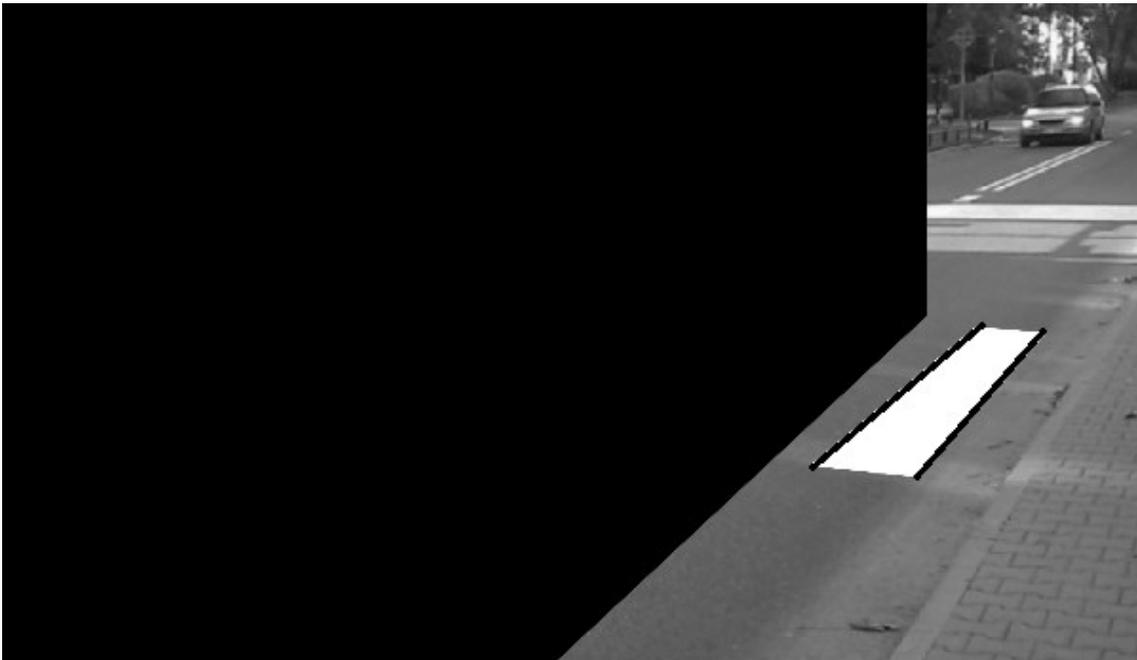
El rectángulo se constituye de dos rectas que se aproximan a la horizontal y de otras dos rectas que se aproximan a la vertical. En primer lugar, se analizan las rectas verticales. Estas rectas se caracterizan por poseer un ángulo superior a  $15^\circ$ , por lo que se evalúa el ángulo de cada línea que se adquiere con la función `HoughLinesP` y las que cumplan la condición escrita pasan a la siguiente fase.

Las líneas verticales deben tener una distancia euclídea superior a 100 píxeles, aproximadamente un 30% de la anchura total de la imagen. La distancia euclídea es la raíz cuadrada de la sumatoria de la diferencia de las coordenadas  $x$  al cuadrado y de la diferencia de las coordenadas  $y$  al cuadrado. Dichas coordenadas se refieren a los puntos extremos de una recta.

La instrucción `HoughLinesP` puede obtener un grupo de líneas de una única línea, por lo que la recta a seleccionar debe cumplir dos condiciones. La recta debe poseer una distancia euclídea superior a las demás y la separación entre ellas no debe ser superior a 10 píxeles. Esta acción se realiza para la totalidad de los grupos de líneas que se adquieren con la función mencionada.



*Ilustración 10. Ejemplo 1. Selección líneas verticales*



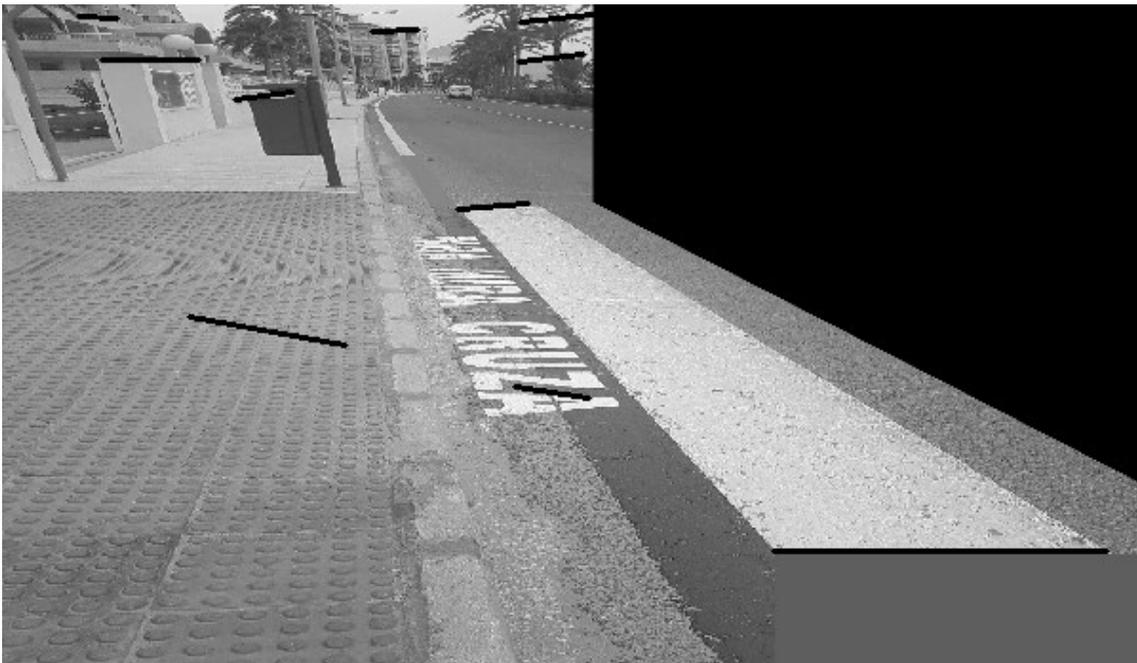
*Ilustración 11. Ejemplo 2. Selección líneas horizontales*

A continuación, se seleccionan las líneas que se caracterizan por estar muy próximas a la horizontal. Debido a ello, se evalúa el ángulo de cada línea que se ha obtenido con la instrucción `HoughLinesP` y las que poseen un ángulo inferior a  $15^\circ$  se introducen en un vector para su posterior análisis. La distancia euclídea de las líneas horizontales se caracteriza por ser superior a 20 píxeles, aproximadamente el



3% de la anchura total de la imagen, por lo que se evalúa la distancia euclídea de las rectas que se encuentran en el vector mencionado y las que cumplan la condición descrita pasan a la siguiente fase.

La función `HoughLinesP` puede adquirir varias líneas a partir de una única línea de contorno. Este hecho, ocurre también en el análisis de las líneas verticales, por lo que la solución es idéntica, es decir, se escoge la línea que posea una distancia euclídea mayor y que no esté separada más de 10 píxeles del resto de las rectas que se encuentran en el mismo grupo.



*Ilustración 12. Ejemplo 1. Selección líneas horizontales*



*Ilustración 13. Ejemplo 2. Selección líneas horizontales*

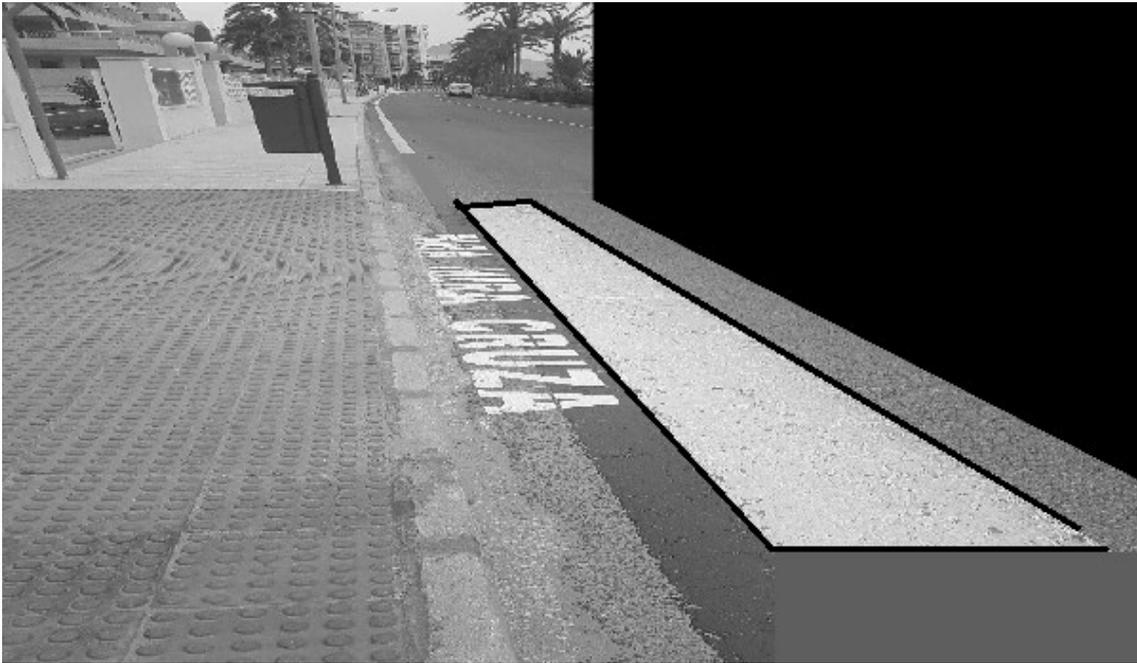
Una vez que se han seleccionado las posibles rectas horizontales y verticales que pueden formar el rectángulo del paso de cebra, se analiza cada línea vertical con la totalidad de las posibles rectas horizontales. De esta forma, se escogen las rectas verticales que pueden formar dos lados mayores del rectángulo. Dichas rectas, deben cumplir dos condiciones. La primera de ellas establece que la distancia entre la recta horizontal y la recta vertical debe ser inferior a 15 píxeles. Se escoge un valor superior a 0 ya que las líneas reconocidas por la función `HoughLinesP` pueden que no alcancen los vértices extremos que posee el rectángulo del paso de cebra. La segunda condición establece el ángulo que deben formar la recta horizontal y la recta vertical. Dicho ángulo debe estar comprendido en un intervalo entre  $20^\circ$  y  $90^\circ$ . El ángulo que constituyen ambas rectas se calcula a partir de la diferencia entre los ángulos que posee cada línea en valor absoluto.

Si se detectan dos rectas verticales que cumplan las condiciones descritas, se analiza la totalidad de las posibles líneas horizontales con las líneas verticales seleccionadas. El requisito que debe cumplir la línea horizontal para que forme parte del rectángulo se establece en la distancia a la que se ubica de cada línea vertical. En el caso de la línea horizontal se encuentre a una distancia inferior a 40 píxeles de cada línea vertical, se puede considerar que pertenece al rectángulo del paso de cebra.

El rectángulo se caracteriza por poseer dos líneas horizontales y dos líneas verticales. Si el sistema detecta un número diferente de líneas verticales, este adquiere otro par de líneas que cumplan las condiciones que se establecen para la selección de dichas rectas, y si el número total de líneas, horizontales y verticales es



diferente a 4, el sistema descarta las líneas adquiridas, disminuye el número de votos que se establece en la función HoughLinesP y vuelve a analizar las nuevas líneas que se obtienen para el nuevo número de votos. El número de votos es la cantidad de píxeles unidos que poseen la misma ecuación de la recta. A un mayor número de votos, la probabilidad de que el contorno obtenido sea una recta aumenta, por lo que si dicho parámetro disminuye, la probabilidad de que el sistema detecte rectas que no lo son, aumenta. En el caso de que no se detecten las rectas que forman el rectángulo, una vez que el número de votos ha alcanzado un valor nulo, se disminuye la tonalidad de blanco que poseen los objetos, se establece de nuevo los votos en 100 y se vuelven a analizar las rectas.



*Ilustración 14. Ejemplo 1. Líneas delimitación rectángulo paso de cebra*



*Ilustración 15. Ejemplo 2. Líneas delimitación rectángulo paso de cebra*

Una vez que se han adquirido las rectas que forman el rectángulo del paso de cebra, se adquiere la recta vertical que está más al exterior para la obtención de la región de interés. Dicha región se caracteriza por ser un polígono de cuatro lados. El primer lado del polígono se caracteriza por ser una recta paralela a la recta exterior y ubicada a 0,5 metros de ella. La región de interés de la zona de espera de los peatones no comienza en la línea exterior del rectángulo ya que se podrían detectar los vehículos que atraviesan la intersección produciendo así falsos positivos. A continuación, se implementan dos rectas horizontales que surgen de los extremos del primer lado del polígono. Dichas rectas se caracterizan por poseer una longitud de 1,5 metros. La recta que cierra el polígono se caracteriza por unir los extremos de las líneas horizontales anteriores.

La implementación de la región de interés se realiza de forma diferente si la cámara de videovigilancia se ubica a la derecha o a la izquierda del cruce. El proceso conoce su posición a partir de su identificador. Dicho identificar se lo envía el programa SIV en el instante en que realiza la apertura de este. Si el número de identificación es 0, la cámara enfoca al lado derecho de la calzada y si es 1, enfoca al lado izquierdo. La región de interés de la cámara que enfoca al lado derecho se adquiere incrementando el valor de las coordenadas x que indican la posición de los píxeles, mientras que la región de interés de la cámara que enfoca al lado izquierdo se obtiene disminuyendo el valor de dichas coordenadas.

Una vez que se obtiene la región de interés, se evalúa el área total del polígono que lo forma mediante la función `contourArea` y las esquinas de dicho polígono. A continuación, se envía dicho área al programa SIV que le crea mediante el segmento



de memoria compartida reservada por el programa SIV. Este programa abre dos procesos SIP, por lo que la transmisión del área total se controla a partir del grupo de semáforos que manejan el acceso a la sección crítica de los datos correspondientes a las áreas de los peatones.

En el momento en que se envían los datos de las áreas totales, comienza la ejecución continua del proceso. Para ello, se analizan las imágenes que se reciben de las cámaras de videovigilancia. El tratamiento de las imágenes se realiza frame a frame. Una vez obtenido el frame se recorta para así analizar únicamente la región de interés establecida anteriormente. Para ello, se implementa en una imagen un polígono a partir de las esquinas del área de análisis y se rellena de blanco, es decir, se asignan a los píxeles interiores el valor 255. Dicho polígono se ubica en la misma posición que la región de interés. A continuación, se realiza la operación lógica and entre el frame y la imagen anterior. En el caso de que el píxel de la imagen que contiene el polígono relleno posea el valor 255, es decir, color blanco, el valor del píxel del frame se asigna a una nueva imagen. Si el píxel de la imagen que contiene el polígono posee un valor nulo, el píxel de la nueva imagen poseerá un valor nulo, color negro.

El método de obtención del área ocupada por los peatones difiere según la fase activa ya que si el semáforo que enfoca a los vehículos está en verde, los peatones están detenidos por lo que se detectan regiones estáticas mientras que si el semáforo está en rojo, los peatones están en movimiento por lo que se detectan regiones dinámicas. La fase activa la envía el programa SIV a los dos procesos SIP que abre a través del segmento de memoria compartida reservado para ello. Para que los procesos accedan a la memoria de forma controlada se emplea el grupo de semáforos creados para la recepción de la fase activa. De esta forma, se evita problemas de exclusión mutua.

La obtención del área ocupada se realiza mediante el método de sustracción de fondo. Este método es el mismo que se empleó para la adquisición de las características del flujo vehicular. Si el semáforo para los peatones está en verde, el tiempo de actualización del modelo de fondo es mayor que si el semáforo está en rojo, ya que en este último caso hay que captar a los peatones antes de que la instrucción BackgroundSubtractorMOG2 les absorba y les incluya como fondo.

El método de sustracción de fondo consiste en adquirir las características estáticas de la imagen a partir de un conjunto de imágenes para así obtener una imagen modelo. La obtención de los objetos en movimiento se realiza mediante la diferencia entre el nuevo frame y la imagen modelo, por lo que se realiza la resta entre los píxeles de ambas imagen que se posicionan en el mismo lugar. Si la diferencia entre ambos píxeles es notable, el píxel que constituye dicho objeto se encuentra en movimiento por lo que en la imagen binaria se le asigna el valor 255,



es decir, el color blanco. En caso contrario, el píxel pertenece a un objeto estático por lo que se establece el valor 0 al píxel que ocupa la misma posición en la imagen binaria.

Los píxeles de un objeto en movimiento pueden tener un valor parecido a los píxeles que constituyen la imagen modelo por lo que la función BackgroundSubtractorMOG2 les reconocerá como píxeles estáticos. Para rellenar los píxeles en movimiento que el sistema no detecta, se aplican operaciones morfológicas. En primer lugar, se aplica un filtro de erosión para eliminar el ruido de la imagen y a continuación un filtro de cierre para que los píxeles que están en movimiento y que la instrucción mencionada no les detecta se pongan en blanco. El elemento estructurante que se establece para el filtro de cierre no puede tener dimensiones elevadas ya que originaría la unión objetos, por lo que para zonas en movimiento con dimensiones elevadas que la función no reconoce, se detectan los contornos y se asignan a los píxeles que se ubican en su interior el valor 255, color blanco.

Si la fase activa corresponde a la fase establecida para el proceso SIV, hay que captar a los peatones antes de que la función BackgroundSubtractorMOG2 les absorba y les incluya como modelo de fondo. El tiempo que tarda dicha función en absorber a los peatones se evalúa en 4 segundos, por lo que durante dicho tiempo se adquieren imágenes cada 0,25 segundos. Una vez transcurridos los 4 segundos, se analizan las 16 imágenes que hay almacenadas en un vector y se comparan entre ellas. Un objeto que en un principio estaba en movimiento y pasa a estático se caracteriza por mantenerse en la imagen durante 4 segundos, por lo que al analizar las imágenes que se ubican en el vector, los píxeles que constituyen el objeto se deben tasar en 255 en la totalidad de las imágenes. Debido a ello, si se aplica la operación and a todas las imágenes, se obtiene el objeto estático. Aunque se han aplicado filtros para disminuir el ruido de las imágenes, puede quedar ruido residual, por lo que si el área ocupada por el objeto estático supera un 2% del área total de la región de interés, el sistema da por bueno la adquisición del objeto estático.

Aunque el semáforo para los peatones esté en rojo, también hay que adquirir los peatones que llegan a la zona de espera y que no han estado más de 4 segundos parados. Para ello, se aplica la operación lógica or entre la imagen estática y la imagen que se obtiene de aplicar la instrucción BackgroundSubtractorMOG2. En el instante en que el semáforo para los peatones se pone en verde y si el área de los peatones en movimiento es superior a un 50% de los peatones que se encontraban en estático, el sistema obtiene únicamente a las personas en movimiento. Esta condición se establece por si los peatones no pueden cruzar la calzada por haber vehículos impidiendo el paso.



El área ocupada de los peatones tanto si están en movimiento como si están detenidos, se calcula en píxeles, por lo que dicho área es la sumatoria de todos los píxeles blancos de la imagen. Una vez obtenido el área ocupada, se envían al programa SIV mediante memoria compartida. Para controlar el acceso de cada programa a su sección crítica, se emplea el grupo de semáforos creados por el programa mencionado para el control de los datos de envío de las áreas totales y ocupadas.

Una vez que se envían dichos datos, el sistema vuelve a adquirir otro frame y lo analiza de idéntica forma, por lo que el programa a partir de este punto vuelve a ejecutar las mismas acciones para la obtención del área ocupada por los peatones.





# CAPÍTULO 4

## RESULTADOS EXPERIMENTALES





## 4 RESULTADOS EXPERIMENTALES

Los programas que constituyen el sistema inteligente de tráfico se dividen en diferentes fases. En este apartado, se analiza el algoritmo implementado en cada una de las fases. El funcionamiento de los programas SIV y SIP se verifica mediante una secuencia de imágenes de vídeo, el programa SCT a partir de un circuito hardware que permite modificar en tiempo real las características del tráfico y el proceso CCT a través del mismo ordenador en el que se ubican los programas anteriores estableciendo una IP y un puerto que simula la conexión a internet.

Los procesos que se encargan de adquirir las características del flujo vehículos se analizan mediante un vídeo adquirido de internet<sup>[64]</sup>. Este vídeo no posee línea de detención y los vehículos están en continuo movimiento. Para que esta secuencia de imágenes se adapte a los requisitos que cumple una intersección, se implanta en dicho vídeo una línea de detención y se detienen los vehículos un intervalo de tiempo limitado para verificar el comportamiento del sistema ante objetos estáticos. Los programas que interpretan el grado de ocupación de los peatones se analizan mediante un video propio ya que no había vídeos en internet que enfocaran a los peatones desde una altura parecida a la de la columna de un semáforo. Además, la cámara de los vídeos de internet estaba en movimiento y para el sistema implementado, la cámara debe estar fija.

El análisis de los datos que el proceso SCT recibe de los programas que interpretan las imágenes de las cámaras de videovigilancia no son válidos, ya que si el sistema dispone que el semáforo está en rojo los vehículos no se detienen. Para verificar el funcionamiento del sistema se emplea una Raspberry Pi conectada a un microcontrolador Arduino que recibe el flujo de vehículos a partir de la tensión que establece un potenciómetro y la detección de peatones mediante un interruptor.

La verificación del correcto funcionamiento del sistema se realiza simulando una intersección con tres accesos de doble sentido. Cada uno de los accesos se compone de tres carriles con un paso de cebra. Debido a ello, dicha intersección posee un total de 3 semáforos para el control del paso de los vehículos a la intersección y de 6 semáforos, 2 por cada acceso, para el control del cruce de la calzada por parte de los peatones.

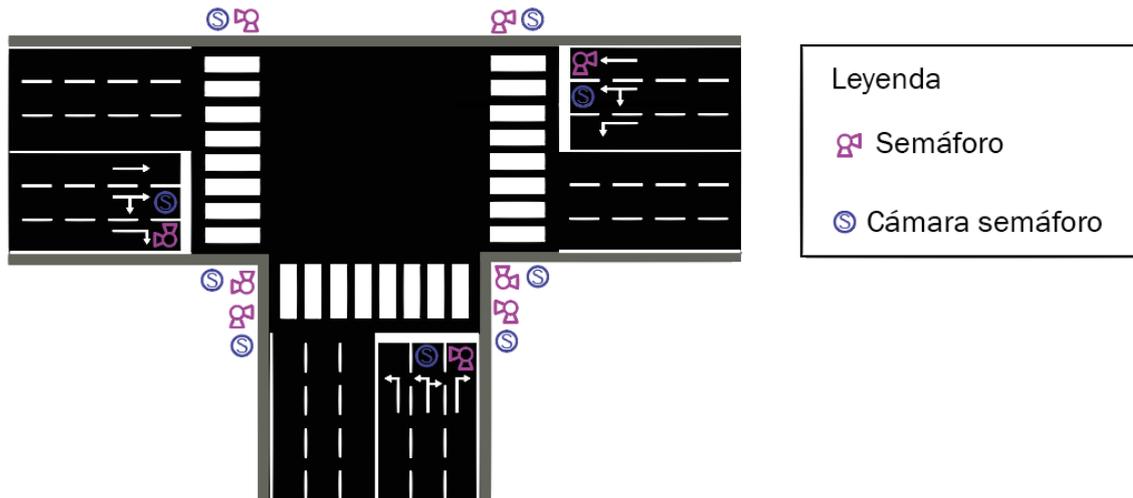


Ilustración 16. Intersección tres accesos

Los vehículos que se ubican en el lado izquierdo, según la señalización horizontal impuesta en la carretera, pueden seguir recto o girar a su derecha, los que se posicionan en el lado derecho, pueden seguir recto o girar a su izquierda y los que se posicionan en el lado frontal, pueden girar a la derecha o a la izquierda. Debido a ello, se establecen un total de tres fases para que no existan movimientos en conflicto. La primera fase corresponde a los vehículos que se encuentran en la derecha, la segunda a los que se ubican en la izquierda y la tercera a los que se posicionan de frente. En las siguientes imágenes se muestran los movimientos que se permiten en las diferentes fases. La primera fase se marca en verde, la segunda en azul y la tercera en amarillo.

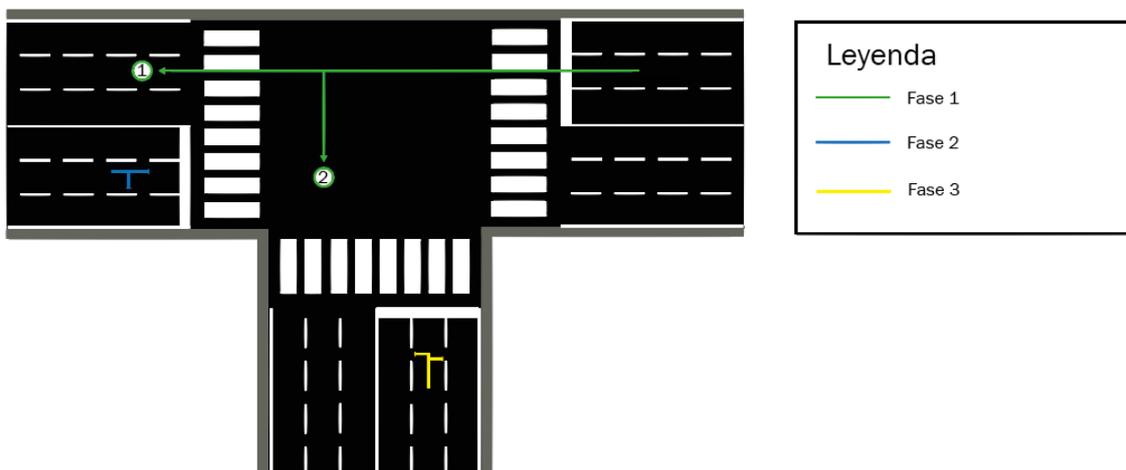


Ilustración 17. Movimientos fase 1

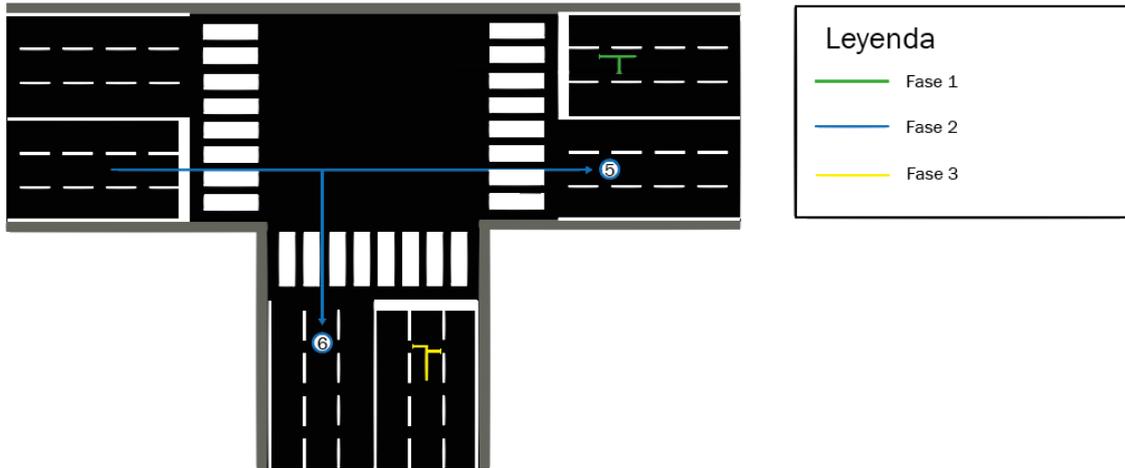


Ilustración 18. Movimientos fase 2

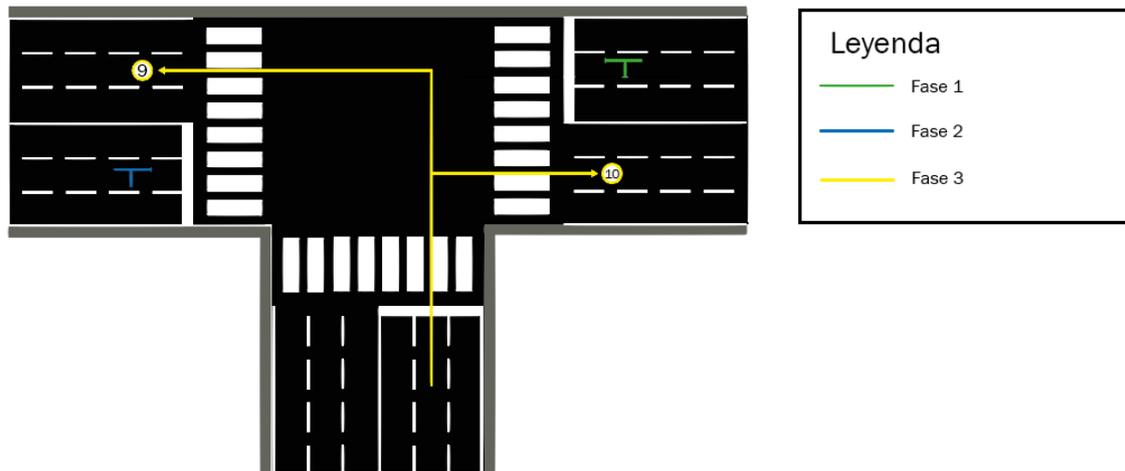


Ilustración 19. Movimientos fase 3

#### 4.1 PROGRAMA SCT

El programa de control SCT se caracteriza por controlar la secuencia de los semáforos a partir de las densidades de ocupación adquiridas. Para ello, en primer lugar, dicho programa solicita el número de fases que constituye la intersección, el número de movimientos de cada fase, si el movimiento procede de un movimiento principal y si se produce un giro en del vehículo en la intersección. En el caso de que el movimiento derive de un principal, el sistema no solicita el último parámetro ya que siempre se producirá un desvío. Una vez introducidas las características anteriores, el proceso solicita la intensidad de vehículos, el número de carriles y la longitud del paso de peatones. A partir de los datos descritos, se obtiene el ciclo semafórico. A continuación, se muestran los datos que se introducen en el sistema



para el control de la intersección compuesta por tres accesos descrita anteriormente, el tiempo de ciclo, y el tiempo asignado a cada fase.

Fase	Número Movimiento	Intensidad	Número Carriles	Factor equivalente	Flujo automóviles directos (1)	Flujo saturación (2)
1	1	510	2	1	255	0,1416
	2	155	2	1,2	93	0,0516
2	3	454	2	1	227	0,1261
	4	133	2	1,2	79,8	0,0443
3	5	240	2	1,2	144	0,0800
	6	216	2	1,2	129.6	0,0720
						Y = 0,5156

Tabla 6. Flujos de saturación iniciales óptimos

$$q_i = \frac{I_i * Feq_i}{n_i} \quad (1)$$

$$q_1 = \frac{I_1 * Feq_1}{n_1} \rightarrow q_1 = \frac{510 * 1}{2} \rightarrow q_1 = 255 \text{ veh.* carril/hora}$$

$$q_2 = \frac{I_2 * Feq_2}{n_2} \rightarrow q_2 = \frac{155 * 1,2}{2} \rightarrow q_2 = 93 \text{ veh.* carril/hora}$$

$$q_3 = \frac{I_3 * Feq_3}{n_3} \rightarrow q_3 = \frac{454 * 1}{2} \rightarrow q_3 = 227 \text{ veh.* carril/hora}$$

$$q_4 = \frac{I_4 * Feq_4}{n_4} \rightarrow q_4 = \frac{133 * 1,2}{2} \rightarrow q_4 = 79,8 \text{ veh.* carril/hora}$$

$$q_5 = \frac{I_5 * Feq_5}{n_5} \rightarrow q_5 = \frac{240 * 1,2}{2} \rightarrow q_5 = 144 \text{ veh.* carril/hora}$$

$$q_6 = \frac{I_6 * Feq_6}{n_6} \rightarrow q_6 = \frac{216 * 1,2}{2} \rightarrow q_6 = 129,6 \text{ veh.* carril/hora}$$

$$y_i = \frac{q_i}{1800} \quad (2)$$

$$y_1 = \frac{q_1}{1800} \rightarrow y_1 = \frac{255 \text{ veh.* carril/h}}{1800 \text{ veh.* carril/h}} \rightarrow y_1 = 0,1416$$

$$y_2 = \frac{q_2}{1800} \rightarrow y_2 = \frac{93 \text{ veh.* carril/h}}{1800 \text{ veh.* carril/h}} \rightarrow y_2 = 0,0516$$



$$y_3 = \frac{q_3}{1800} \rightarrow y_3 = \frac{227 \text{ veh.} * \text{ carril} / \text{ h}}{1800 \text{ veh.} * \text{ carril} / \text{ h}} \rightarrow \boxed{y_3 = 0,1261}$$

$$y_4 = \frac{q_4}{1800} \rightarrow y_4 = \frac{79,8 \text{ veh.} * \text{ carril} / \text{ h}}{1800 \text{ veh.} * \text{ carril} / \text{ h}} \rightarrow \boxed{y_4 = 0,0443}$$

$$y_5 = \frac{q_5}{1800} \rightarrow y_5 = \frac{144 \text{ veh.} * \text{ carril} / \text{ h}}{1800 \text{ veh.} * \text{ carril} / \text{ h}} \rightarrow \boxed{y_5 = 0,0800}$$

$$y_6 = \frac{q_6}{1800} \rightarrow y_6 = \frac{129,6 \text{ veh.} * \text{ carril} / \text{ h}}{1800 \text{ veh.} * \text{ carril} / \text{ h}} \rightarrow \boxed{y_6 = 0,0720}$$

$$Y_1 = 0,1416 + 0,0516 \rightarrow \boxed{Y_1 = 0,1932}$$

$$Y_2 = 0,1261 + 0,0443 \rightarrow \boxed{Y_2 = 0,1704}$$

$$Y_3 = 0,0800 + 0,0720 \rightarrow \boxed{Y_3 = 0,1520}$$

$$Y = Y_1 + Y_2 + Y_3$$

$$Y = 0,1932 + 0,1704 + 0,1520 \rightarrow \boxed{Y = 0,5156}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y}$$

$$T_p = N^{\circ} \text{ Fases} * \text{ Tiempo ambar}$$

$$T_p = 3 * 3 \text{ s} \rightarrow T_p = 9 \text{ s}$$

$$T_c = \frac{(1,5 * 9) + 5}{1 - 0,5156} \rightarrow T_c = 38,1915 \text{ s} \rightarrow \boxed{T_c \cong 38 \text{ s}}$$

$$G_1 = \frac{y_1}{y} \cdot (T_c - T_p) \rightarrow G_1 = \frac{0,1932}{0,5156} * (38 - 9) \rightarrow G_1 = 10,8665 \text{ s}$$

$$G_2 = \frac{y_2}{y} \cdot (T_c - T_p) \rightarrow G_2 = \frac{0,1704}{0,5156} * (38 - 9) \rightarrow G_2 = 9,5841 \text{ s}$$

$$G_3 = \frac{y_3}{y} \cdot (T_c - T_p) \rightarrow G_3 = \frac{0,1520}{0,5156} * (38 - 9) \rightarrow G_3 = 8,5492 \text{ s}$$



El tiempo en verde de las diferentes fases se ajusta a un número entero, sin decimales, para adaptarse al tiempo de ciclo. Para ello, se realiza el sumatorio de la parte decimal del tiempo en verde evaluado y así obtener el número de segundos que se incrementa o incrementan una o más fases. El número de fases que se incrementan depende del número de segundos y se aumenta o aumentan las fases que posean un mayor número decimal.

$$T_{\Delta} = 0,8665 + 0,5841 + 0,5492 \rightarrow T_{\Delta} = 1,9998 \rightarrow T_{\Delta} \cong 2 \text{ s}$$

El sumatorio de la parte decimal del tiempo en verde de cada fase se evalúa en 2 s, por lo que a las dos fases que se caracterizan por ser mayores, se aumenta su tiempo en un segundos.

$$G_1 = 11 \text{ s}$$

$$G_2 = 10 \text{ s}$$

$$G_3 = 8 \text{ s}$$

El tiempo en rojo de cada fase es la diferencia entre el tiempo de ciclo, el tiempo en verde de la fase y el tiempo en ámbar. A continuación, se muestran los resultados del ciclo semafórico evaluados teóricamente.

	Fase 1	Fase 2	Fase 3
Rojo	24 s	25 s	27 s
Ámbar	3 s	3 s	3 s
Verde	11 s	10 s	8 s

Tabla 7. Tiempos de fase óptimos

Acto seguido, se evalúa la duración máxima en que una fase puede estar activa. Se implementa un tiempo máximo para que el tiempo de ciclo no supere los 120 segundos, tiempo en que se acomoda fácilmente la mentalidad tanto de los conductores como de los peatones. En primer lugar, se calcula el flujo crítico para la duración máxima, y se incrementa el flujo de saturación óptimo, el tanto por ciento del total inicial.

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y} \rightarrow Y = 1 - \frac{(1,5 * T_p) + 5}{T_c}$$

$$Y_{m\acute{a}x} = 1 - \frac{(1,5 * 9) + 5}{120} \rightarrow Y_{m\acute{a}x} = 0,8458$$

$$Y_{1 \text{ m\acute{a}x}} = \frac{Y_1 * Y_{m\acute{a}x}}{Y} \rightarrow Y_{1 \text{ m\acute{a}x}} = \frac{0,1932 * 0,8458}{0,5156} \rightarrow Y_{1 \text{ m\acute{a}x}} = 0,3169$$



$$Y_{2 \text{ máx}} = \frac{Y_2 * Y_{\text{máx}}}{Y} \rightarrow Y_{2 \text{ máx}} = \frac{0,1704 * 0,8458}{0,5156} \rightarrow \boxed{Y_{2 \text{ máx}} = 0,2795}$$

$$Y_{3 \text{ máx}} = \frac{Y_3 * Y_{\text{máx}}}{Y} \rightarrow Y_{3 \text{ máx}} = \frac{0,1520 * 0,8458}{0,5156} \rightarrow \boxed{Y_{3 \text{ máx}} = 0,2493}$$

$$G_{1 \text{ máx}} = \frac{y_{1 \text{ máx}}}{y_{\text{máx}}} \cdot (T_c - T_p) \rightarrow G_{1 \text{ máx}} = \frac{0,3169}{0,8458} * (120 - 9) \rightarrow G_{1 \text{ máx}} = 41,5889 \text{ s}$$

$$G_{2 \text{ máx}} = \frac{y_{2 \text{ máx}}}{y_{\text{máx}}} \cdot (T_c - T_p) \rightarrow G_{2 \text{ máx}} = \frac{0,2795}{0,8458} * (120 - 9) \rightarrow G_{2 \text{ máx}} = 36,6806 \text{ s}$$

$$G_{3 \text{ máx}} = \frac{y_{3 \text{ máx}}}{y_{\text{máx}}} \cdot (T_c - T_p) \rightarrow G_{3 \text{ máx}} = \frac{0,2493}{0,8458} * (120 - 9) \rightarrow G_{3 \text{ máx}} = 32,7173 \text{ s}$$

$$T_{\Delta} = 0,5889 + 0,6806 + 0,7173 \rightarrow T_{\Delta} = 1,9868 \text{ s} \rightarrow T_{\Delta} \cong 2 \text{ s}$$

$$\boxed{G_{1 \text{ máx}} = 41 \text{ s}}$$

$$\boxed{G_{2 \text{ máx}} = 37 \text{ s}}$$

$$\boxed{G_{3 \text{ máx}} = 33 \text{ s}}$$

	Fase 1	Fase 2	Fase 3
Rojo	76 s	80 s	84 s
Ámbar	3 s	3 s	3 s
Verde	41 s	37 s	33 s

Tabla 8. Tiempos de fase máximos



```

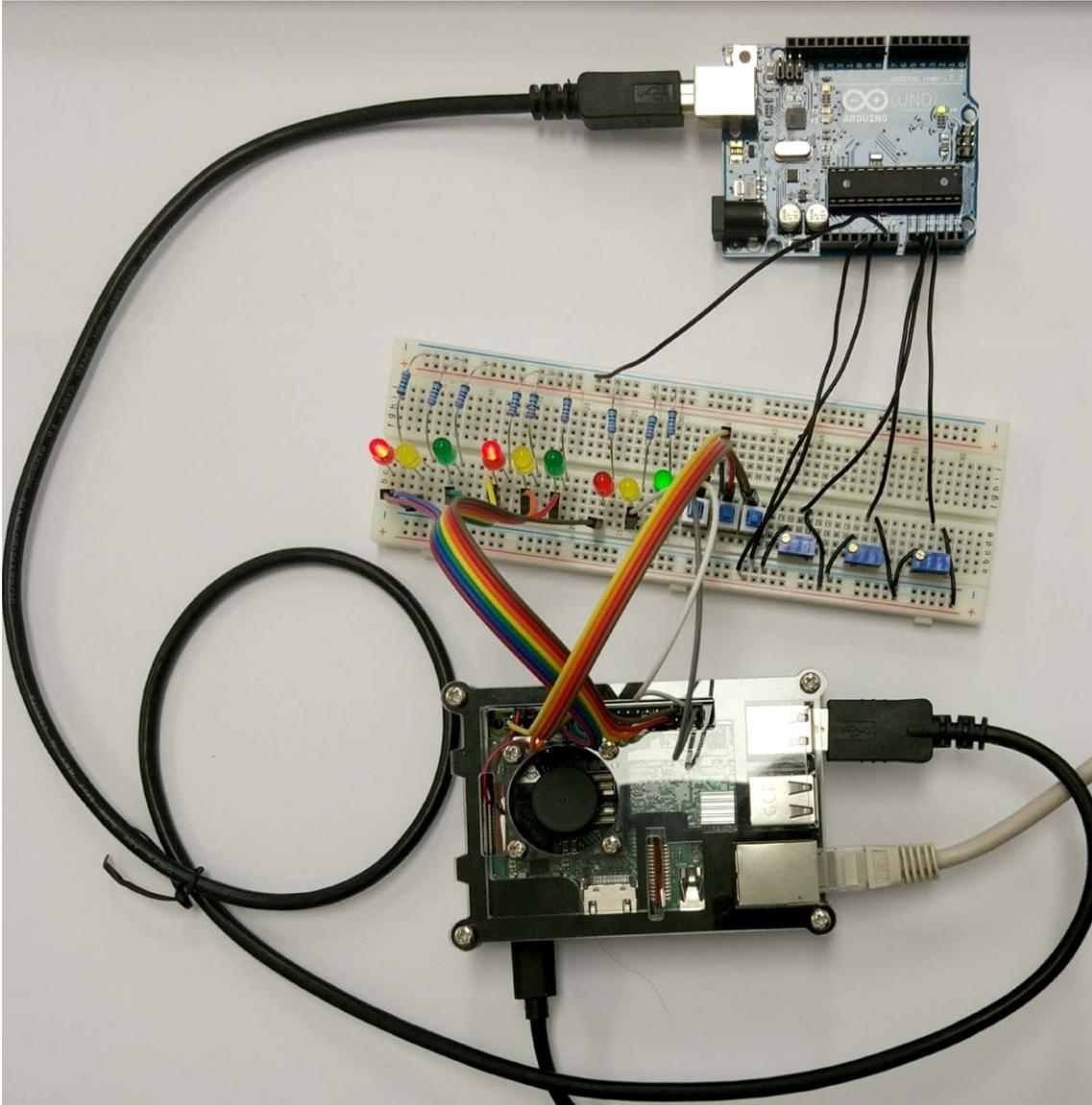
Numero Carriles: 2
Longitud de Cruce: 10
Introduzca los siguientes datos de circulacion Fase 2
Introduza los datos del carril Nº4
Intensidad Carril: 133
Numero Carriles: 2
Introduzca los siguientes datos de circulacion Fase 3
Introduza los datos del carril Nº5
Intensidad Carril: 240
Numero Carriles: 2
Longitud de Cruce: 10
Introduzca los siguientes datos de circulacion Fase 3
Introduza los datos del carril Nº6
Intensidad Carril: 216
Numero Carriles: 2
TIEMPOS INICIALES
[1, 11, 3, 24]
[2, 10, 3, 25]
[3, 8, 3, 27]
TIEMPOS MAXIMOS :
[1, 41, 3, 76]
[2, 37, 3, 80]
[3, 33, 3, 84]

```

*Ilustración 20. Tiempos óptimos y máximos obtenidos con el programa SCT*

Una vez obtenidos los tiempos iniciales y los tiempos máximos, se abren los programas SIV que controlan las cámaras que enfocan a los movimientos principales y se mantiene el proceso SCT a la espera de recibir el área total de las regiones de interés tanto de los vehículos como de los peatones. En el momento en que el programa recibe la totalidad de las áreas totales comienza la ejecución continua en el que se controla la secuencia de los semáforos a partir del flujo de vehículos y la presencia de peatones.

El funcionamiento de dicha ejecución, se verifica mediante una Raspberry Pi y un Arduino. La densidad de ocupación de los vehículos se adquiere mediante la variación de la tensión de salida producida por la modificación de la resistencia de un potenciómetro. Este aparato se conecta a una entrada analógica del microcontrolador Arduino, ya que la Raspberry Pi no posee el tipo de entradas descritas, únicamente tiene entradas digitales. Debido a ello, se implementa un programa en Arduino para que envíe los valores de tensión de salida de los potenciómetros a la Raspberry Pi mediante el Universal Serial Bus, USB. La presencia de peatones en la zona de espera se simula mediante interruptores, es decir, si hay peatones la señal vale 1 y si no hay esta vale 0. Esta señal se caracteriza por ser digital por lo que se conectan los interruptores en las entradas digitales de la Raspberry Pi. La secuencia de encendido de los discos de los semáforos, se muestra a partir de diodos LED conectados a las salidas digitales de la Raspberry Pi. Si el programa establece a la salida digital el valor 1, se enciende el diodo LED, y si le establece el valor 0, se apaga dicho diodo.



*Figura 17. Circuito simulación programa SCT*

Los potenciómetros, los interruptores y los diodos led, se colocan en una protoboard. Los potenciómetros se conectan al Arduino con cables macho - macho, y los interruptores y los diodos led se acoplan a la Raspberry Pi con cables macho - hembra.

Los potenciómetros empleados para la simulación de la densidad de ocupación, se alimentan a una tensión de 5V por la patilla N°1. La variación de tensión se origina por la modificación de la resistencia. Dicha modificación se adquiere entre las patillas N°1 y N°2 por lo que se conecta la patilla N°2 a la entrada analógica del Arduino. La patilla N°3 se acopla a masa, GND, para que se derive por dicho pin la mayor parte de la intensidad.

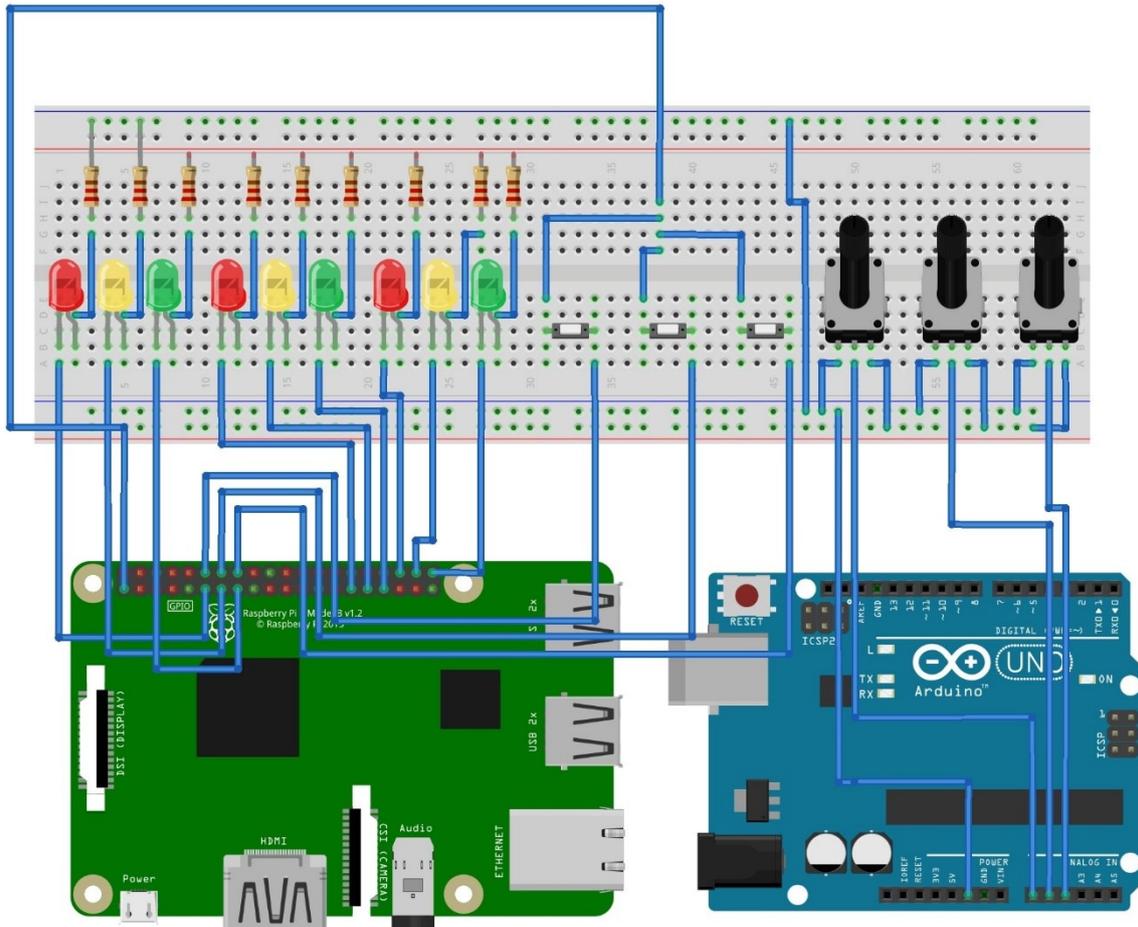


La intersección a simular dispone de tres movimientos principales, por lo que se instala un potenciómetro en la protoboard por cada movimiento. Los pines N°2 de los potenciómetros se conectan a las entradas analógicas A0, A1 y A2 del Arduino. El potenciómetro conectado a la entrada A0 dispone la densidad de ocupación del primer movimiento principal, la entrada A1, la densidad del segundo movimiento principal, y la entrada A2, la densidad del tercer movimiento principal. La densidad de ocupación aumenta rotando los tornillos de los potenciómetros en sentido de las agujas del reloj y disminuye rotando dichos tornillos en sentido contrario a las agujas del reloj.

Los interruptores se caracterizan por simular la presencia de peatones. Estos funcionan como entradas digitales ya que si están pulsados, la entrada GPIO recibe la tensión de 3,3V y le asigna el valor 1 y sin no están pulsados, la entrada GPIO recibe la tensión de 0V y le asigna el valor 0. Al funcionar como entradas digitales, se conectan directamente a los GPIO de la Raspberry Pi. Estos interruptores se alimentan conectando el cable macho – hembra a la patilla N°1 y permite o no dependiendo si esta pulsado, el paso de la tensión a la patilla N°2. Dicha patilla se conecta a una entrada digital de la Raspberry Pi. La tensión de los tres interruptores se adquiere del pin N°1, 3,3V y las entradas digitales se conectan a los pines 12, 16 y 18. El interruptor conectado al pin 12 corresponde al primer movimiento principal, el pin 16 al segundo movimiento principal y el 18 al tercer movimiento principal.

Los diodos LED reproducen el encendido de los discos de los semáforos que se ubican en la intersección. Estos se conectan a las salidas digitales GPIO de la Raspberry Pi. Si se activa un disco, se envía un 1 lógico al pin al que está conectado para que disponga su salida a 3,3V. Si se desactiva un disco, se remite un 0 lógico a dicho pin para que establezca su salida a 0V. Por cada acceso a la intersección hay 3 diodos LED, uno rojo otro amarillo y otro verde. Los diodos LED correspondientes al primer acceso se conectan a los pines 11, 13 y 15, al segundo acceso a los pines 29, 31 y 33, y al tercer acceso a los pines 36, 38 y 40. El primer pin pertenece a los diodos LED rojos, el segundo pin a los diodos LED amarillos y el tercer pin a los diodos LED verdes.

En la siguiente figura, se muestran las conexiones entre los potenciómetros y el Arduino y entre los interruptores y los diodos LED que se acoplan a la Raspberry Pi.



Fotografía 27. Esquema eléctrico para la simulación del programa SCT

Una vez implementado el hardware que simula la densidad de ocupación de los vehículos, la presencia de peatones y la secuencia de los semáforos, se expone a continuación el funcionamiento del programa. Por pantalla se imprimen los parámetros mencionados y los tiempos asignados a cada fase, en el caso de que se cambie o se amplíe la fase activa. Los tiempos máximos se modifican y se imprimen cuando finaliza el ciclo semafórico. En el instante en que comienza la ejecución continua, el programa establece como fase activa, la fase principal correspondiente a la fase número 0. Acto seguido, realiza la lectura de los potenciómetros que simulan el tiempo de flujo y si hay vehículos en la fase principal la mantiene activa hasta se cumpla una condición de cambio de fase. Si no hay vehículos, realiza el cambio de fase al acceso que sí contenga.

El caso mostrado a continuación, se caracteriza por poseer una densidad de ocupación nula para la primera y la segunda fase y para la tercera fase, una densidad tasada en 28%. Debido a ello, en el arranque del sistema se desactiva la fase principal, fase número 0, y se activa la fase número 2. El tiempo en que las fases están activas es proporcional al tanto por ciento que indica el grado de ocupación y el flujo de saturación inicial que se evalúa a partir de los datos introducidos por el



operario en la puesta en marcha. Las acciones que realiza el programa para la obtención del tiempo de fases, se muestra a continuación.

Fase	Número Movimiento	Densidad ocupación	Flujo saturación inicial	Flujo saturación actual	
1	1	0	0,1416	0	0 Y1
	2	0	0,0516	0	
2	3	0	0,1261	0	0 Y2
	4	0	0,0443	0	
3	5	28 %	0,0800	0,0224	0,0425 Y3
	6	28 %	0,0720	0,0201	
					Y = 0,0425

Tabla 9. Flujos de saturación para  $\rho_1 = 0\%$ ,  $\rho_2 = 0\%$  y  $\rho_3 = 28\%$

$$y_1 = \rho_1 * y_{i1} \rightarrow y_1 = 0 * 0,1416 \rightarrow \boxed{y_1 = 0}$$

$$y_2 = \rho_2 * y_{i2} \rightarrow y_2 = 0 * 0,0516 \rightarrow \boxed{y_2 = 0}$$

$$y_3 = \rho_3 * y_{i3} \rightarrow y_3 = 0 * 0,1261 \rightarrow \boxed{y_3 = 0}$$

$$y_4 = \rho_4 * y_{i4} \rightarrow y_4 = 0 * 0,0443 \rightarrow \boxed{y_4 = 0}$$

$$y_5 = \rho_5 * y_{i5} \rightarrow y_5 = 0,28 * 0,0800 \rightarrow \boxed{y_5 = 0,0224}$$

$$y_6 = \rho_6 * y_{i6} \rightarrow y_6 = 0,28 * 0,0720 \rightarrow \boxed{y_6 = 0,0201}$$

$$Y = Y_1 + Y_2 + Y_3 \rightarrow Y = 0 + 0 + 0,0425 \rightarrow \boxed{Y = 0,0425}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y}$$

$$T_p = N^{\circ} \text{ Fases} * \text{Tiempo ambar}$$

$$T_p = 3 * 3 \text{ s} \rightarrow T_p = 9 \text{ s}$$

$$T_c = \frac{(1,5 * 9) + 5}{1 - 0,0425} \rightarrow T_c = 19,3211 \text{ s} \rightarrow \boxed{T_c \cong 19 \text{ s}}$$

$$G_1 = \frac{y_1}{y} * (T_c - T_p) \rightarrow G_1 = \frac{0}{0,0425} * (19 - 9) \rightarrow \boxed{G_1 = 0 \text{ s}}$$

$$G_2 = \frac{y_2}{y} * (T_c - T_p) \rightarrow G_2 = \frac{0}{0,0425} * (19 - 9) \rightarrow \boxed{G_2 = 0 \text{ s}}$$

$$G_3 = \frac{y_3}{y} * (T_c - T_p) \rightarrow G_3 = \frac{0,0425}{0,0425} * (19 - 9) \rightarrow \boxed{G_3 = 10 \text{ s}}$$



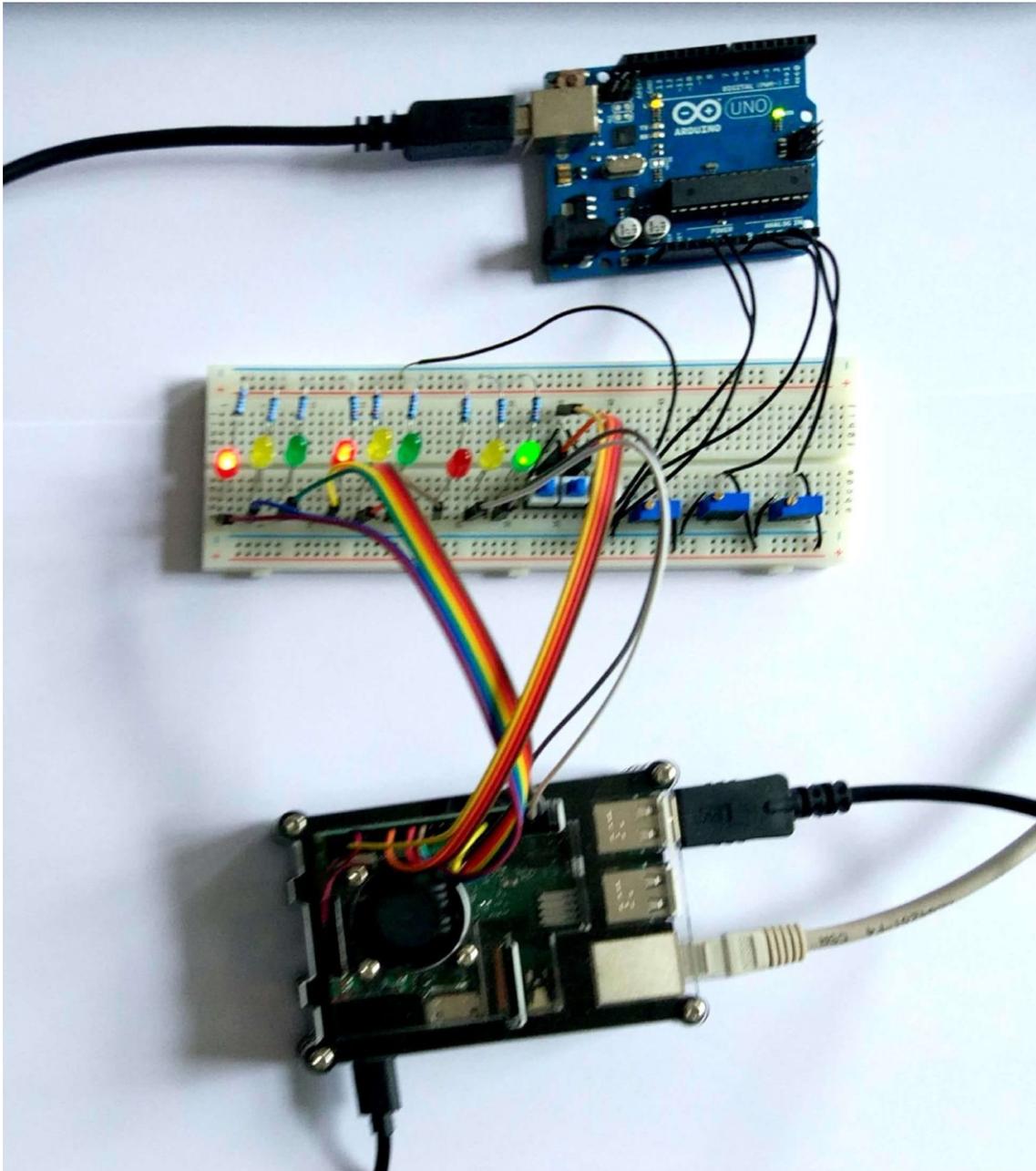
	Fase 1	Fase 2	Fase 3
Rojo	16 s	16 s	6 s
Ámbar	3 s	3 s	3 s
Verde	0 s	0 s	10 s

Tabla 10. Tiempos de fase óptimos para  $p_1 = 0\%$ ,  $p_2 = 0\%$  y  $p_3 = 28\%$

```

Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Numero Proceso Densidad : 2
Area Vehiculos Densidad: 28
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Cronometro tiempo optimo: 3
Cronometro tiempo maximo: 3
Cambio de fase ---> No hay vehiculos ni peatones en la fase
TIEMPOS MODIFICADOS
[1, 0, 3, 16]
[2, 0, 3, 16]
[3, 10, 3, 6]
Se modifica ciclo
Posición Semáforo Fase 1: [0, 1, 0]
Posición Semáforo Fase 1: [0, 0, 1]
Posición Semáforo Fase 3: [1, 0, 0]
Tiempo verde: 10
Tiempo maximo: 33
Posición Semáforo Fase 1 :[0, 0, 1]
Posición Semáforo Fase 2 :[0, 0, 1]
Posición Semáforo Fase 3 :[1, 0, 0]
Fase Activa: 3
    
```

Ilustración 21. Tiempos óptimos obtenidos con el programa SCT para  $p_1 = 0\%$ ,  $p_2 = 0\%$  y  $p_3 = 28\%$



*Ilustración 22. Cambio de fase en circuito de simulación para  $\rho_1 = 0\%$ ,  $\rho_2 = 0\%$  y  $\rho_3 = 28\%$*

En el momento en que se detectan peatones se modifican los tiempos de fase para permitir el cruce de ellos. Para ello, se obtienen los tiempos mencionados a partir de la densidad de ocupación de los vehículos que se ubican en las áreas de análisis y se verifica si el tiempo de cada fase en rojo es igual o superior al tiempo que tardan los peatones en cruzar la calzada. Si el tiempo en rojo cumple la condición mencionada, dichos tiempos serán válidos pero si el tiempo en rojo es inferior al tiempo que tardan en cruzar, se modifican los tiempos de fase. En el siguiente ejemplo, se exponen las acciones que realiza el programa para este caso con unas densidades de ocupación del 3%, 4% y 16% para cada una de las fases y se accionan los interruptores correspondientes a la segunda y a la tercera fase para simular la



presencia de peatones. La longitud de cruce para los tres casos se establece en 10 metros.

Fase	Número Movimiento	Densidad ocupación	Flujo saturación inicial	Flujo saturación actual	
1	1	3 %	0,1416	0,0042	0,0057 Y1
	2	3 %	0,0516	0,0015	
2	3	4 %	0,1261	0,0050	0,0067 Y2
	4	4 %	0,0443	0,0017	
3	5	16 %	0,0800	0,0128	0,0243 Y3
	6	16 %	0,0720	0,0115	
					Y = 0,0367

Tabla 11. Flujos de saturación para  $\rho_1 = 3\%$ ,  $\rho_2 = 4\%$  y  $\rho_3 = 16\%$

$$y_1 = \rho_1 * y_{i1} \rightarrow y_1 = 0,03 * 0,1416 \rightarrow \boxed{y_1 = 0,0042}$$

$$y_2 = \rho_2 * y_{i2} \rightarrow y_2 = 0,03 * 0,0516 \rightarrow \boxed{y_2 = 0,0015}$$

$$y_3 = \rho_3 * y_{i3} \rightarrow y_3 = 0,04 * 0,1261 \rightarrow \boxed{y_3 = 0,0050}$$

$$y_4 = \rho_4 * y_{i4} \rightarrow y_4 = 0,04 * 0,0443 \rightarrow \boxed{y_4 = 0,0017}$$

$$y_5 = \rho_5 * y_{i5} \rightarrow y_5 = 0,16 * 0,0800 \rightarrow \boxed{y_5 = 0,0128}$$

$$y_6 = \rho_6 * y_{i6} \rightarrow y_6 = 0,16 * 0,0720 \rightarrow \boxed{y_6 = 0,0115}$$

$$Y = Y_1 + Y_2 + Y_3 \rightarrow Y = 0,0057 + 0,0067 + 0,0243 \rightarrow \boxed{Y = 0,0367}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y}$$

$$T_p = N^{\circ} \text{ Fases} * \text{Tiempo ambar}$$

$$T_p = 3 * 3 \text{ s} \rightarrow T_p = 9 \text{ s}$$

$$T_c = \frac{(1,5 * 9) + 5}{1 - 0,0367} \rightarrow T_c = 19,2048 \text{ s} \rightarrow \boxed{T_c \cong 19 \text{ s}}$$

$$G_1 = \frac{y_1}{y} * (T_c - T_p) \rightarrow G_1 = \frac{0,0057}{0,0367} * (19 - 9) \rightarrow G_1 = 1,5531 \text{ s}$$

$$G_2 = \frac{y_2}{y} * (T_c - T_p) \rightarrow G_2 = \frac{0,0067}{0,0367} * (19 - 9) \rightarrow G_2 = 1,8256 \text{ s}$$

$$G_3 = \frac{y_3}{y} * (T_c - T_p) \rightarrow G_3 = \frac{0,0243}{0,0367} * (19 - 9) \rightarrow G_3 = 6,6212 \text{ s}$$



$$T_{\Delta} = 0,5531 + 0,8256 + 0,6212 \rightarrow T_{\Delta} = 1,9999 \text{ s} \rightarrow T_{\Delta} \cong 2 \text{ s}$$

$$G_1 = 1 \text{ s}$$

$$G_2 = 2 \text{ s}$$

$$G_3 = 7 \text{ s}$$

	Fase 1	Fase 2	Fase 3
Rojo	15 s	14 s	9 s
Ambar	3 s	3 s	3 s
Verde	1 s	2 s	7 s

Tabla 12. Tiempos de fase óptimos para  $\rho_1 = 3\%$ ,  $\rho_2 = 4\%$  y  $\rho_3 = 16\%$

```

Fase Activa Anterior: 3
Numero Proceso Densidad : 0
Area Vehiculos Densidad: 3
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Numero Proceso Densidad : 1
Area Vehiculos Densidad: 4
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Numero Proceso Densidad : 2
Area Vehiculos Densidad: 16
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Cronometro tiempo optimo: 6
Cronometro tiempo maximo: 54
Ampliacion Tiempo por flujo
TIEMPOS MODIFICADOS
[1, 1, 3, 15]
[2, 2, 3, 14]
[3, 7, 3, 9]
    
```

Ilustración 23. Tiempos óptimos obtenidos con el programa SCT para  $\rho_1 = 3\%$ ,  $\rho_2 = 4\%$  y  $\rho_3 = 16\%$

$$V_p = \frac{e_c}{t}$$

Ecuación 15. Movimiento rectilíneo uniforme, MRU

$$t = \frac{e_c}{V_p}$$



$$t_1 = \frac{e_{c1}}{V_p} \rightarrow t_1 = \frac{10 \text{ m}}{1,1 \text{ m/s}} \rightarrow t_1 = 9,09 \text{ s} \rightarrow \boxed{t_1 = 10 \text{ s}}$$

$$t_2 = \frac{e_{c2}}{V_p} \rightarrow t_2 = \frac{10 \text{ m}}{1,1 \text{ m/s}} \rightarrow t_1 = 9,09 \text{ s} \rightarrow \boxed{t_1 = 10 \text{ s}}$$

$$t_3 = \frac{e_{c3}}{V_p} \rightarrow t_2 = \frac{10 \text{ m}}{1,1 \text{ m/s}} \rightarrow t_1 = 9,09 \text{ s} \rightarrow \boxed{t_1 = 10 \text{ s}}$$

*Tiempo garantizado cruce sí  $T_R > T_c$*

*Fase 1  $\rightarrow 15 \text{ s} > 10 \text{ s} \rightarrow$  Tiempo suficiente para el cruce de la Fase 1*

*Fase 2  $\rightarrow 14 \text{ s} > 10 \text{ s} \rightarrow$  Tiempo suficiente para el cruce de la Fase 2*

*Fase 3  $\rightarrow 9 \text{ s} < 10 \text{ s} \rightarrow$  Tiempo insuficiente para el cruce de la Fase 3*

$$T_c = T_{c \text{ anterior}} + (T_{\text{cruce}} - T_R)$$

$$T_c = 19 + (10 - 9) \rightarrow \boxed{T_c = 20 \text{ s}}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - y} \rightarrow Y = 1 - \frac{(1,5 * T_p) + 5}{T_c}$$

$$y = 1 - \frac{(1,5 * 9) + 5}{20} \rightarrow \boxed{y = 0,0750}$$

$$G_3 = \frac{y_3}{y} \cdot (T_c - T_p) \rightarrow y_3 = \frac{G_3 \cdot y}{T_c - T_p} \rightarrow y_3 = \frac{7 \cdot 0,075}{20 - 9} \rightarrow \boxed{y_3 = 0,0477}$$

$$\Delta y_{(Fase 1 + Fase 2)} = 0,075 - 0,0477 \rightarrow \Delta y_{(Fase 1 + Fase 2)} = 0,0273$$

$$\Delta y_{(Fase 1 \text{ ant.} + Fase 2 \text{ ant.})} = 0,0057 + 0,0067 \rightarrow \Delta y_{(Fase 1 \text{ ant.} + Fase 2 \text{ ant.})} = 0,0124$$

$$y_1 = \frac{y_{1 \text{ ant.}} \cdot \Delta y_{(Fase 1 + Fase 2)}}{\Delta y_{(Fase 1 \text{ ant.} + Fase 2 \text{ ant.})}} \rightarrow y_1 = \frac{0,0057 \cdot 0,0273}{0,0124} \rightarrow \boxed{y_1 = 0,0125}$$

$$y_2 = \frac{y_{2 \text{ ant.}} \cdot \Delta y_{(Fase 1 + Fase 2)}}{\Delta y_{(Fase 1 \text{ ant.} + Fase 2 \text{ ant.})}} \rightarrow y_1 = \frac{0,0067 \cdot 0,0273}{0,0124} \rightarrow \boxed{y_2 = 0,0147}$$

$$G_1 = \frac{y_1}{y} \cdot (T_c - T_p) \rightarrow G_1 = \frac{0,0125}{0,0750} * (20 - 9) \rightarrow G_1 = 1,8333 \text{ s}$$

$$G_2 = \frac{y_2}{y} \cdot (T_c - T_p) \rightarrow G_2 = \frac{0,0147}{0,0750} * (20 - 9) \rightarrow G_2 = 2,1560 \text{ s}$$



$$T_{\Delta} = 0,8333 + 0,1560 + 0 \rightarrow T_{\Delta} = 0,9893 \text{ s} \rightarrow T_{\Delta} \cong 1 \text{ s}$$

$$G_1 = 2 \text{ s}$$

$$G_2 = 2 \text{ s}$$

$$G_3 = 7 \text{ s}$$

	Fase 1	Fase 2	Fase 3
Rojo	15 s	15 s	10 s
Ambar	3 s	3 s	3 s
Verde	2 s	2 s	7 s

Tabla 13. Tiempos de fase óptimos para  $\rho_1 = 3\%$ ,  $\rho_2 = 4\%$  y  $\rho_3 = 16\%$  con peatones en Fase 2 y 3

Los tiempos en que los semáforos están en rojo para cada fase, mostrados en la tabla anterior, se caracterizan por igualar o superar al tiempo de cruce, por lo que se garantiza tiempo suficiente como para que los peatones atraviesen el paso de cebra con seguridad.

```
Fase Activa Anterior: 3
Numero Proceso Densidad : 0
Area Vehiculos Densidad: 3
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Numero Proceso Densidad : 1
Area Vehiculos Densidad: 4
Area Peatones Densidad[0]: 1
Area Peatones Densidad[1]: 1
Numero Proceso Densidad : 2
Area Vehiculos Densidad: 16
Area Peatones Densidad[0]: 1
Area Peatones Densidad[1]: 1
Cronometro tiempo optimo: 3
Cronometro tiempo maximo: 102
Modificacion ciclo ----> HAY PEATONES
Tiempo maximo cumplido
TIEMPOS MODIFICADOS
[1, 2, 3, 15]
[2, 2, 3, 15]
[3, 7, 3, 10]
```

Ilustración 24. Tiempos óptimos obtenidos con el programa SCT para  $\rho_1 = 3\%$ ,  $\rho_2 = 4\%$  y  $\rho_3 = 16\%$  con peatones en Fase 2 y 3

En determinadas ocasiones puede que no haya vehículos en las áreas de análisis y peatones en uno o en varios accesos a la intersección. En este caso, al no haber vehículos, los peatones que estuvieran en la fase principal estarían esperando hasta que hubiese flujo de vehículos por los otros accesos. Para que esto no ocurra, se



incrementa en una unidad el tiempo de ciclo y se evalúan las densidades de ocupación como si hubiese vehículos a partir de los flujos de saturación iniciales y del flujo crítico evaluado para un tiempo de ciclo incrementado en 1 segundo. Si el tiempo en rojo es inferior al tiempo de cruce en alguna de las fases, se modifican dichos tiempos de igual forma que la realizada en el caso anterior.

Fase	Número Movimiento	Densidad ocupación	Flujo saturación inicial	Flujo saturación actual	
1	1	0 %	0,1416	0	0 Y1
	2	0 %	0,0516	0	
2	3	0 %	0,1261	0,	0 Y2
	4	0 %	0,0443	0	
3	5	0 %	0,0800	0	0 Y3
	6	0 %	0,0720	0	
					Y = 0

Tabla 14. Flujos de saturación para  $\rho_1 = 0\%$ ,  $\rho_2 = 0\%$  y  $\rho_3 = 0\%$

$$y_1 = \rho_1 * y_{i1} \rightarrow y_1 = 0 * 0,1416 \rightarrow \boxed{y_1 = 0}$$

$$y_2 = \rho_2 * y_{i2} \rightarrow y_2 = 0 * 0,0516 \rightarrow \boxed{y_2 = 0}$$

$$y_3 = \rho_3 * y_{i3} \rightarrow y_3 = 0 * 0,1261 \rightarrow \boxed{y_3 = 0}$$

$$y_4 = \rho_4 * y_{i4} \rightarrow y_4 = 0 * 0,0443 \rightarrow \boxed{y_4 = 0}$$

$$y_5 = \rho_5 * y_{i5} \rightarrow y_5 = 0 * 0,0800 \rightarrow \boxed{y_5 = 0}$$

$$y_6 = \rho_6 * y_{i6} \rightarrow y_6 = 0 * 0,0720 \rightarrow \boxed{y_6 = 0}$$

$$Y = Y_1 + Y_2 + Y_3 \rightarrow Y = 0 + 0 + 0 \rightarrow \boxed{Y = 0}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y}$$

$$T_p = N^{\circ} \text{ Fases} * \text{Tiempo ambar}$$

$$T_p = 3 * 3 \text{ s} \rightarrow T_p = 9 \text{ s}$$

$$T_c = \frac{(1,5 * 9) + 5}{1 - 0} \rightarrow T_c = 18,5 \text{ s} \rightarrow \boxed{T_c \cong 19 \text{ s}}$$

$$\boxed{G_1 = 0 \text{ s}}$$

$$\boxed{G_2 = 0 \text{ s}}$$



$$G_3 = 0 \text{ s}$$

	Fase 1	Fase 2	Fase 3
Rojo	16 s	16 s	16 s
Ambar	3 s	3 s	3 s
Verde	0 s	0 s	0 s

Tabla 15. Tiempos de fase óptimos para  $\rho_1 = 0\%$ ,  $\rho_2 = 0\%$  y  $\rho_3 = 0\%$

$$T_c = T_{c \text{ anterior}} + 1 \text{ s} \rightarrow T_c = 19 \text{ s} + 1 \text{ s} \rightarrow T_c = 20 \text{ s}$$

$$T_c = \frac{(1,5 * T_p) + 5}{1 - Y} \rightarrow Y = 1 - \frac{(1,5 * T_p) + 5}{T_c} \rightarrow Y = 0,0750$$

$$y_1 = \frac{y_{i1} * Y}{Y_i} \rightarrow y_1 = \frac{0,1416 * 0,0750}{0,5156} \rightarrow \boxed{y_1 = 0,0205}$$

$$y_2 = \frac{y_{i2} * Y}{Y_i} \rightarrow y_2 = \frac{0,0516 * 0,0750}{0,5156} \rightarrow \boxed{y_2 = 0,0075}$$

$$y_1 = \frac{y_{i1} * Y}{Y_i} \rightarrow y_1 = \frac{0,1261 * 0,0750}{0,5156} \rightarrow \boxed{y_3 = 0,0183}$$

$$y_1 = \frac{y_{i1} * Y}{Y_i} \rightarrow y_1 = \frac{0,0433 * 0,0750}{0,5156} \rightarrow \boxed{y_4 = 0,0062}$$

$$y_1 = \frac{y_{i1} * Y}{Y_i} \rightarrow y_1 = \frac{0,0800 * 0,0750}{0,5156} \rightarrow \boxed{y_5 = 0,0116}$$

$$y_1 = \frac{y_{i1} * Y}{Y_i} \rightarrow y_1 = \frac{0,0720 * 0,0750}{0,5156} \rightarrow \boxed{y_6 = 0,0104}$$

$$Y_1 = y_1 + y_2 \rightarrow Y_1 = 0,0280$$

$$Y_2 = y_3 + y_4 \rightarrow Y_2 = 0,0245$$

$$Y_3 = y_5 + y_6 \rightarrow Y_3 = 0,0220$$

$$G_1 = \frac{y_1}{y} \cdot (T_c - T_p) \rightarrow G_1 = \frac{0,0280}{0,0750} * (20 - 9) \rightarrow G_1 = 4,1066 \text{ s}$$

$$G_2 = \frac{y_2}{y} \cdot (T_c - T_p) \rightarrow G_2 = \frac{0,0245}{0,0750} * (20 - 9) \rightarrow G_2 = 3,5933 \text{ s}$$



$$G_3 = \frac{y_3}{y} \cdot (T_c - T_p) \rightarrow G_3 = \frac{0,0220}{0,0750} * (20 - 9) \rightarrow G_3 = 3,2266 \text{ s}$$

$$T_\Delta = 0,1066 + 0,5933 + 0,2266 \rightarrow T_\Delta = 0,9265 \text{ s} \rightarrow T_\Delta \cong 1 \text{ s}$$

$$G_1 = 4 \text{ s}$$

$$G_2 = 4 \text{ s}$$

$$G_3 = 3 \text{ s}$$

	Fase 1	Fase 2	Fase 3
Rojo	13 s	13 s	14 s
Ambar	3 s	3 s	3 s
Verde	4 s	4 s	3 s

Tabla 16. Tiempos de fase óptimos para  $p_1 = 0\%$ ,  $p_2 = 0\%$  y  $p_3 = 0\%$  con peatones en Fase 2 y 3

```
Fase Activa Anterior: 1
Numero Proceso Densidad : 0
Area Vehiculos Densidad: 0
Area Peatones Densidad[0]: 0
Area Peatones Densidad[1]: 0
Numero Proceso Densidad : 1
Area Vehiculos Densidad: 0
Area Peatones Densidad[0]: 1
Area Peatones Densidad[1]: 1
Numero Proceso Densidad : 2
Area Vehiculos Densidad: 1
Area Peatones Densidad[0]: 1
Area Peatones Densidad[1]: 1
Cronometro tiempo optimo: 3
Cronometro tiempo maximo: 3
Modificacion ciclo ----> HAY PEATONES
Cambio de fase ---> No hay vehiculos en la fase
TIEMPOS MODIFICADOS
[1, 4, 3, 13]
[2, 4, 3, 13]
[3, 3, 3, 14]
```

Ilustración 25. Tiempos óptimos obtenidos con el programa SCT para  $p_1 = 0\%$ ,  $p_2 = 0\%$  y  $p_3 = 0\%$  con peatones en Fase 2 y 3

## 4.2 PROGRAMA IOT Y THINGSPEAK

El proceso IOT, internet of things, desarrollado en Python IOT, adquiere las densidades de ocupación de las regiones de interés de los vehículos y la presencia



de peatones que se posicionan en la zona de espera del programa SCT. Estos datos se remiten a la plataforma web ThingSpeak. Dicha plataforma se caracteriza por admitir con la tarifa gratuita un total de 8200 mensajes al día, 1 año de almacenamiento de datos históricos y un total de 3 dispositivos enviando o recibiendo información. La visualización de los datos se puede hacer pública con lo que la información del tránsito de vehículos y la presencia de peatones lo podría ver cualquier persona desde cualquier lugar.

Para el ejemplo de intersección con tres accesos expuesto anteriormente, se abren un total de cuatro canales, uno para la fase activa y los otros tres para los datos de densidad de ocupación vehicular y presencia de peatones para cada uno de los accesos. El canal que adquiere el número de movimiento de vehículos a los que se permite el paso, se denomina “Fase Activa” y los canales que reciben los datos de los accesos, “Fase 0”, “Fase 1”, y “Fase 2”. Cada canal puede administrar un total de 8 datos, por lo que se establece en el programa IOT el campo en que se va a incluir la información. La fase que se encuentra activa, se fija en el primer campo del canal “Fase Activa”, las densidades de ocupación se disponen en el primer campo de cada uno de los canales restantes y la presencia de peatones se establecen en el segundo campo de dichos canales. En el instante en que se abre un canal, la plataforma ThingSpeak le asigna una clave. Dicha clave se emplea para remitir los datos al canal correspondiente. El programa IOT envía la información a la plataforma a través del protocolo HTTP, por lo que los datos se remiten a través de la dirección web “[https://api.thingspeak.com/update?api\\_key=](https://api.thingspeak.com/update?api_key=)”. Al final de dicha dirección, se introduce la clave y los campos en los que se disponen los datos.

A continuación, se muestran las gráficas correspondientes a los ejemplos expuestos para el programa SCT.

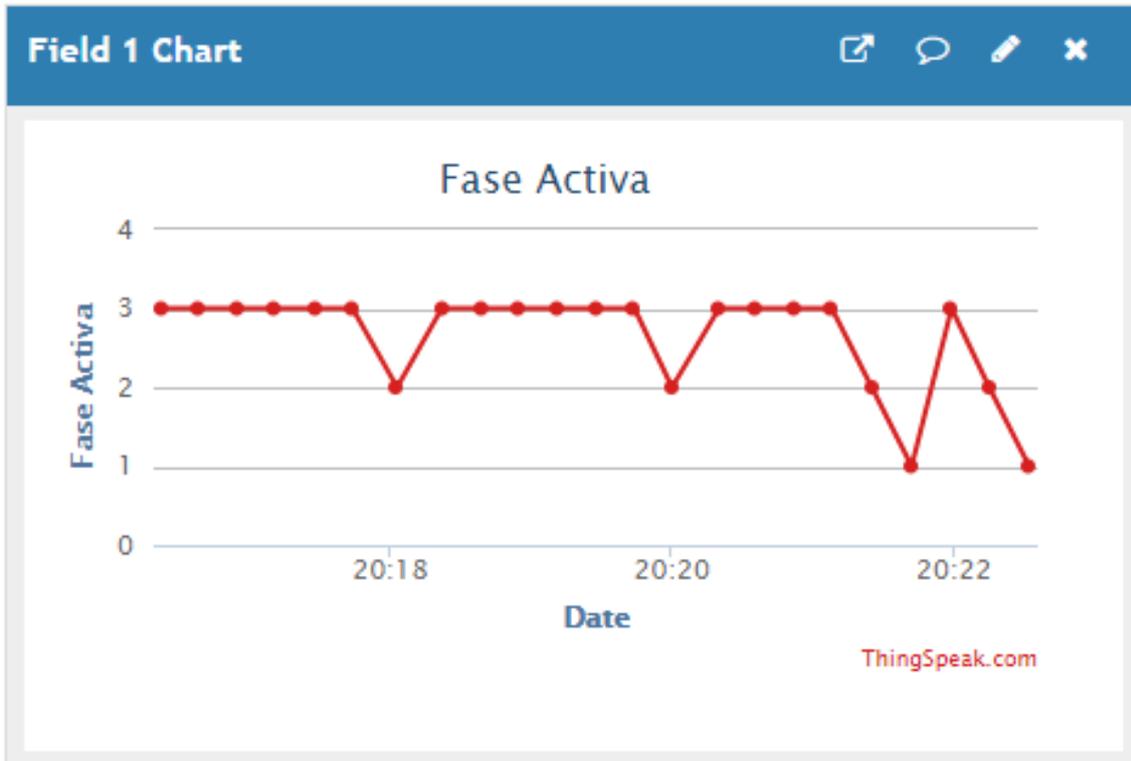


Ilustración 26. Secuencia de fases activas

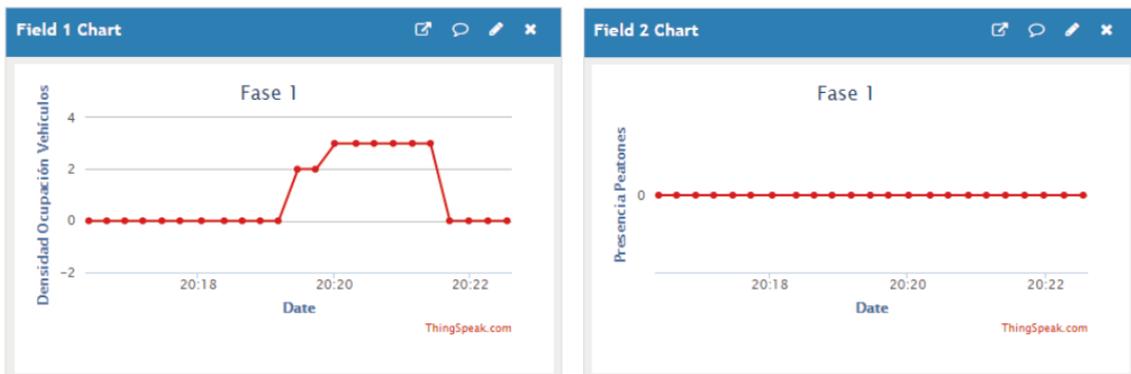


Ilustración 27. Densidad de ocupación y presencia de peatones de la fase 1



Ilustración 28. Densidad de ocupación y presencia de peatones de la fase 2





```
Archivo Editar Ver Buscar Terminal Ayuda
Fase Activa: 0
Movimientos: 1 2
;Desea modificar la fase activa? (Control+C):
█
```

*Ilustración 30. Programa CCT. Fase activa inicial*

En momento en que se analiza por primera vez la densidad de ocupación de los vehículos y la presencia de peatones, la fase activa puede cambiar si se producen las condiciones para ello. Los datos que salen por pantalla se mantienen hasta que se produce un cambio de fase o se cambia la fase activa de modo manual.

```
Archivo Editar Ver Buscar Terminal Ayuda
Fase Activa: 1
Movimientos: 3 4
;Desea modificar la fase activa? (Control+C):
█
```

*Ilustración 31. Programa CCT. Fase activa*

Si el operario desea cambiar la fase activa, debe pulsar ctrl-c. En ese instante, el programa se detiene y solicita la fase que se requiere activar.

```
Archivo Editar Ver Buscar Terminal Ayuda
Fase Manual Activada
Introduzca la fase a activar: 2█
```

*Ilustración 32. Programa CCT. Cambio a fase manual*

Una vez introducida, se muestra por pantalla la fase activa, los movimientos que posee dicha fase y un recordatorio de que el control de la intersección está en modo manual.



```

Archivo Editar Ver Buscar Terminal Ayuda
Fase Manual Activada
Introduzca la fase a activar: 2
Fase Activa: 2
Movimientos: 5 6
Pulse Control+C para desactivar fase manual

```

*Ilustración 33. Programa CCT. Fase manual*

En el momento en que el operario considere retornar al modo automático, pulsa de nuevo ctrl-c y el proceso SCT a partir de los datos que recibe de los procesos que tratan las cámaras de videovigilancia decide si mantiene activa la fase impuesta por el operario o la modifica. En el caso de que el operario decidiese cambiar la fase activa si el sistema está en modo manual, este deberá retornar al modo automático y volver a pulsar ctrl-c para entrar de nuevo en el modo manual. Una vez que se abandona el modo manual, en pantalla aparecen el mismo tipo de datos que los establecidos para su inicio de ejecución continua.

```

Archivo Editar Ver Buscar Terminal Ayuda
Fase Manual Desactivada
Fase Activa: 2
Movimientos: 5 6
¿Desea modificar la fase activa? (Control+C):

```

*Ilustración 34. Programa CCT. Fase manual desactivada*

En el caso de que la intersección solo posea una fase, el sistema no permitirá la introducción de ninguna fase por teclado y se muestra por pantalla si la fase está o no activa.

```

Archivo Editar Ver Buscar Terminal Ayuda
Fase Activa
Movimientos: 1
¿Desea modificar la fase activa? (Control+C):

```

*Ilustración 35. Programa CCT. Modo automático activo para una fase*



Si la fase está activa y el operario pulsa ctrl-c, dicha fase se desactivará y por pantalla saldrá el mensaje de “Fase desactivada”. Si la fase está desactivada y el operario pulsa las teclas mencionadas, la fase se activará y saldrá por pantalla el mensaje “Fase activada”. Tanto si se produce un caso como otro, se muestra en la pantalla el recordatorio de que el sistema se encuentra funcionando en modo manual.

```
Archivo Editar Ver Buscar Terminal Ayuda
Fase Manual Activada
Fase Desactivada
Movimientos: 1
Pulse Control+C para desactivar fase manual
█
```

*Ilustración 36. Programa CCT. Modo manual activo para una fase*

```
Archivo Editar Ver Buscar Terminal Ayuda
Fase Manual Desactivada
Fase Desactivada
Movimientos: 1
¿Desea modificar la fase activa? (Control+C):
█
```

*Ilustración 37. Programa CCT. Modo manual desactivo para una fase*

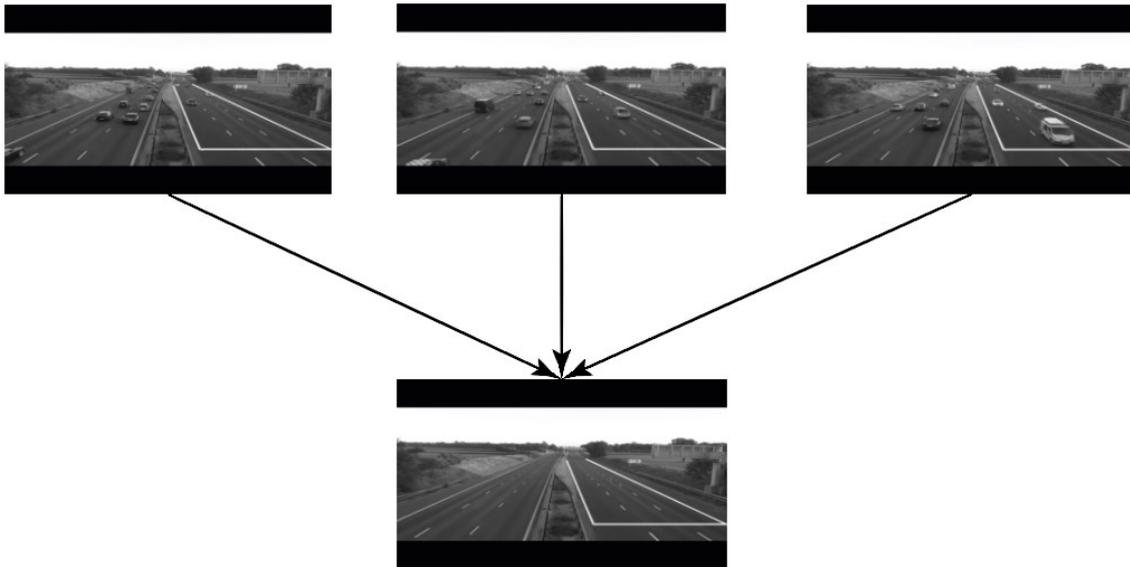
#### 4.4 Programa SIV

El programa SIV, sistema inteligente de vehículos, se encarga de interpretar la secuencia de imágenes de las cámaras que adquieren las características del tráfico. En la puesta en marcha del proceso, se adquiere la región de interés y el área total de dicha región.

El área de análisis se obtiene a partir de una imagen base. Dicha imagen se caracteriza por obtener los objetos estáticos mediante el estudio de un grupo de imágenes. Para ello, se obtiene la moda entre los píxeles que se posicionan en el mismo lugar. Las imágenes a analizar están en escala de grises ya que para la obtención de las líneas no es necesario que esté en color ya que en escala de grises se mantiene la tonalidad blanca. Además, el análisis de imágenes en escala de grises



disminuye el coste computacional por lo que la duración en obtener la imagen base disminuye.



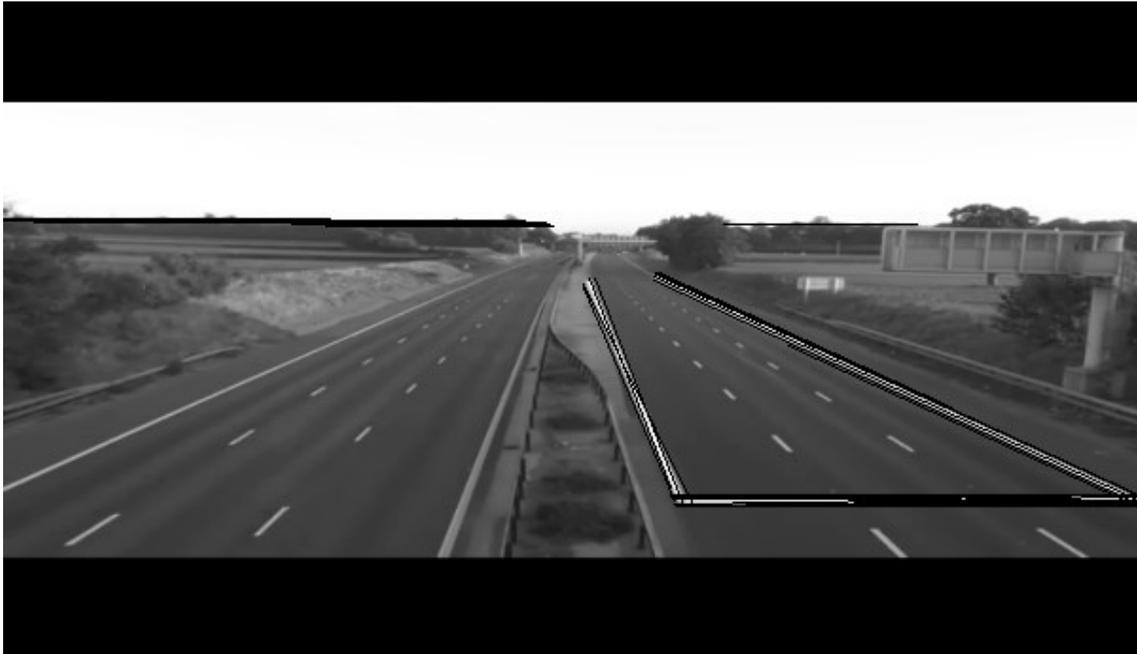
*Ilustración 38. Imagen base obtenida del análisis de un conjunto de frames*

A continuación, se adquieren las líneas de delimitación de carril y la línea de detención. Estas líneas se caracterizan por poseer una tonalidad blanca, por lo que en primer lugar, se obtienen los objetos de la imagen que tienen dicha tonalidad.



*Ilustración 39. Objetos con tonalidad blanca (izquierda). Contornos objetos blancos (derecha)*

Una vez adquiridas las regiones con tonalidad blanca, se obtienen las rectas de la imagen anterior que constituyen los objetos mediante la función HoughLinesP. La imagen contendrá rectas que no corresponden con las líneas de delimitación de carril ni de detención, por lo que se realiza una selección.



*Ilustración 40. Líneas detectas por la función HoughLinesP*



*Ilustración 41. Líneas horizontales seleccionadas*



*Ilustración 42. Líneas verticales seleccionadas*



*Ilustración 43. Líneas verticales y línea horizontal que cumplen las condiciones*

En la imagen anterior, se observa el proceso de selección de las rectas. En primer lugar, se obtienen la totalidad de ellas. A continuación, se separan las líneas verticales de las horizontales para la selección de las líneas de delimitación y la línea de detención respectivamente. Las líneas se obtienen a partir de los contornos de los objetos, por lo que la función HoughLinesP adquirirá más de una recta por cada



línea de la imagen. La elección de la recta que define a una línea se fundamenta en la distancia euclídea, por lo que la que posea mayor distancia se selecciona. Una vez que se obtiene una línea por cada recta, se adquieren las líneas que marcan la región de interés. Para ello, se analiza la distancia euclídea de cada línea horizontal con la totalidad de las líneas verticales y si hay dos rectas verticales que se ubican muy próximas a la recta horizontal se asignan tanto la recta horizontal como las rectas verticales como delimitación de la región de interés. La región de interés se caracteriza por ser un polígono, por lo que se implementa una línea que enlaza los extremos que se encuentran más alejados de las líneas verticales, es decir, de las líneas de delimitación de carril. En la siguiente imagen, se observa la región de interés adquirida automáticamente por el programa.



*Ilustración 44. Área análisis*

Una vez obtenida la región de interés, se obtiene la matriz de corrección de perspectiva. Para ello, se obtiene una ecuación por cada carril que forma el acceso a la intersección, por lo que se obtienen las líneas de delimitación de cada carril.



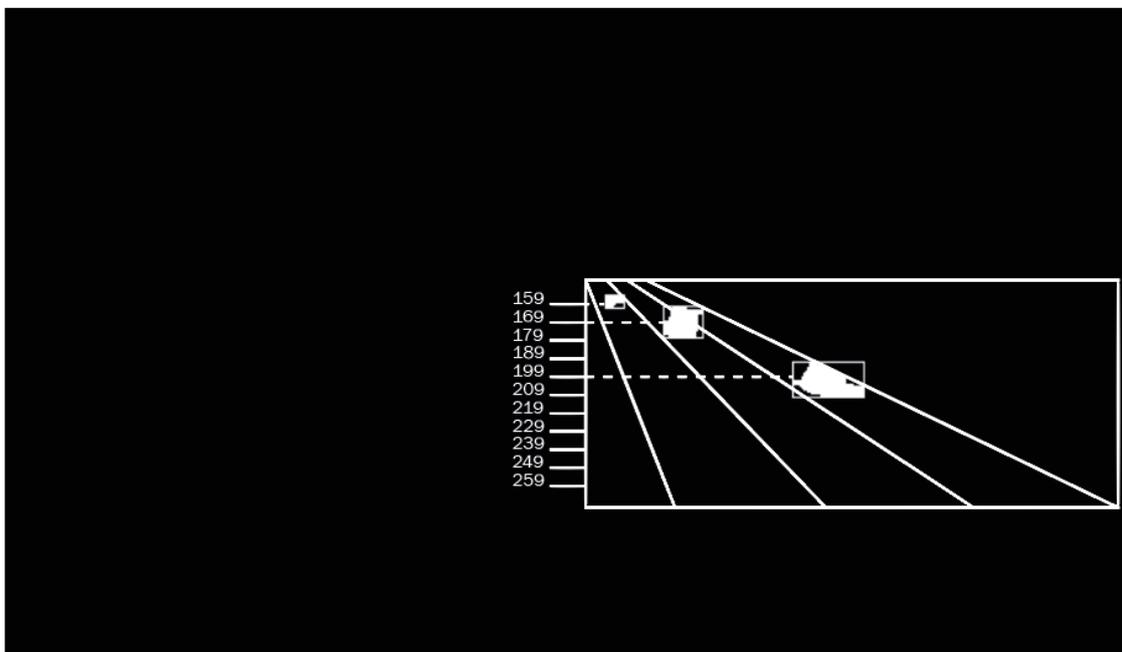
*Ilustración 45. Líneas de delimitación de los carriles*

A continuación, se divide la línea vertical que forma el rectángulo circunscrito en el área de análisis en 10 secciones para adquirir el rango en que se obtienen las áreas según la posición del vehículo. Éstas no se adquieren cerca de las líneas de detención ni de detección ya que el área obtenida sería incorrecto, por lo que se cogen dichas áreas a partir de una distancia superior o igual al 10% tanto del punto superior de la recta vertical como del punto inferior. La dimensión de la altura para este caso se evalúa en 120 píxeles, por lo que si se le resta el 10% en cada extremo, la distancia se tasa en 108 píxeles. Dicho valor, se divide entre las secciones y se obtiene entre los puntos una distancia de 10,8 píxeles, redondeando a la baja, 10 píxeles.



*Ilustración 46. Divisiones línea vertical*

La adquisición de las áreas que ocupan los vehículos en los rangos de los puntos de control, se realiza mediante el método de sustracción de fondo. En el momento en que el vehículo entra en el área de análisis y pasa por el rango establecido, se almacena en un vector de vectores, la coordenada y exacta y el área que ocupa.



*Ilustración 47. Obtención área de vehículo en distintas coordenadas y*



Una vez que se adquiere el área de los vehículos que circulan por diferente carril en todos los puntos y, se obtiene una ecuación de la recta por cada carril y se ajusta con el método de mínimos cuadrados para disminuir el error. Los puntos de la coordenada Y, se establecen como puntos X y el área como puntos Y en la ecuación de la recta.

	X	Y	X*Y	X <sup>2</sup>
	154	48	7392	23716
	162	280	45360	26244
	172	494	84968	29584
	182	684	124488	33124
	192	945	181440	36864
	203	1044	211932	41209
	216	1493	322488	46656
	226	1808	408608	51076
	235	2312	543320	55225
	242	2726	659692	58564
Sumatorio	1984	11834	2589688	402262

Tabla 17. Ajuste por mínimos cuadrados carril izquierdo

$$m = \frac{\sum xy - \frac{(\sum x) \cdot (\sum y)}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}}$$

Ecuación 16. Pendiente de una recta

$$b = \bar{y} - m\bar{x}$$

Ecuación 17. Intersección en y

$$m = \frac{2589688 - \frac{1984 \cdot 11834}{10}}{402262 - \frac{1984^2}{10}} \rightarrow \boxed{m = 28,0003}$$

$$\bar{x} = \frac{\sum x}{n} \rightarrow \bar{x} = 198,4$$

$$\bar{y} = \frac{\sum y}{n} \rightarrow \bar{y} = 1183,4$$

$$\boxed{b = -4371,8735}$$

$$\boxed{y = 28,0003x - 4371,8735}$$

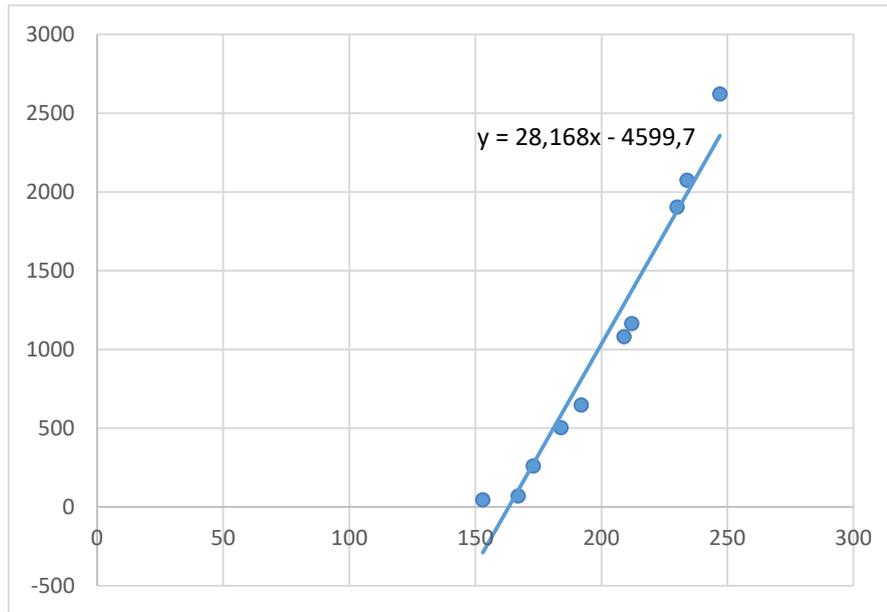


Figura 18. Ecuación de la recta del carril izquierdo

	X	Y	X*Y	X <sup>2</sup>
	153	45	6885	23409
	167	70	11690	27889
	172	78	13416	29584
	182	472	85904	33124
	193	654	126222	37249
	202	858	173316	40804
	215	860	184900	46225
	222	2251	499722	49284
	239	3192	762888	57121
	243	3612	877716	59049
Sumatorio	1988	12092	2742659	403738

Tabla 18. Ajuste por mínimos cuadrados carril central

$$m = \frac{2742659 - \frac{1988 \cdot 12092}{10}}{403738 - \frac{1988^2}{10}} \rightarrow m = 39,7448$$

$$\bar{x} = \frac{\sum x}{n} \rightarrow \bar{x} = 198,8$$

$$\bar{y} = \frac{\sum y}{n} \rightarrow \bar{y} = 1209,2$$



$$b = -6692,0808$$

$$y = 39,7448x - 6692,0808$$

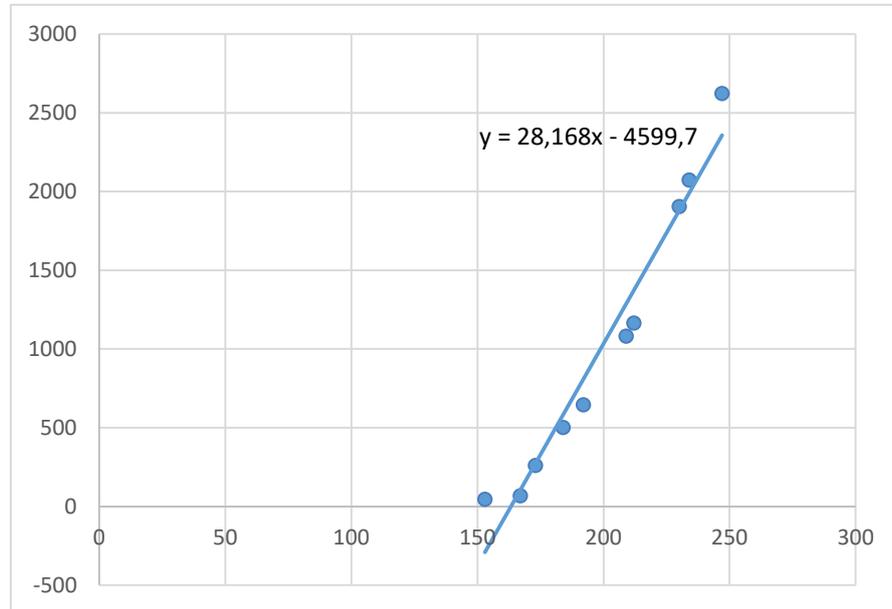


Figura 19. Ecuación de la recta del carril central

	X	Y	X*Y	X <sup>2</sup>
	153	45	6885	23409
	167	69	11523	27889
	173	260	44980	29929
	184	502	92368	33856
	192	646	124032	36864
	209	1081	225929	43681
	212	1164	246768	44944
	230	1904	437920	52900
	234	2074	485316	54756
	247	2622	647634	61009
Sumatorio	2001	10367	2323355	409237

Tabla 19. Ajuste por mínimos cuadrados carril derecho

$$m = \frac{2323355 - \frac{2001 \cdot 10367}{10}}{409237 - \frac{2001^2}{10}} \rightarrow m = 28,1680$$

$$\bar{x} = \frac{\sum x}{n} \rightarrow \bar{x} = 200,1$$



$$\bar{y} = \frac{\sum y}{n} \rightarrow \bar{y} = 1036,7$$

$$b = -4599,7281$$

$$y = 28,1680x - 4599,7281$$

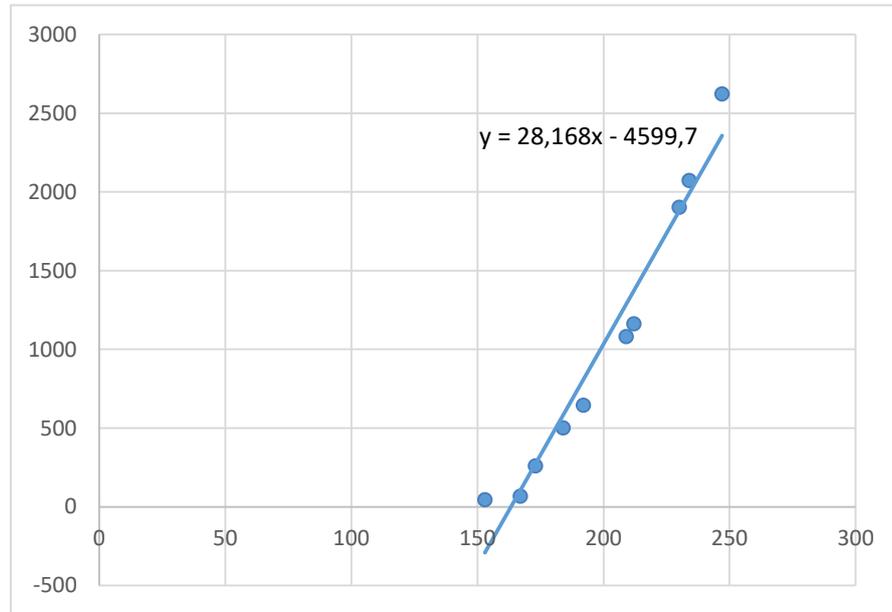


Figura 20. Ecuación de la recta del carril derecho

A partir de las ecuaciones obtenidas, se calcula el área que ocupa cada vehículo en cada punto de la recta vertical que une la línea de detención con la línea de delimitación y se establece la relación para cada fila mediante la división entre el área que ocupa el vehículo en el punto de control más próximo a la línea de detención y el área en cada punto de la recta anterior. A continuación, se obtiene la relación para las cuatro primeras filas.

$$y_1 = 28,0003x_1 - 4371,8735$$

Coordenada Y	Área
270	3188,2075
269	3160,2072
268	3132,2069
267	3104,2066

Tabla 20. Áreas para 4 coordenadas Y carril izquierdo

$$\text{Peso 1}^\circ \text{ línea} = \frac{\text{Área}_0}{\text{Área}_0} \rightarrow \text{Peso 1}^\circ \text{ línea} = \frac{3188,2075}{3188,2075} \rightarrow \text{Peso 1}^\circ \text{ línea} = 1$$



$$\text{Peso 2}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_1} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = \frac{3188,2075}{3160,2072} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = 1,0088$$

$$\text{Peso 3}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_2} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = \frac{3188,2075}{3132,2069} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = 1,0178$$

$$\text{Peso 4}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_3} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = \frac{3188,2075}{3104,2066} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = 1,0270$$

$$y_2 = 39,7448x_2 - 6692,0808$$

Coordenada Y	Área
270	4039,0152
269	3999,2704
268	3959,5256
267	3919,7808

Tabla 21. Áreas para 4 coordenadas Y carril central

$$\text{Peso 1}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_0} \rightarrow \text{Peso 1}^{\text{o}} \text{ línea} = \frac{4039,0152}{4039,0152} \rightarrow \text{Peso 1}^{\text{o}} \text{ línea} = 1$$

$$\text{Peso 2}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_1} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = \frac{4039,0152}{3999,2704} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = 1,0099$$

$$\text{Peso 3}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_2} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = \frac{4039,0152}{3959,5256} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = 1,0200$$

$$\text{Peso 4}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_3} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = \frac{4039,0152}{3919,7808} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = 1,0304$$

$$y_3 = 28,1680x_3 - 4599,7281$$

Coordenada Y	Área
270	3005,6319
269	2977,4639
268	2949,2959
267	2921,1279

Tabla 22. Áreas para 4 coordenadas Y carril derecho



$$\text{Peso 1}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_0} \rightarrow \text{Peso 1}^{\text{o}} \text{ línea} = \frac{3005,6319}{3005,6319} \rightarrow \text{Peso 1}^{\text{o}} \text{ línea} = 1$$

$$\text{Peso 2}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_1} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = \frac{3005,6319}{2977,4639} \rightarrow \text{Peso 2}^{\text{o}} \text{ línea} = 1,0094$$

$$\text{Peso 3}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_2} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = \frac{3005,6319}{2949,2959} \rightarrow \text{Peso 3}^{\text{o}} \text{ línea} = 1,0191$$

$$\text{Peso 4}^{\text{o}} \text{ línea} = \frac{\text{Área}_0}{\text{Área}_3} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = \frac{3005,6319}{2921,1279} \rightarrow \text{Peso 4}^{\text{o}} \text{ línea} = 1,0289$$

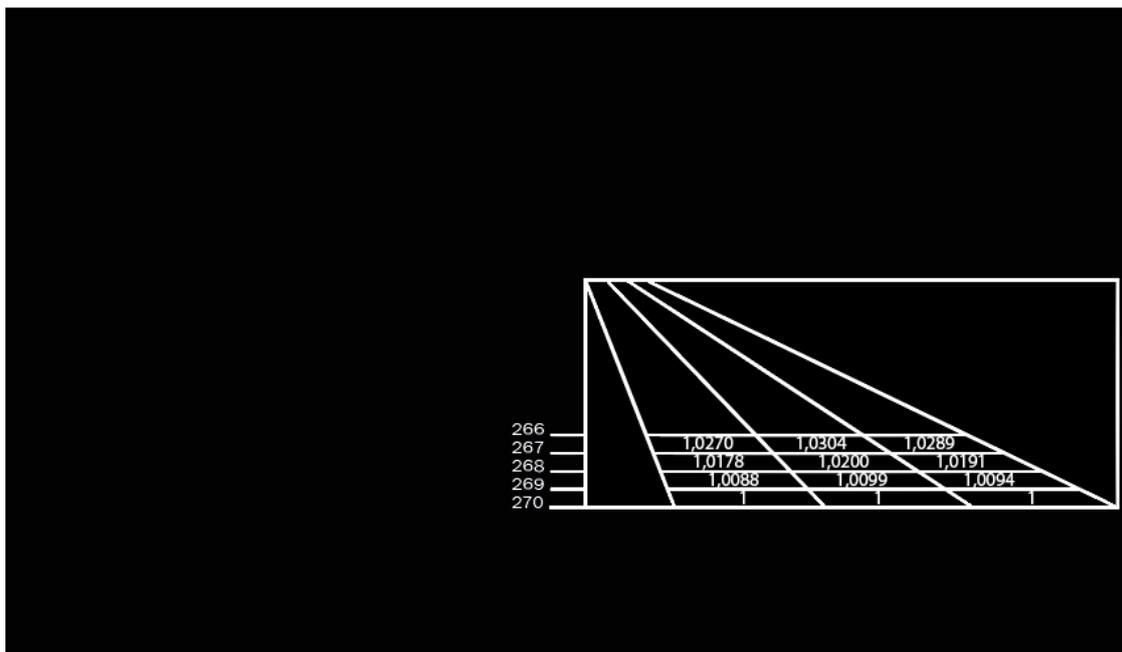
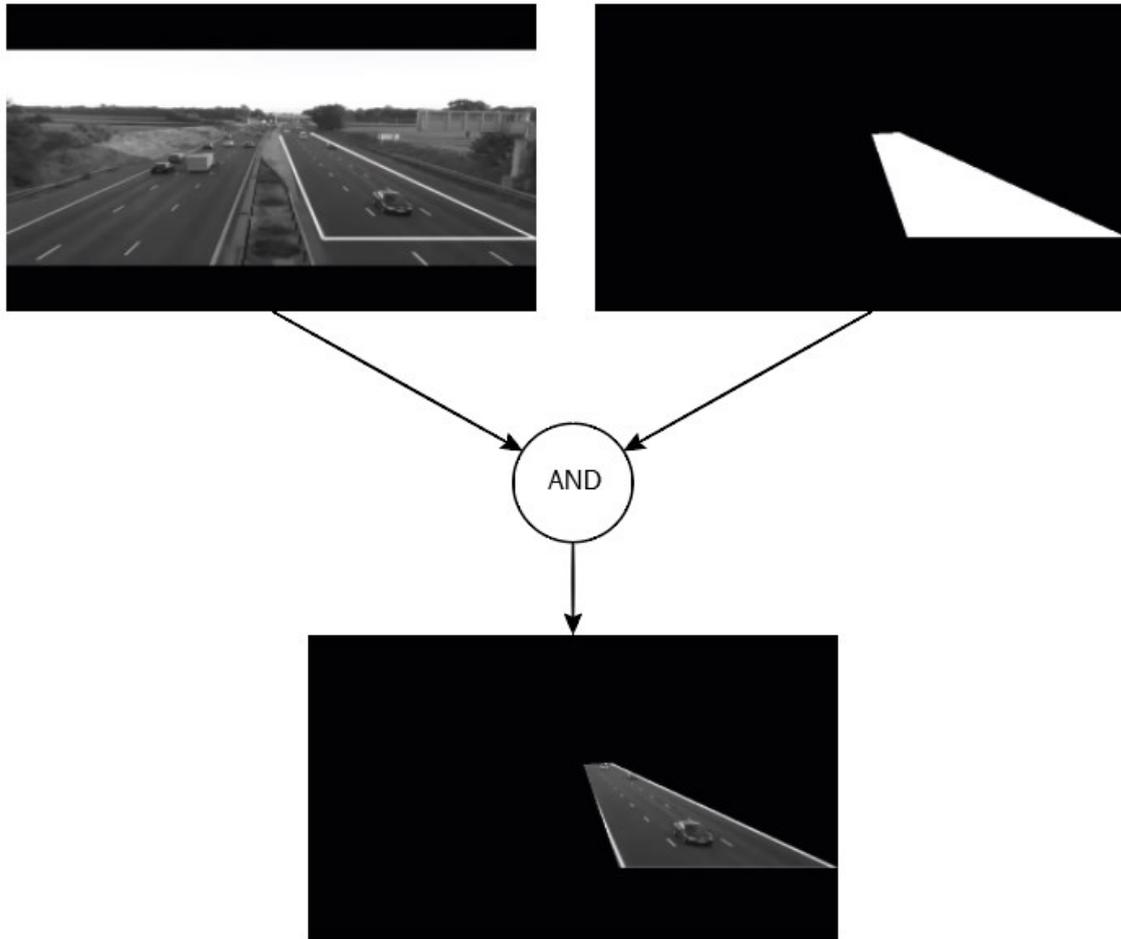


Tabla 23. Peso píxeles por fila

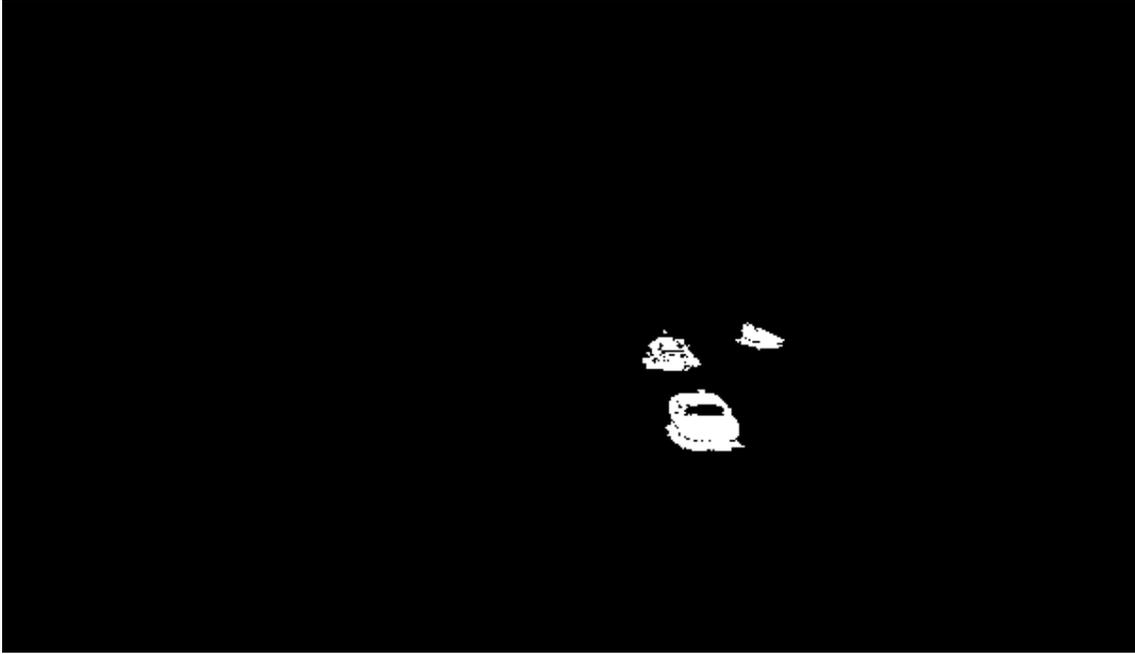
Una vez que se implementan diferentes pesos para cada fila y carril, se realiza el sumatorio de todos los pesos y se obtiene el área total, XXXXX píxeles. A continuación, comienza la ejecución continua.

El proceso SIV obtiene el área ocupada por los vehículos. En primer lugar, se recorta el área a analizar del frame recibido. Para ello, se rellena el polígono y se realiza la operación lógica and entre la imagen que contiene el polígono y el frame. Si el píxel de la imagen del polígono se evalúa en 255, el píxel que se ubica en la misma posición en la imagen de salida adquiere el valor del píxel del frame que se sitúa en la misma ubicación. En caso contrario, el píxel de la imagen de salida se tasa con un valor nulo, color negro. La imagen posee perspectiva por lo que se corrige para que el área de un vehículo sea igual en toda la región de análisis



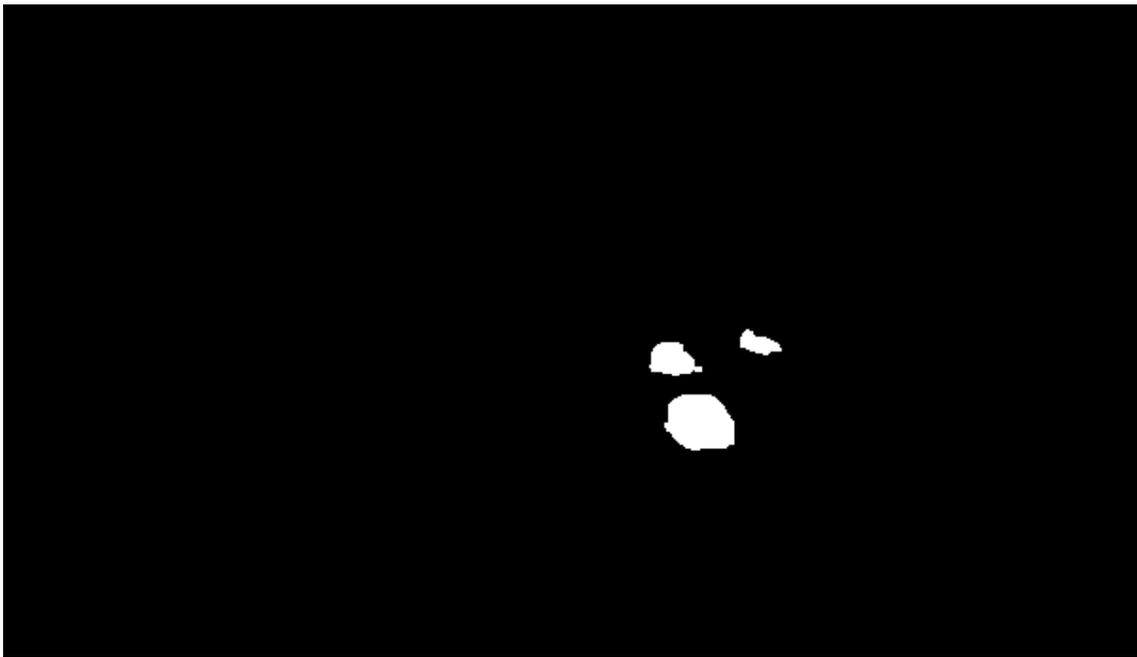
*Ilustración 48. Frame recibido de la cámara de videovigilancia (Imagen superior izquierda). Área de análisis (Imagen superior derecha). Resultado operación lógica AND entre ambas imágenes (Imagen inferior)*

El área ocupada por los vehículos se adquiere mediante el modelo de sustracción de fondo. Este método se caracteriza por restar el frame actual con una imagen modelo. Dicha imagen se obtiene a través de un conjunto de frames que analizan los objetos que se mantienen estáticos.



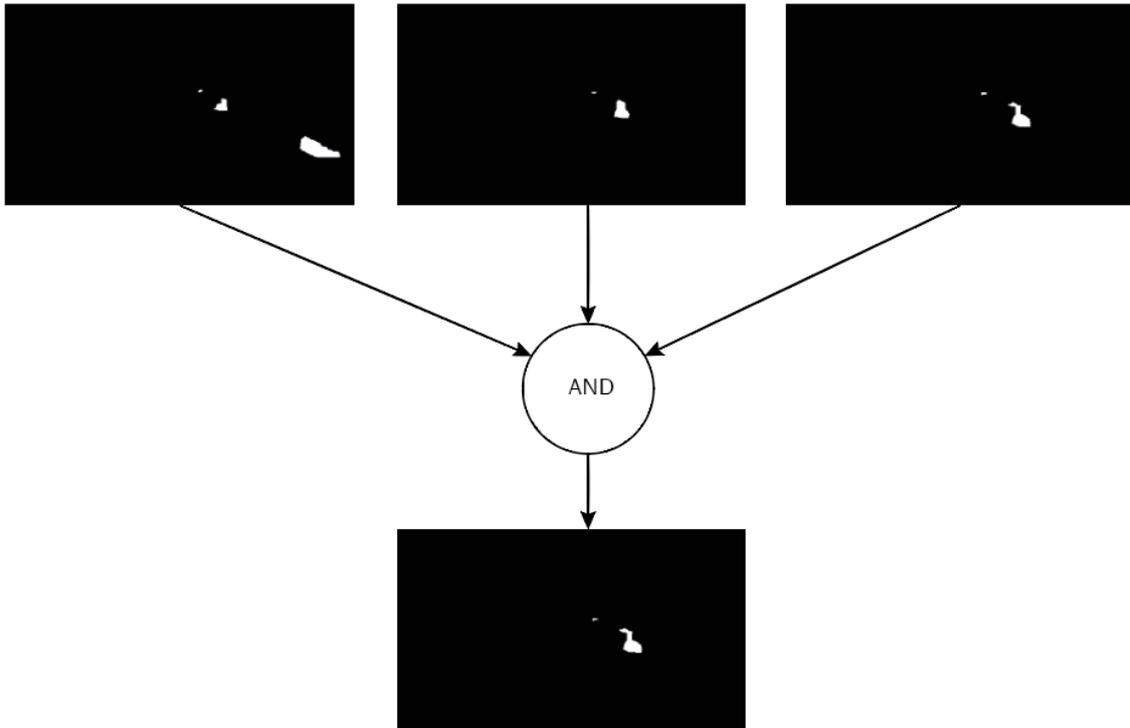
*Ilustración 49. Detección movimiento mediante el método de sustracción de fondo*

El método empleado, tal y como se observa en la imagen anterior, no detecta el movimiento de determinadas zonas de los vehículos por poseer los píxeles del frame y los del modelo unos valores semejantes. Para que dichas zonas el programa las adquiera en movimiento, se emplean operaciones morfológicas, primero una apertura y después un cierre. El elemento estructural del cierre no es elevado ya que provocaría la unión de vehículos., por lo que para detectar zonas en movimiento que poseen un área elevada, se adquieren los contornos de dichas áreas y se rellena. El resultado de aplicar las acciones mencionadas, se expone en la siguiente imagen.



*Ilustración 50. Detección movimiento mediante el método de sustracción de fondo aplicando operaciones morfológicas*

Si el semáforo está en rojo, el sistema debe adquirir a los vehículos antes de que el método de sustracción de fondo las adquiriera como modelo. Para ello, se almacenan cada 0,25 segundos imágenes en un vector y a los 4 segundos que es el tiempo en que dicho método las absorbe, se analizan. La detección de vehículos detenidos se realiza aplicando la operación and a las imágenes almacenadas.



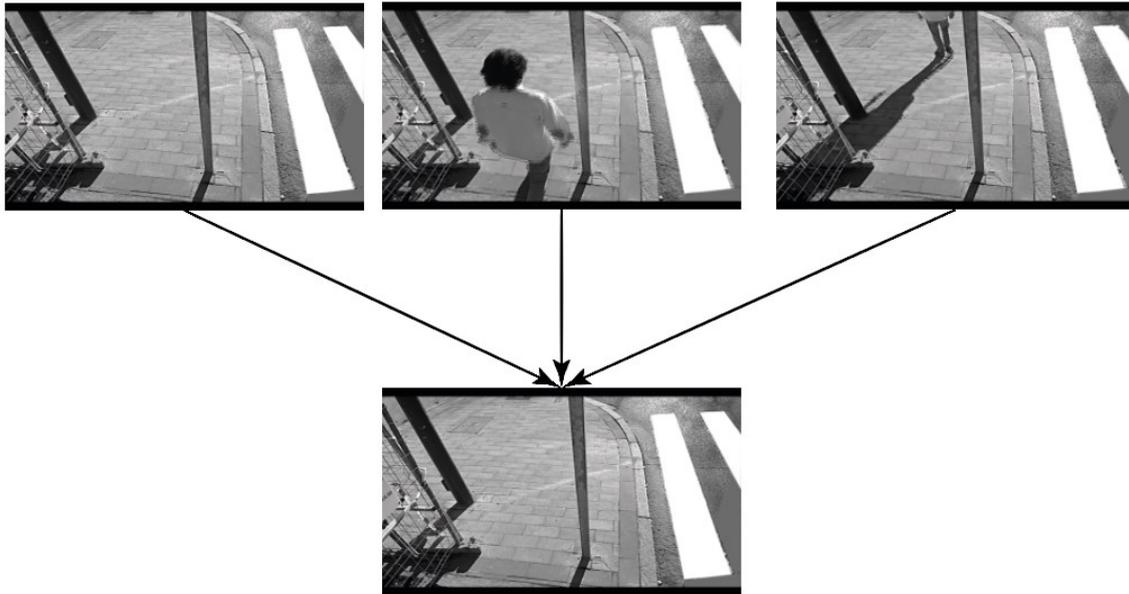
*Ilustración 51. Imagen estática resultado de aplicar la operación lógica AND entre varias imágenes*

Por último, se evalúa el área ocupada. Para ello se recorre la matriz que contiene la imagen de los vehículos y si un píxel de dicha imagen posee el valor 255, color blanco, se adquiere el peso de dicho píxel que se ubica en la misma posición en la matriz de perspectiva. El área ocupada es el sumatorio de todos los pesos que se obtienen de dicha matriz. Este dato junto con las áreas ocupadas de los procesos que tratan las imágenes de los peatones se envían al proceso SCT para su análisis.

#### 4.5 PROGRAMA SIP

El programa SIP, sistema inteligente de peatones, se encarga del tratamiento de la secuencia de imágenes que recibe de las cámaras de videovigilancia que enfocan a la zona de espera. En la puesta en marcha se obtiene el área de análisis y en su ejecución continua el área ocupada.

La adquisición del área de análisis se realiza a partir de una imagen base, sin peatones. Para ello, se almacena una secuencia de imágenes y se obtiene la imagen base a partir de los píxeles que más se repiten, es decir, se calcula la moda de los píxeles que se ubican en la misma posición y se asigna dicho píxel a una imagen nueva.



*Ilustración 52. Imagen base obtenida del análisis de un conjunto de frames del paso de peatones*

A partir de la imagen base se adquiere el rectángulo del paso de cebrá más cercano a la zona de espera. Dichas líneas se caracterizan por poseer una tonalidad blanca elevada, por lo que se realiza la segmentación de la imagen según el color mencionado y se obtienen los contornos de las siluetas.



*Ilustración 53. Segmentación por color blanco (izquierda). Contornos imagen segmentada (derecha)*

A continuación, se consiguen las líneas rectas mediante la función `HoughLinesP`. Esta función obtiene la totalidad de las líneas de la imagen, por lo que se realiza una clasificación de dichas líneas a partir de las condiciones descritas en la descripción del programa SIV.



*Ilustración 54. Líneas obtenidas por la función HoughLinesP*



*Ilustración 55. Líneas verticales seleccionadas*



*Ilustración 56. Líneas horizontales seleccionadas*



*Ilustración 57. Líneas verticales y horizontales que cumplen las condiciones establecidas*

Una vez que se adquieren las líneas que delimitan el paso de cebra más cercano a la calzada, se obtiene la línea que se ubica más al exterior, es decir, la más próxima a la zona de espera para fijar el área de análisis. En el caso de que la cámara enfoque al lado izquierdo se adquiere la línea que posee la coordenada  $x$  menor y si la cámara enfoca al lado derecho, se obtiene la línea que detenta un coordenada  $x$  mayor.



*Ilustración 58. Línea vertical más próxima a la zona de espera obtenida por una cámara situada a la izquierda*



*Ilustración 59. Línea vertical más próxima a la zona de espera obtenida por una cámara situada a la derecha*

A partir de la línea exterior, se obtiene el área de análisis a una distancia de 0,5 metros con respecto a dicha línea. La región de interés se caracteriza por ser un polígono cerrado de cuatro lados. El primer lado lo forma una línea paralela a la recta exterior a la distancia antes mencionada, el segundo y el tercer lado, lo componen dos líneas paralelas que surgen de los extremos de la recta paralela a la línea exterior



con una longitud de 1,5 metros, y por último, la recta que cierra el polígono une los extremos del par de líneas paralelas.

La distancia en una imagen se mide en píxeles, por lo que se realiza la transformación a metros mediante el ángulo del objetivo de la cámara de videovigilancia, la altura a la que se sitúa dicha cámara, y el número de líneas horizontales y verticales que constituyen la imagen. Las cámaras que se instalan se caracterizan por poseer un ángulo de  $100^\circ$  y por situarse a una altura de 2,4 metros. El número de líneas horizontales y verticales se tasan en 350 y en 600, respectivamente.

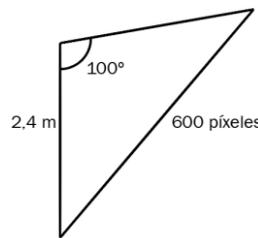


Figura 21. Píxeles captados por la cámara situada a 2,4 metros

El número de píxeles que corresponden a una distancia de 2 metros en horizontal se obtiene implementando un triángulo cuya base surge desde el suelo y se desplaza la longitud descrita y la hipotenusa se establece desde la ubicación de la cámara hasta el extremo de la base, tal y como se muestra en la siguiente figura.

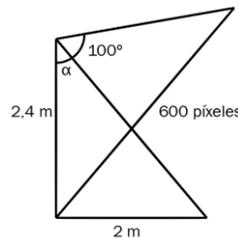


Figura 22. Triángulo que permite obtener el número de píxeles para 2 metros de longitud

A continuación, se evalúa el ángulo  $\alpha$  a partir de la fórmula trigonométrica correspondiente a la tangente y se calculan los píxeles que corresponden a 2 metros. El número de píxeles que equivalen a la distancia mencionada es la sumatoria de los píxeles que se ubican en el segmento de línea que corta a la hipotenusa del triángulo y finaliza en el vértice en que confluyen los dos catetos. Esta cantidad es equivalente a los píxeles que se posicionan en la línea horizontal del suelo.

$$\tan \alpha = \frac{c. o.}{c. c.} \rightarrow \alpha = \tan^{-1} \frac{c. o.}{c. c.} \rightarrow \alpha = \tan^{-1} \frac{2}{2,4} \rightarrow \boxed{\alpha = 39,8055^\circ}$$



$$\text{Número píxeles} = \frac{N^{\circ} \text{ píxeles en horizontal} \cdot \text{Ángulo}}{\text{Ángulo cámara}}$$

$$\text{Número píxeles} = \frac{600 \cdot 39,80055}{100} \rightarrow \boxed{\text{Número píxeles} = 238,8 \text{ píxeles}}$$

La distancia en que se ubican el área de análisis de los peatones respecto de la recta exterior, se tasa en 0,5 metros, y la distancia de las rectas horizontales que surgen de la primera recta del polígono, se evalúan en 1,5 metros, por lo que se obtiene la distancias mencionadas en píxeles a partir del número de píxeles evaluados para 2 metros.

$$N^{\circ} \text{ píxeles} = \frac{\text{Número píxeles totales} * \text{longitud línea}}{\text{distancia total}}$$

$$N^{\circ} \text{ píxeles (0,5 metros)} = \frac{238,8 \cdot 0,5}{2} \rightarrow N^{\circ} \text{ píxeles (0,5 metros)} = 59,7 \text{ píxeles}$$

$$N^{\circ} \text{ píxeles (1,5 metros)} = \frac{238,8 \cdot 1,5}{2} \rightarrow N^{\circ} \text{ píxeles (1,5 metros)} = 179,1 \text{ píxeles}$$

$$\boxed{N^{\circ} \text{ píxeles (0,5 metros)} \cong 60 \text{ píxeles}}$$

$$\boxed{N^{\circ} \text{ píxeles (1,5 metros)} \cong 179 \text{ píxeles}}$$

Una vez evaluadas las distancias en píxeles, se adquieren los vértices del polígono. El proceso que controla la cámara que se sitúa en el lado derecho, obtiene el vértice superior izquierdo sumado la coordenada x de la recta exterior 60 píxeles, el vértice superior derecho incrementado la coordenada x del vértice anterior 179 píxeles, el vértice inferior derecho, asignando al vértice anterior la coordenada y superior de la línea exterior, y el vértice inferior izquierdo, asignando al vértice superior izquierdo, la coordenada y de la recta exterior superior. El proceso que trata la cámara que se sitúa en el lado izquierdo, obtiene los vértices de la misma forma que el proceso que controla la cámara del lado derecho pero en vez de realizar la suma de las coordenadas, realiza la resta de ellas. En las siguientes imágenes, se observa el área de análisis de la cámara que enfoca a la derecha y de la cámara que enfoca a la izquierda.



*Ilustración 60. Área análisis peatones cámara derecha*

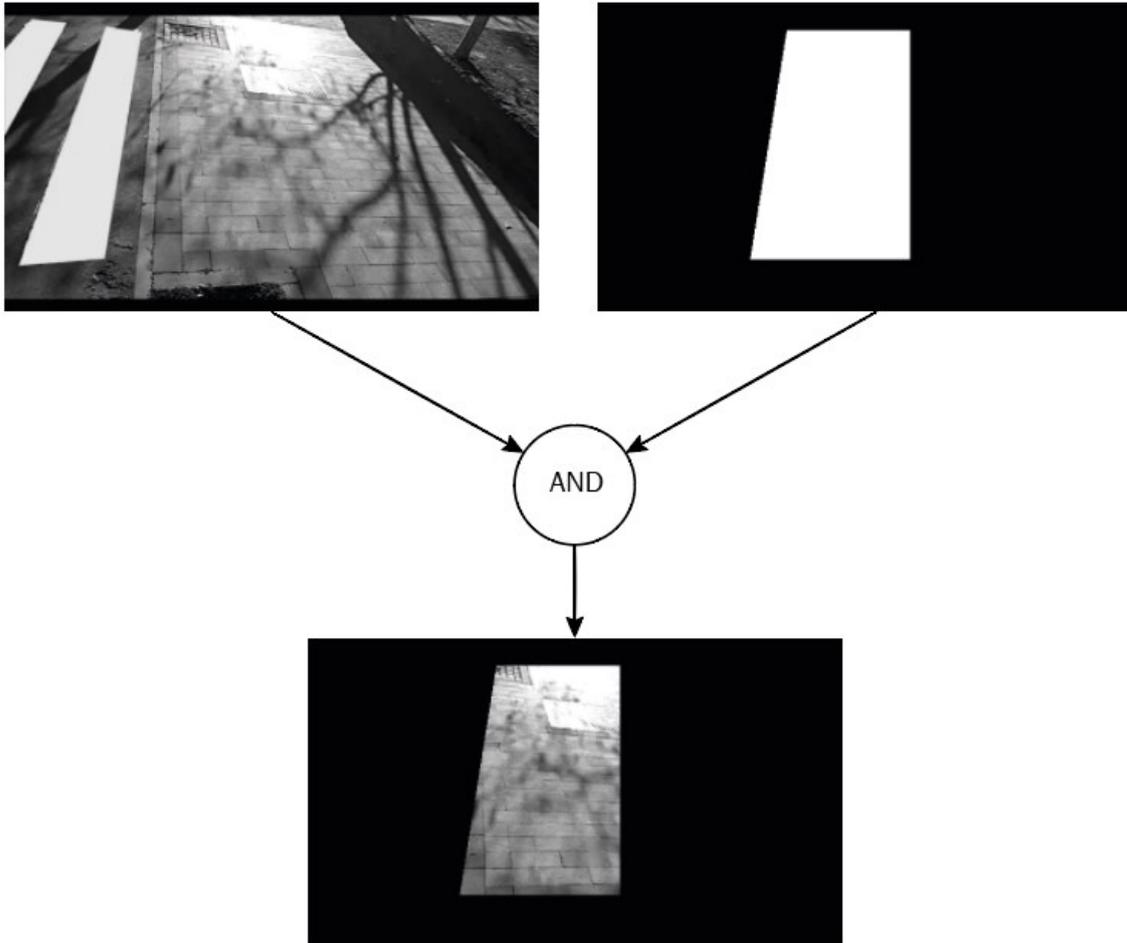


*Ilustración 61. Área análisis peatones cámara izquierda*

Acto seguido, se realiza el sumatorio de los píxeles que se ubican en el interior de la región de interés para obtener el área total y comienza la ejecución continua del programa. En primer lugar, se recorta el área de análisis del frame que se recibe de la cámara de videovigilancia que enfoca a la zona de espera de los peatones. Para ello, se realiza la operación lógica AND entre el frame y el polígono relleno. Si el píxel

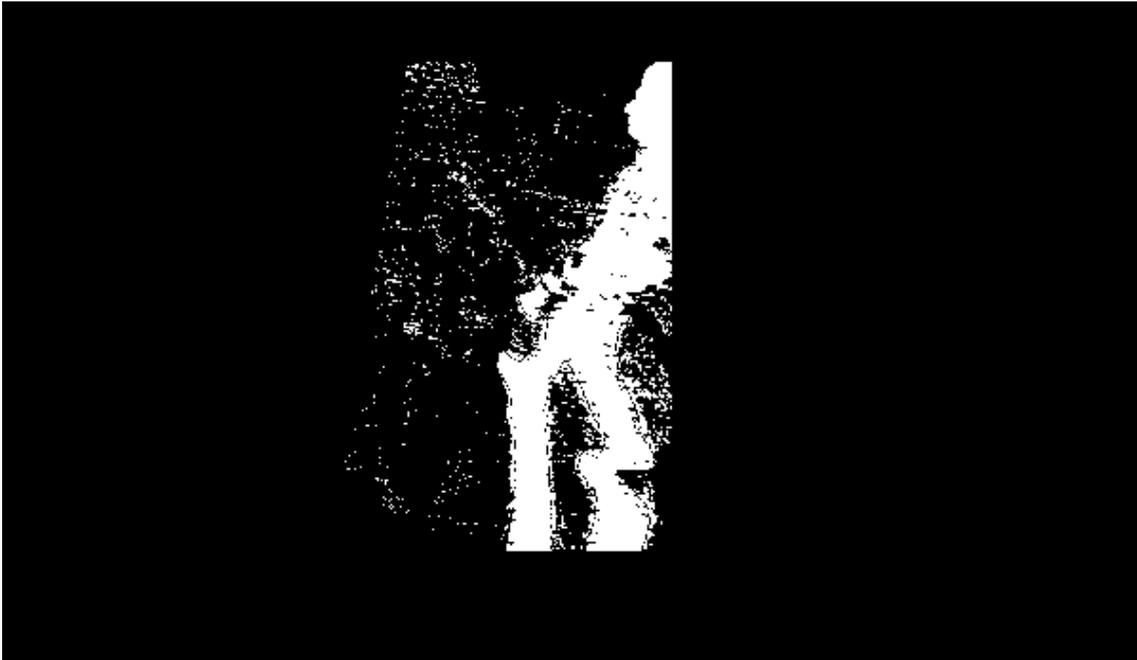


a analizar en el polígono se evalúa en 255, color blanco, el valor del píxel que se posiciona en la misma ubicación se asigna a una nueva imagen. En el caso de que el píxel del polígono se evalúe en 0, color negro, se asigna al píxel de la imagen resultante el valor 0.



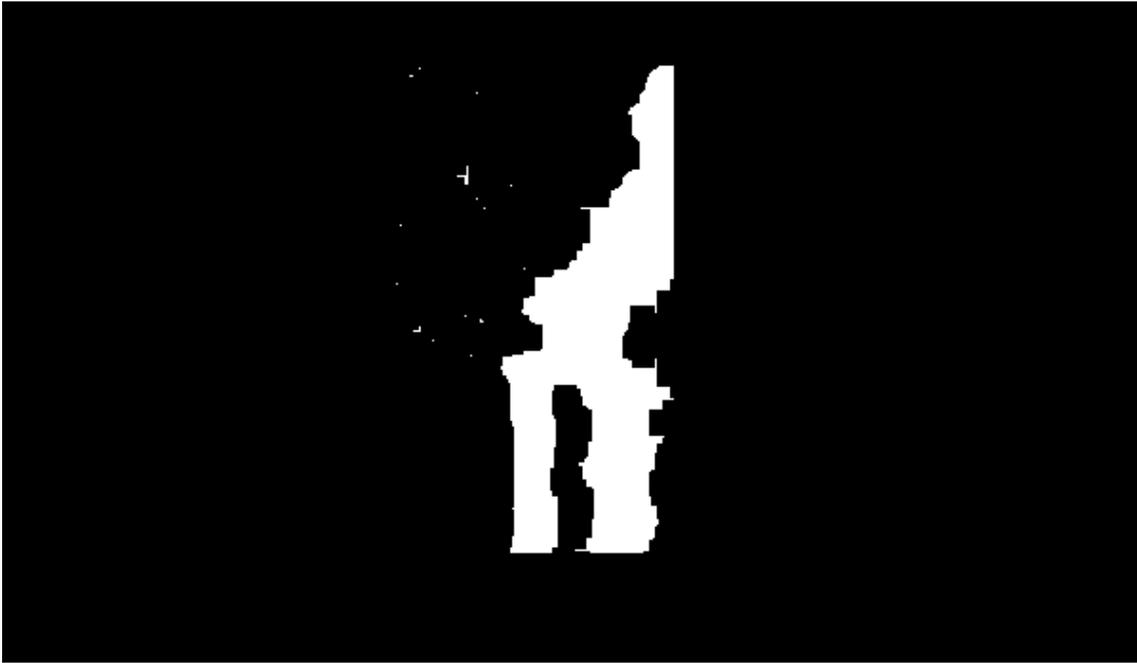
*Ilustración 62. Frame (superior izquierda). Polígono relleno área análisis (superior derecha). Resultado operación lógica AND (inferior)*

Los peatones que se ubican en el área de análisis se detectan mediante el método de sustracción de fondo. Este método efectúa la resta entre el frame actual y el frame modelo. El frame modelo posee los objetos estáticos de la imagen y su obtención se basa en la comparación de un conjunto de frames.



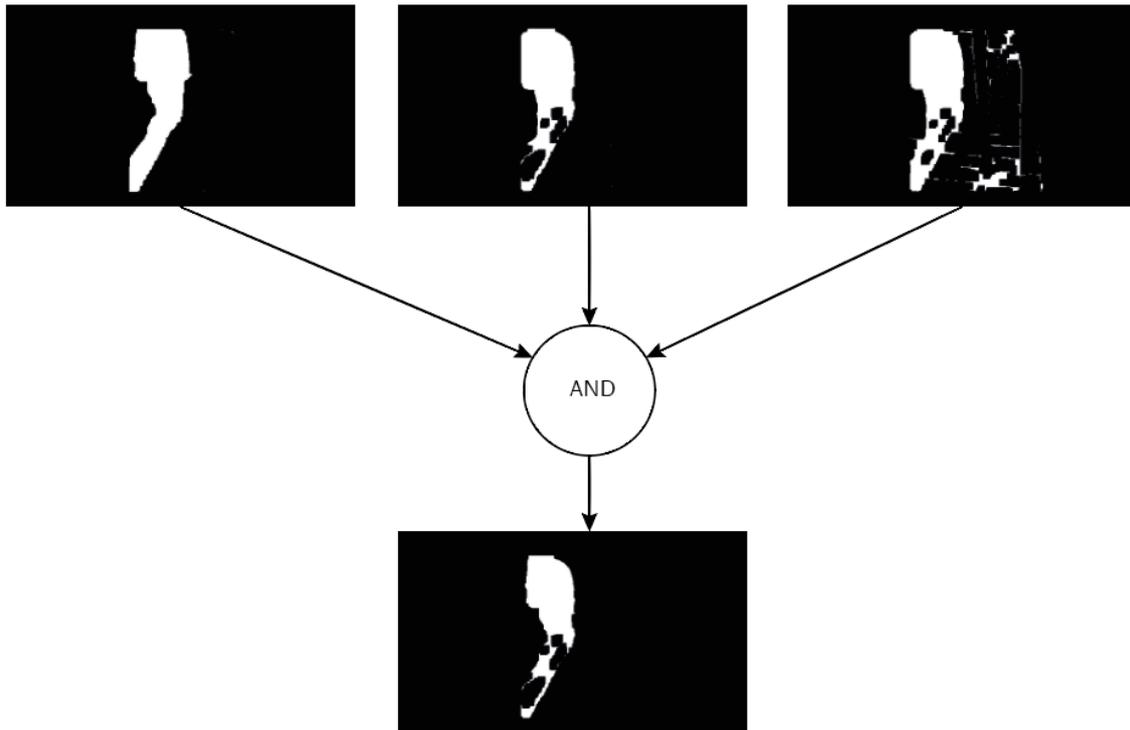
*Ilustración 63. Detección movimiento mediante el método de sustracción de fondo*

Los píxeles en movimiento del frame que se capta de la cámara de videovigilancia pueden tener un brillo muy próximo a los píxeles que forman la imagen modelo, por lo que el sistema les reconocerá como estáticos. Para que el sistema les reconozca como si estuviesen en movimiento, se aplican operaciones morfológicas, en primer lugar, se implementa un filtro de erosión para eliminar el ruido y a continuación un filtro de cierre para que los píxeles que se ubican en el interior de los peatones adquieran el valor 255, color blanco.



*Ilustración 64. Detección movimiento mediante el método de sustracción de fondo aplicando operaciones morfológicas*

Si el semáforo para los peatones está en rojo, el método de sustracción de fondo les absorberá, por lo que se captan antes de que esto ocurra. El método mencionado establece a los peatones como fondo en 4 segundos, por lo que se almacena cada 0,25 segundos los frames que se obtienen de la cámara y se realiza la operación lógica AND entre ellos. Esta operación efectúa la comparación entre los píxeles que se ubican en la misma posición de las imágenes guardadas y si en todas ellas, el píxel se evalúa en 255, color blanco, dicho píxel corresponde a un objeto estático por lo que en la imagen estática se le asigna al píxel que se ubica en la posición de análisis el valor 255. En el caso de que haya una imagen en que dicho píxel se tase en 0, se asignará a la imagen estática el valor 0 ya que pertenecerá a un objeto en movimiento.



*Ilustración 65. Imagen estática resultado de aplicar la operación lógica AND entre varias imágenes de peatones*

Si el semáforo de los peatones está en verde y estos no pueden cruzar por haber vehículos impidiéndolo, el sistema sigue empleando la detección de objetos estáticos hasta que el número de píxeles en movimiento supera el 50% de los que posee la imagen estática. El área ocupada, tanto si los peatones están en estático o en movimiento se evalúa mediante el sumatorio de los píxeles blancos, es decir, los que se tasan en 255. Este dato, se envía al proceso SIV que le ha abierto para su posterior análisis en el programa SCT.



# CAPÍTULO 5

## CONCLUSIONES





## 5 CONCLUSIONES

El proyecto desarrollado analiza el flujo de vehículos y la presencia de peatones para evaluar de forma variable el tiempo que están activas las fases. De esta forma, se disminuyen las retenciones, y por tanto, la contaminación ambiental, y el estrés de los conductores y los peatones.

A partir de los resultados experimentales obtenidos en el apartado anterior, se analizan los diferentes objetivos expuestos al principio de este proyecto. En primer lugar, se realizó un estudio de la situación actual de la tecnología, analizando los sistemas de gestión de tráfico que actualmente se emplean en las ciudades y los métodos existentes para la detección del flujo vehicular y peatonal. A continuación, se efectuó un análisis de los componentes hardware que se encuentran actualmente en el mercado para que el proyecto fuese lo más rentable posible. Se desarrolló un programa capaz de adquirir de forma automática la región de análisis de los vehículos y la densidad de ocupación tanto si estos están en movimiento como detenidos. Además, se implementó otro programa para captar la región de análisis de los peatones y la detección de ellos. La verificación de los programas anteriores se realizó mediante imágenes y vídeos adquiridos de internet a excepción de un vídeo propio realizado para la detección de peatones ya que en internet no había vídeos de pasos peatonales captados desde una determinada altura y con un cierto ángulo.

Para el control de la secuencia de fases en una intersección se realizó un programa que recibe las densidades de ocupación y la presencia de peatones de todos los movimientos principales y calcula el tiempo óptimo y máximo para garantizar las mínimas detenciones y el mínimo tiempo de espera. También, se establecieron las condiciones que se deben cumplir para que se realice el cambio de fase en el momento adecuado. La verificación de este programa se realizó con potenciómetros conectados a un Arduino que simulaban la densidad de ocupación vehicular, con interruptores conectados a las GPIO de la Raspberry Pi que reproducían la presencia o no de peatones y diodos LED acoplados a las GPIO de la Raspberry Pi, que mostraban la secuencia de fases de los semáforos de una intersección.

La información de la fase activa, se remite a un centro de control en el que se puede modificar desde él dicha fase si el operario lo creyese conveniente.

Los datos referentes a la fase activa, a las densidades de ocupación de los vehículos y a la presencia de peatones, se envían a una plataforma web, IOT, para conocer desde cualquier sitio desde el que se disponga conexión a internet, el estado del tráfico y la frecuencia con el que se realiza el cambio de fase.



## 5.1 TRABAJO FUTURO

Una vez desarrollado el proyecto, se exponen a continuación mejoras que se pueden realizar para aumentar la funcionalidad del mismo.

Un sistema inteligente de tráfico instalado en una ciudad debe asegurar que todos los semáforos que se ubican en la misma vía principal tienen que estar en verde circulando a la velocidad máxima permitida. A este hecho se le denomina “Onda Verde”. Debido a ello, un estudio del caso descrito permitiría el despeje de los vehículos que se encuentran en la vía principal disminuyendo así las detenciones por semáforos en rojo que se ubican en intersecciones posteriores.

La aplicación Maps de Google obtiene la ruta más rápida para el desplazamiento en coche desde un punto de partida hasta un destino, por lo que si se analiza la densidad de ocupación vehicular a partir de los datos que envía cada Raspberry Pi instalada en cada intersección, la aplicación Maps indicaría una ruta diferente en el caso de que la más rápida estuviera saturada.

Los programas que constituyen el sistema se ejecutan a través de terminal, por lo que sería aconsejable desarrollar una interfaz gráfica para que las personas que no estén acostumbradas a trabajar con él puedan utilizar el sistema sin ninguna dificultad.

Un mismo carril puede permitir seguir recto en una intersección o realizar un giro a la derecha o a la izquierda, por lo que sería conveniente realizar un estudio para distinguir los casos mencionados y así mejorar los tiempos óptimos de las fases.



# CAPÍTULO 6

## BIBLIOGRAFÍA





## 6. BIBLIOGRAFÍA

1. Fageda X, Flores-Fillol R. Fedea Policy Papers -2018/04 Atascos y contaminación en grandes ciudades: Análisis y soluciones. 2018;
2. European Environment Agency. Air quality in Europe – 2017 report No 13/2017. EEA Technical Report. 2017. 80 p.
3. Federico I, Frisius C. Guía para el usuario de transyt.
4. Hunt PB, Robertson DI, Bretherton RD, Winton RI. SCOOT-a traffic responsive method of coordinating signals. TRRL Lab Rep 1014. 1981;1-45.
5. Sims A. the Sydney Co-Ordinated Adaptive Traffic (Scat) System. Publ Aust Road Res Board. 1979;V(May):130-7.
6. Mirchandani P, Head L. RHODES: A Real-Time Traffic Signal Control System. Archit Algorithms Anal. 1998;1-15.
7. Bances M, Ramos M. Semáforos Inteligentes para la regulación del tráfico vehicular Rev. Rev Ing Ciencia, Tecnol e Innovación. 2014;1(1):1.
8. Hejun W, Changyun M, Ji W, Jianxiong L. Research of intelligent traffic light control scheme based on fuzzy control. Proc - 2011 Int Symp Comput Sci Soc ISCCS 2011. 2011;111-3.
9. Pino Díaz S. Tema 24. Regularización Semafórica. Objeto. Tipos de semáforos. En 2011. p. 1-18.
10. Webster F V. Traffic signal settings. Technical Report Road Research Technical Paper No. 39. 1958.
11. Areny RP. Sensores y Acondicionadores de Señal. 2004. p. 500. ISBN: 8426713440.
12. XUK0ARCTL2 | Sensor Fotoeléctrico Difuso, retrorreflectivo, haz pasante (receptor), LED Infrarrojo, Alcance 0,28 m; 0,8 m; 4 m; 30 m | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/sensores-fotoelectricos/4554983/?sra=pmpn>
13. XUZC50 | Reflector Telemecanique Sensors, para Serie XU | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/accesorios-para-sensores-fotoelectricos/2811751/>
14. XUK0ARCTL2T | Sensor Fotoeléctrico Haz pasante (emisor), LED Infrarrojo, Alcance 30 m, Cuerpo Compacto, Cable de 3 hilos de 2 m, IP65 | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/sensores-fotoelectricos/4554999/?relevancy->



data=636F3D3226696E3D4931384E4B6E6F776E41734D504E266C753D6573266D6D3D6D61746368616C6C7061727469616C26706D3D5E5B5C707B4C7D5C707B4E647D2D2C2F255C2E5D2B2426706F3D313326736E3D592673723D

15. XUB0APSNM12 | Sensor Fotoeléctrico Difuso, retrorreflectivo, haz pasante (receptor), LED Infrarrojo, Alcance 0,12 m; 0,3 m; 2 m; 15 m | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/sensores-fotoelectricos/4443401/?sra=pmpn>
16. XUB0BKSNM12T | Sensor Fotoeléctrico Haz pasante (emisor), LED Infrarrojo, Alcance 15 m, Cuerpo Cilíndrico, Conector M12 4 Pines | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/sensores-fotoelectricos/4443473/?sra=pmpn>
17. XUK1ARCNL2 | Sensor Fotoeléctrico Retrorreflexivo, LED Infrarrojo, Alcance 7 m, Cuerpo Compacto, Salida Relé | RS Components [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/sensores-fotoelectricos/5124640/?sra=pmpn>
18. Detector lazo ldp1sa1b230 (ld-100 220v) - Detectores lazo magnético - Sensórica - Automatización y control industrial [Internet]. [citado 3 de marzo de 2019]. Disponible en: [https://www.centraelectroventas.com/automatizacion-y-control-industrial/sensorica/detectores-lazo-magnetico/detector-lazo-ldp1sa1b230-ld-100-220v.html?\\_\\_SID=U](https://www.centraelectroventas.com/automatizacion-y-control-industrial/sensorica/detectores-lazo-magnetico/detector-lazo-ldp1sa1b230-ld-100-220v.html?__SID=U)
19. MATRIX | BEA Europe [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://www.bea-sensors.com/es/producto/matrix/>
20. Foscam FI9804W - Cámara IP de vigilancia 720p, 1 MP, H264, WiFi, de exterior, 70°, visión nocturna 20 m, 5.5 W | Expertos en Seguridad [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://tienda.prosegur.es/exterior/foscam-fi9804w-camara-ip-de-vigilancia-720p-1-mp-h264-wifi-de-exterior-70-vision-nocturna-20-m-5-5-w.html>
21. Foscam FI9805E - Cámara de vigilancia, plateado | Expertos en Seguridad [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://tienda.prosegur.es/exterior/foscam-fi9805e-camara-de-vigilancia-plateado.html>
22. Cámara IP Wifi de exterior X-Security, 3 Mpx., gran angular de 100 grados, alcance IR 30m [Internet]. [citado 3 de marzo de 2019]. Disponible en: <https://www.evoseguridad.es/camara-ip-wifi-de-exterior-x-security-3-mpx-gran-angular-de-100-grados-alcance-ir-30m-496.html>
23. Cámara Wi-Fi de videovigilancia con visor nocturno Resolución 720p ONV524 Visión Nocturna Protección IP66 Uso con Wi-Fi o Ethernet.



24. Arduino Due [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://store.arduino.cc/due>
25. Genuino Yún Shield - Boards & Modules - Arduino [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://store.arduino.cc/genuino-yun-shield>
26. Raspberry Pi 3 Model B+ - Raspberry Pi [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
27. Raspberry Pi Zero W - Raspberry Pi [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
28. Raspberry Pi Zero - Raspberry Pi [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-zero/#buy-now-modal>
29. Cable para Entornos Hostiles RS PRO, 0,5 mm<sup>2</sup>, 20 AWG, 16/0,2 mm, Rojo, Caucho de Silicona, -60 °C, +180 °C, 750 V | RS Components [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/cables-para-entornos-hostiles/0359683/>
30. 100G0111-0.50-9 | Cable para Entornos Hostiles TE Connectivity, 100G0111-0.50-9, 0,5 mm<sup>2</sup>, 19/0,18 mm, Blanco, Polímero, -55 °C, +125 °C | RS Components [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://es.rs-online.com/web/p/cables-para-entornos-hostiles/2441453/>
31. Bobina 100m Cable Red Rígido UTP Cat. 6 10/100/1000 [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://www.pccomponentes.com/bobina-100m-cable-red-r-gido-utp-cat-6-10-100-1000>
32. Bobina Cable FTP Cat 6 Flexible 100 Mts [Internet]. [citado 4 de marzo de 2019]. Disponible en: <https://www.pccomponentes.com/bobina-cable-ftp-cat-6-flexible-100-mts>
33. ESP8266EX Datasheet. 2018.
34. ESP32 Series Datasheet Including. 2019.
35. Waveshare 4G/3G/2G/GSM/GPRS/GNSS Hat for Raspberry Pi Zero W WH 2B 3B 3B+ Based on SIM7600E-H Supports LTE CAT4 for Downlink Data Transfer: Amazon.es: Electrónica [Internet]. [citado 16 de junio de 2019]. Disponible en: <https://www.amazon.es/Waveshare-4G-Raspberry-SIM7600E-H-Supports/dp/B07F1NSGQ8?th=1>
36. SIM900 GPRS/GSM Shield Development Board Quad-Band Module with Antenna | eBay [Internet]. [citado 5 de marzo de 2019]. Disponible en: <https://www.ebay.es/i/183418168668?chn=ps>



37. Vélez Serrano JF, Moreno Díaz AB, Sanchez Calle Á, Esteban Sánchez-Maríb JL. Vision Por Computador. 2001. 1-168 p. ISBN: 9788497720694.
38. Robbins KA, Robbins S. UNIX programación práctica: guía para la concurrencia, la comunicación y los multihilos. 1997. 4-11 p. ISBN: 0134437063
39. Robbins KA, Robbins S. UNIX programación práctica: guía para la concurrencia, la comunicación y los multihilos. 1997. 431-436 p. ISBN: 0134437063
40. Rifflet J-M. Comunicaciones en Unix. 1992. 151-153 p. ISBN: 9788476157909.
41. Márquez FM. UNIX Programación avanzada. 2004. 379-387 p. ISBN: 8478976035.
42. Robbins KA, Robbins S. UNIX programación práctica: guía para la concurrencia, la comunicación y los multihilos. 1997. 289-323 p. ISBN: 0134437063.
43. Robbins KA, Robbins S. UNIX programación práctica: guía para la concurrencia, la comunicación y los multihilos. 1997. 169-185 p. ISBN: 0134437063.
44. Rifflet J-M. Comunicaciones en Unix. 1992. 233-278 p. ISBN: 9788476157909.
45. Márquez FM. UNIX Programación avanzada. 2004. 427-482 p. ISBN: 8478976035.
46. Lin A, Vives A, Bagula A, Zennaro M, Pietrosemoli E. Internet de las Cosas. 2015. 1-9 p.
47. Microcontroladores vs. Microprocesadores – apLOOP – Medium [Internet]. [citado 16 de junio de 2019]. Disponible en: <https://medium.com/@aploopve/microcontroladores-vs-microprocesadores-9e8c7edfb746>
48. Hardware del Internet de las Cosas - Equipo Altran [Internet]. [citado 4 de mayo de 2019]. Disponible en: <https://equipo.altran.es/hardware-iot-internet-de-las-cosas/>
49. ¿En qué lenguaje se programara el internet de las cosas? BBVA Open [Internet]. [citado 4 de mayo de 2019]. Disponible en: <https://bbvaopen4u.com/es/actualidad/en-que-lenguaje-se-programara-el-internet-de-las-cosas>
50. IoT Compatible Devices - Samsung ARTIK cloud services [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://artik.cloud/works-with/>



51. IoT Cloud Pricing - Samsung ARTIK cloud services [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://artik.cloud/pricing/>
52. About [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://arest.io/about>
53. Pricing [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://arest.io/pricing>
54. Thinger.io Documentation - Open Source Internet of Things [Internet]. [citado 5 de mayo de 2019]. Disponible en: <http://docs.thinger.io/>
55. Pricing – Thinger.io [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://thinger.io/pricing/>
56. ThingSpeak Documentation - MathWorks España [Internet]. [citado 5 de mayo de 2019]. Disponible en: <https://es.mathworks.com/help/thingspeak/>
57. How to Buy - Standard License - ThingSpeak IoT [Internet]. [citado 5 de mayo de 2019]. Disponible en: [https://thingspeak.com/prices/thingspeak\\_standard](https://thingspeak.com/prices/thingspeak_standard)
58. MQTT man page | Eclipse Mosquitto [Internet]. [citado 5 de mayo de 2019]. Disponible en: <http://mosquitto.org/man/mqtt-7.html>
59. ¿Qué es Cloud Computing? | Salesforce [Internet]. [citado 7 de mayo de 2019]. Disponible en: <https://www.salesforce.com/mx/cloud-computing/>
60. Mell P, Grance T. The NIST Definition of Cloud Computing. 2011;
61. OpenCV: Background Subtraction [Internet]. [citado 30 de marzo de 2019]. Disponible en: [https://docs.opencv.org/3.4/db/d5c/tutorial\\_py\\_bg\\_subtraction.html](https://docs.opencv.org/3.4/db/d5c/tutorial_py_bg_subtraction.html)
62. OpenCV: How to Use Background Subtraction Methods [Internet]. [citado 30 de marzo de 2019]. Disponible en: [https://docs.opencv.org/3.1.0/d1/dc5/tutorial\\_background\\_subtraction.htm](https://docs.opencv.org/3.1.0/d1/dc5/tutorial_background_subtraction.htm)
63. Wang R, Bunyak F, Seetharaman G, Palaniappan K. Static and moving object detection using flux tensor with split gaussian models. IEEE Comput Soc Conf Comput Vis Pattern Recognit Work. 2014;(June):420-4.
64. M6 Motorway Traffic - YouTube [Internet]. [citado 1 de abril de 2019]. Disponible en: <https://www.youtube.com/watch?v=PNCJQkvALVc>

