

SEMÁFOROS AUTÓNOMOS PARA CONTROL DE TRÁFICO  
EN LA CIUDAD DE NEIVA

Rodrigo Alberto Sanabria Cortes  
Ingeniería de Sistemas

Universidad Nacional abierta y a distancia

Facultad de Ingeniería

Programa de ingeniería de sistemas

Neiva – Huila

2018

SEMÁFOROS AUTÓNOMOS PARA CONTROL DE TRÁFICO EN LA CIUDAD DE  
NEIVA

Proyecto de grado para optar el título de  
Ingeniero de Sistemas

Director

Daniel Andrés Guzmán Arévalo

Ing. Sistemas. M.SC.

Universidad Nacional abierta y a distancia

Facultad de Ingeniería

Programa de ingeniería de sistemas

Neiva – Huila

2018

**Nota de aceptación**

---

---

---

---

---

**Director de investigación**

---

**Daniel Andrés Guzmán Arévalo**

**Jurados**

---

**Jaime Rubiano Llorente**

---

**Fecha: Marzo 2018**

## AGRADECIMIENTOS

Agradezco a quienes hicieron posible la elaboración de este proyecto; a Dios por la sabiduría para culminar con éxito los cursos de mi carrera profesional.

Daniel Andrés Guzmán Arévalo por su apoyo y asesoría durante esta etapa de mi carrera profesional como requisito para optar el título de AI Ingeniería de Sistemas.

Por ultimo un agradecimiento muy especial a mi familia por su paciencia, amor apoyo incondicional para el cumplimiento de este objetivo en desarrollo de mi vida profesional.

Rodrigo Alberto Sanabria Cortes

## **DEDICATORIA**

A Dios por haberme acompañado y guiado a lo largo de este aprendizaje en mi carrera, por ser mi mayor fortaleza en los momentos de debilidad y por brindarme sabiduría, experiencias y sobre todo felicidad.

Este trabajo realizado con esfuerzo por varios meses, está dedicado a mis padres, familiares y amigos, por estar en los momentos de mi vida, doy fe del crecimiento personal como profesional

## Tabla de contenido

RESUMEN .....	11
ABSTRACT.....	12
INTRODUCCIÓN .....	13
CAPITULO I .....	16
1. EL PROBLEMA .....	16
1.1. Formulación del problema.....	19
2. Objetivos .....	20
2.1. Objetivo General.....	20
2.2. Objetivos Específicos .....	20
3. Justificación .....	21
4. Delimitación.....	22
4.1. Delimitación .....	22
4.2. Delimitación temporal .....	23
4.3. Delimitación temática.....	23
CAPITULO II.....	24
5. MARCO TEOIRICO .....	24

5.1.	Bases teóricas .....	28
5.2.	Sistemas .....	28
5.3.	Semáforos .....	30
5.4.	Modelo Informático .....	31
CAPITULO III.....		60
6.	MARCO METODOLOGICO.....	60
6.1.	Tipo de Investigación .....	60
6.2.	Técnica de recolección de datos. ....	64
6.3.	Procedimiento metodológico.....	65
6.4.	FASE II: ELABORACIÓN .....	67
6.5.	FASE III: CONSTRUCCIÓN .....	68
6.6.	FASE IV: TRANSICIÓN.....	68
CAPITULO IV.....		70
7.	ANÁLISIS DE RESULTADOS .....	70
7.1.	Concepción del sistema .....	70
7.2.	Diseño del sistema .....	72
7.3.	Diseño y estructura del circuito de semáforos .....	76
7.4.	Diseño y solución para las distintas intersecciones .....	77
7.5.	Funcionamiento: .....	78
7.6.	Intersección con desvío: .....	79

7.7. Intersección con desvío en forma de Y: .....	81
7.8. Intersección en forma de T: .....	83
7.9. Construcción o desarrollo del sistema inteligente .....	85
7.10. Declaración de variables .....	86
7.11. Configuración inicial .....	87
7.12. Sensor Ultrasonido.....	89
8. Evaluación de resultados.....	90
CONCLUSIONES .....	92
RECOMENDACIONES.....	93
BIBLIOGRAFIA .....	94

## Índices de ilustración

Ilustración 1 Placa Arduino UNO R3 .....	73
Ilustración 2 Diagrama de Conexiones Arduino UNO R3. ....	74
Ilustración 3 Circuito del Semáforo Inteligente.....	77
Ilustración 4 Intersección Sencilla .....	79
Ilustración 5 Intersección con Desvío.....	81
Ilustración 6 Intersección con Desvío en Forma de Y.....	82
Ilustración 7 Intersección en Forma de T .....	84
Ilustración 8 Nombrando puertos a utilizar. ....	85
Ilustración 9 Nombrado puertos a utilizar .....	86
Ilustración 10 Declaración de variables.....	87
Ilustración 11 Programación sensor ultrasonido.....	89
Ilustración 12 Activar a los 15 cm. ....	90

## Índices de Tablas

Tabla 1. Resolución y precisión de sensores .....	49
Tabla 2 Tipos de Sensores .....	51
Tabla 3 Requerimientos del Hardware .....	71
Tabla 4 Requerimientos de Software.....	71
Tabla 5 Características de Arduino UNO R3 .....	75
Tabla 6 Dirección y Sensores con desvío. ....	79
Tabla 7 Dirección y Sensores Y.....	81
Tabla 8 Dirección Y Sensores en forma T.....	83
Tabla 9 Ventajas de sistema sobre el tradicional.....	90

## RESUMEN

El propósito de esta investigación se centra en desarrollar un sistema de Semáforos Autónomos para control de tráfico en la ciudad de Neiva para el control vehicular basado en hardware programado en lenguajes de alto nivel, para llevar a cabo esta investigación y proyecto se realizara un dispositivo capaz de tomar decisiones sobre como dirigir el tráfico de vehículos en una intersección, de manera más efectiva que los semáforos convencionales ya que estos no toman una decisión, solo realizan un cambio de luces cada cierto tiempo de manera repetitiva y esto genera problemas como largos tiempo de espera innecesarios para desplazarse por la ciudad de Neiva generando congestiones de vehículos y motos que pueden ser resueltos de manera más efectiva y en tiempo real. Sistema de Semáforos Autónomos para control de tráfico en la ciudad de Neiva, se recopilaron los elementos necesarios para la elaboración y diseño, luego de esto se estableció una base de elaboración de la arquitectura del sistema de control de tráfico vehicular que cuenta la ciudad.

La siguiente investigación se consideró del tipo descriptiva ya que al principio del proyecto fueron planteados los requerimiento para el desarrollo del proyecto y la función que tendrán los semáforos para el control de tráfico vehicular en la ciudad, a su vez se considera del tipo proyectiva ya que no solo se basa en una propuesta propone varias alternativas de cambio y tampoco es desligada de un proceso de recolección de datos de información y diseño, dicho esto se establece como un proyecto factible ya que el objetivo principal es el desarrollo de un sistema que dará solución a un problema específico.

Palabras clave: Semáforos Autónomos, Tráfico, Contaminación, Hardware.

## ABSTRACT

The purpose of this research is to develop a system of autonomous traffic lights for traffic control in the city of Neiva for vehicular control based on hardware programmed in high level languages, to carry out this research and project a capable device will be realized. to make decisions about how to direct vehicle traffic at an intersection, more effectively than conventional traffic lights because they do not make a decision, they only make a change of lights every so often in a repetitive manner and this generates problems such as long time It waits unnecessarily to travel through the city of Neiva generating congestions of vehicles and motorcycles that can be solved more effectively and in real time. Autonomous Traffic Light System for traffic control in the city of Neiva, the necessary elements for the elaboration and design were collected, after which a basis for the development of the architecture of the vehicle traffic control system that counts the city was established.

The following investigation was considered of the descriptive type since at the beginning of the project the requirements for the development of the project and the function that the traffic lights will have for the control of vehicular traffic in the city, in turn is considered of the projective type since not only based on a proposal proposes several alternatives for change and is not separated from a process of data collection of information and design, said this is established as a feasible project since the main objective is the development of a system that will provide a solution to a specific problem.

Keywords: Smart traffic lights, Traffic, Pollution, Hardware.

## INTRODUCCIÓN

La presente investigación posee como objetivo primordial la realización de un sistema de tráfico vehicular por medio de Semáforos Autónomos para control de tráfico en la ciudad de Neiva. “que está enfocado a la vida cotidiana del público en general ya que existe una cantidad de población muy alta en las ciudades del planeta, por lo tanto, es necesario la implementación de estos sistemas porque no solamente las calles son circuladas por personas sino también por vehículos automóviles que es el medio de transporte más popular en el planeta.

Para llevar a cabo esta investigación y proyecto se realizara un dispositivo capaz de tomar decisiones sobre como dirigir el tráfico de vehículos en una intersección, de manera más efectiva que los semáforos convencionales ya que estos no toman una decisión, solo realizan un cambio de luces cada cierto tiempo de manera repetitiva y esto genera problemas como largos tiempo de espera innecesarios para desplazarse por la ciudad o congestiones de vehículos que pueden ser resueltos de manera más efectiva.

La siguiente investigación se consideró del tipo descriptiva ya que al principio del proyecto fueron planteados los requerimiento para el desarrollo del proyecto y la función que tendrán los semáforos para el control de tráfico vehicular en la ciudad, a su vez se considera del tipo proyectiva ya que no solo se basa en una propuesta propone varias alternativas de cambio

y tampoco es desligada de un proceso de recolección de datos de información y diseño, dicho esto se establece como un proyecto factible ya que el objetivo principal es el desarrollo de un sistema que dará solución a un problema específico.

En primer lugar, fue necesario cubrir las necesidades para el correcto funcionamiento del semáforo, por lo que en cada capítulo se desarrollan diferentes puntos que permitirán el éxito del sistema.

Se desarrollaron cuatro capítulos tal y como se señalan a continuación:

Capítulo I, el cual se enfoca en el problema y planteamiento de la causa de la investigación, el objetivo general, los objetivos específicos, la justificación y por último la delimitación del proyecto.

Capítulo II, titulado Marco teórico, donde se desarrollaron puntos como los antecedentes relacionados con el objeto de estudio a desarrollar, Asimismo se explicaron las bases teóricas sustentadas en la investigación y por último se definieron términos básicos claves dentro del proyecto.

Capítulo III, denominado Marco metodológico, en donde se procede a definir el tipo de investigación, diseño, unidad de análisis y las técnicas de recolección de la información. Asimismo, se explican de forma breve las fases y sub fases específicas utilizadas para el desarrollo del sistema.

Capítulo IV, titulado Análisis e interpretación de resultados, se desarrollan las fases de la metodología seleccionada y se explican los resultados obtenidos a medida que se aplican cada una de las fases, asimismo, se definen los requerimientos necesarios. De igual forma se muestra el diseño y parte de la programación específica de cada algoritmo de decisión del sistema.

## CAPITULO I

### 1. EL PROBLEMA

A continuación, en este capítulo se presenta un enfoque general de lo que es el planteamiento general del problema de la investigación, se presentaran los objetivos generales y específicos y también la justificación del problema desde los enfoques teóricos, metodológicos, prácticos y sociales, para finalmente concluir con las delimitaciones de este trabajo.

El crecimiento desordenado de la población en distintas ciudades del planeta ha traído como consecuencia que las ciudades crezcan y por lo tanto también su infra estructura ya que es necesario mantener interconectadas las distintas zonas de la ciudad como si se tratara de una red, estas infraestructuras no solo serán circuladas por humanos, también lo serán por automóviles, el cual es uno de los medios de transporte actuales más comunes y cómodos dado que cada individuo puede adquirir su propio automóvil, sin embargo esto ha hecho que la cantidad de vehículos en las calles sea muy elevada y si se tiene en cuenta que en algunas ciudades la venta de vehículos anual es de gran magnitud como en Colombia que solo en el año 2016 se vendieron aproximadamente 905.888 vehículos, y si además se tiene en cuenta que generalmente estas ciudades no habían sido planificadas para manejar este volumen vehicular se hace necesario el uso de herramientas que permitan acelerar al máximo la fluidez del tráfico vehicular.

El control de tráfico comenzó en Europa a finales de la década de 1960 como un esfuerzo

para cambiar el comportamiento de los conductores con el fin de hacer que las calles sean más seguras para los niños, peatones y ciclistas.

Frustrados por el tráfico de paso que hacía que sus calles fueran inseguras, los residentes de la ciudad holandesa de Delft, convirtieron su calle en un camino de obstáculos para los vehículos motorizados y un sitio seguro para los niños. Esta primera solución de control de tráfico, llamada “yardas vivientes” o “Woonerven”, desaceleró el tráfico y disminuyeron su volumen con la colocación de mesas, bancas, areneros y plataformas de estacionamiento que se extendían hacia las calles. Las Woonerven recibieron el apoyo del gobierno casi una década después en 1976. En los años subsecuentes, la idea se propagó a otros países y se crearon normas y leyes para gobernar las dimensiones y ubicación de los diseños Woonerven. Para 1990, millones de calles en países como Austria, Dinamarca, Francia, Alemania, Israel, Japón, Suecia y Suiza se beneficiaron de soluciones Similares. Los sistemas autónomos en el siglo XXI son muy utilizados en ciertas áreas de la ingeniería como en el área automotriz en los sistemas de amortiguación que varían la dureza del amortiguador según la condición lo necesite, estos sistemas consisten en una programación o algoritmo computacional que reúnen características y comportamientos asimilables a los de un ser humano y la parte sensorial o los sentidos del sistema que son una serie de componentes o sensores que ayudan al sistema a responder a las distintas situaciones a las que se someterá.

Cuando se habla de los semáforos tradicionales se hace referencia a un dispositivo eléctrico que controla el tráfico de vehículos en las intersecciones de las ciudades, estos vienen comúnmente en una configuración de 3 luces que representan señales para el tráfico de

vehículos, rojo para detenerse, amarillo o ámbar como paso intermedio del verde a rojo y verde para avanzar.

Razón por la cual ha sido necesario colocar semáforos en las ciudades, sin embargo, en muchas ocasiones éstos no resuelven el problema de tráfico debido a la inadecuada sincronización entre semáforos y al hecho de que los sistemas actuales no toman en cuenta el flujo de autos en las calles, avenidas ni mucho menos cuando se genera congestión en lo mal llamado horas pico donde se genera un colapso vehicular en la ciudad de Neiva.

Entre los principales problemas que se generan debido a la situación actual de los semáforos los más preocupantes son los siguientes:

- Los accidentes automovilísticos que representan una pérdida para la sociedad, ya que es una de las principales causas de mortalidad en el país y en consecuencia Neiva tiene una de las tasas más altas.
- Los largos tiempos de espera para poder desplazarse en la ciudad de Neiva.
- Las altas cantidades de emisiones de CO<sub>2</sub> al ambiente debido a la gran cantidad de autos y motocicletas que quedan atrapados en las congestiones y casos vehicular en la ciudad de Neiva.

Científicos norteamericanos y rumanos han desarrollado un modelo informático basado en información real que atribuye inteligencia a los semáforos para optimizar la gestión del tráfico. De esta forma han comprobado que se reduce un 28% el tiempo de espera en los cruces en hora

pico y un 6,5% las emisiones de CO<sub>2</sub>. El modelo puede potenciarse si se incorpora a los automóviles un software específico que avise a los conductores tanto de las velocidades recomendables en función de las luces de los semáforos, como en función de la cantidad de coches que se pueden encontrar en los atascos. Esta aplicación también podría trasladar información al sistema para mejorar la regulación del tráfico mediante los semáforos.

A principios del siglo XXI donde hablamos del internet de la cosas se emplean en la ciudad de Neiva (Huila) semáforos con contadores digitales, alumbrados por Leds a los que llaman “Semáforos Autónomos”, sin embargo, estos no hacen más que informar a los conductores de los vehículos el tiempo restante para la luz que se encuentra encendida en ese momento, sin la capacidad de decidir por sí mismos a que vías darle más prioridad debido al tráfico vehicular en tiempo real, esto quizá pueda ayudar a resolver algunos de los problemas como lo son los accidentes, sin embargo estos semáforos son deficientes para resolver el problema de la congestión ya que poseen un solo conjunto de rutinas para las luces la cual hace el cambio de luces con tiempos predefinidos que no varían y no están conscientes de la cantidad de vehículos en la vía y por lo tanto no pueden tomar decisiones basadas en la densidad del tráfico y cogestión del mismo en tiempo real.

### **1.1. Formulación del problema**

Luego del planteamiento expuesto, la siguiente investigación estará orientada al desarrollo de un sistema de control de tráfico mediante Semáforos Autónomos para control de tráfico en la ciudad de Neiva. ¿Cómo desarrollar un Semáforos Autónomos para control de

tráfico?, ¿Cuáles son los requerimientos para estos controles de semáforos?, ¿Cómo estará estructurado el semáforo?

## **2. Objetivos**

En este trabajo de grado se presentan las metas que se pautaron para el desarrollo y la elaboración del control “Semáforos Autónomos para control de tráfico en la ciudad de Neiva “.

### **2.1. Objetivo General**

Proponer el desarrollo de un sistema de control de tráfico vehicular por medio Semáforos Autónomos para control de tráfico en la ciudad de Neiva.

### **2.2. Objetivos Específicos**

- Analizar el funcionamiento actual de los semáforos existentes de la ciudad de Neiva.
- Establecer los requerimientos técnicos para la realización del sistema de semáforos autónomos para control de tráfico en la ciudad de Neiva
- Recopilar los elementos necesarios para cumplir cada uno de los objetivos en la elaboración y diseño del sistema de control de tráfico vehicular.
- Establecer la base de la elaboración de la arquitectura para el sistema de control de tráfico vehicular.
- Desarrollar el sistema de control de tráfico vehicular en base a la arquitectura definida

para los semáforos autónomos para control de tráfico en la ciudad de Neiva

- Evaluar la funcionalidad y rendimiento del sistema construido dando solución al problema planteado.

### 3. Justificación

Teóricamente este proyecto estará desarrollado en hardware programado bajo lenguaje de alto nivel compilado, usando la tecnología de microprocesadores y leds, tomando como base los semáforos que se encuentran en la ciudad para mejorarlos en este trabajo de investigación.

Metodológicamente este proyecto sirve para futuros estudiantes que quieran mejorar el diseño de este trabajo de investigación o tomarlo de guía para su uso en otras áreas de control y desarrollo de nuevas tecnologías aplicadas a la movilidad.

Su justificación de manera social en el hecho de que un Semáforo Autónomo para control de tráfico en la ciudad de Neiva ya que existe gran cantidad de vehículos y motocicletas lo que conlleva a una congestión. En este punto se presentan las ventajas que trae la implementación del proyecto en la ciudad, entre ellas están:

Establecer un control de multas basado en parámetros medidos por el sistema. Gracias a estas multas los usuarios serán más precavidos al conducir y por ende se reducirían la cantidad de accidentes.

Eliminar la congestión producto del exceso de vehículos ya que el semáforo calcularía un promedio de tiempo para dar paso a ciertos caminos, basándose en la cantidad de vehículos

Dar paso peatonal cuando exista un peatón presente durante un período de tiempo para luego dejar fluir los vehículos de nuevo. Los peajes pueden obtener información acerca de si hay un exceso de velocidad en los automóviles en tiempo real.

La ingeniería en computación resuelve el problema de sincronización y nos trae un beneficio muy importante como lo es el uso de micro controladores con algoritmos de programación.

Las soluciones específicas dadas por la ingeniería en computación son el empleo de algoritmos y sistemas electrónicos que nos permiten el uso de herramientas que nos ayudaran a monitorear adecuadamente el flujo del tráfico vehicular por las distintas vías de la ciudad de Neiva.

#### **4. Delimitación**

La investigación de Semáforos Autónomos para control de tráfico en la ciudad de Neiva se realizó según las delimitaciones siguientes:

##### **4.1. Delimitación**

El trabajo especial de grado se realizará en la Universidad Nacional Abierta y a Distancia, ubicada en la Carrera 15 # 8-06, Altico, Neiva, Huila.

#### **4.2. Delimitación temporal**

El cual se efectuó entre los meses comprendidos desde septiembre de 2016 hasta noviembre del 2017, en el transcurso de este tiempo se realizará el estudio de los sistemas de control de tráfico mediante el uso de Semáforos Autónomos para control de tráfico en la ciudad de Neiva.

#### **4.3. Delimitación temática**

Esta investigación estará enmarcada en los sistemas de control de tráfico, sistemas de semáforos y sistemas autónomos para integrar en solo sistema el desarrollo del **SIT** es la sigla de “Sistema Inteligente de Transporte”.

## CAPITULO II

### 5. MARCO TEORICO

En este capítulo se presentan las bases teóricas y los trabajos realizados por otros investigadores que sustentan esta investigación “Semáforos Autónomos para control de tráfico en la ciudad de Neiva”, se exponen de una forma clara y sencilla, considerando todo el aspecto necesario para el análisis de las variables que intervienen en la misma, se realizó una búsqueda de todo lo que se sabe sobre Semáforos Autónomos y control del tráfico vehicular.

Para respaldar el siguiente estudio se consultaron varias investigaciones que estudien los sistemas de control de tráfico y Semáforos Autónomos en las cuales se pueda seleccionar ciertos aspectos, que nos aporten alguna información significativa para el desarrollo del estudio, estos se presentaran a continuación:

Deternoz y Fernández (2006) en el trabajo el cual se titula: “Sistema de gestión y monitoreo del tránsito a través de Semáforos Autónomos” este proyecto fue realizado en la escuela de ingeniería en informática en la universidad católica Andrés Bello de Caracas, el objetivo fue el diseño de un sistema de gestión y monitoreo del tránsito, cuya principal funcionalidad es administrar, analizar y almacenar información proveniente de los dispositivos de interconexión, los cuales se encontraron a su vez conectados al semáforo.

Se implementó la metodología de cascada con fases solapadas, que constituyen cada una de las etapas del ciclo de vida de desarrollo del software desde la investigación preliminar hasta

las pruebas del sistema ya que en las fases en el modelo SASHIMI se superponen, lo que implica que se puede actuar durante las etapas anteriores.

Los resultados obtenidos fueron la elaboración de un módulo que calcula, gestiona y controla las luces de los semáforos automáticamente. Este módulo, además, hace que los semáforos vecinos trabajen en conjunto para descongestionar la vía, cualidad importante para la autonomía del sistema.

Dicha investigación es considerada un antecedente ya que genera aportes significativos al proyecto, debido a que justifica el uso de sensores como controladores del tráfico para la contribución del descongestionamiento de las vías principales e intersecciones trabajando en forma sincronizada con los semáforos vecinos. Así mismo nos servirá para justificar teóricamente esta investigación ya que las causas y síntomas planteados en la misma son similares a las que se manejan en este estudio.

Marian Gómez y Carolina Mas (2007) en el trabajo titulado “Diseño de un semáforo inalámbrico, soportado por un subsistema de la plataforma de control de tráfico PISACOTA” este proyecto fue realizado en la escuela de ingeniería en telecomunicaciones en la universidad católica Andrés Bello en Caracas, el objetivo de este proyecto fue el diseño de una comunicación inalámbrica para el control de semáforos formando parte del proyecto PISACOTA (Plataforma Integrada para el Seguimiento, Análisis y Control del Tráfico Automotor), el cual busca ser una alternativa destinada a mitigar el los efectos del congestionamiento del tráfico vehicular.

Los resultados obtenidos fueron la creación de un subsistema capaz de controlar una intersección sincronizada, compuesta por 4 semáforos vehiculares y 4 semáforos de peatones, dotados de dos botones, capaz de reportar a la central, el estado de conexión con cada una de las unidades de control de semáforo y las solicitudes de alarma y paso de peatones.

Se empleó una metodología de cinco fases 1) Investigación y estudio, 2) diseño y programación 3) Construcción del hardware, 4) Ensamblaje e implementación y 5) pruebas. La primera de ellas completamente teórica y abarcó el estudio de los antecedentes en cuanto a la construcción de semáforos y sistemas ya instalados; la segunda implicó la selección de un método confiable de transporte de datos entre la central y la intersección así como también incluyó la selección de los micro controladores; la tercera, consistió en la programación de los micro controladores; la cuarta en el diseño y construcción de las tarjetas requeridas; y la quinta, conformó la integración del método de transporte con las diferentes tarjetas, la realización de pruebas de laboratorio y la detección y corrección de las fallas presentadas.

Dicha investigación genera un aporte significativo al presente proyecto porque en ella se lleva a cabo el uso micro controlador e interconexiones inalámbricas para transmitir datos hacia una central sobre alarmas y peatones, controlar una intersección sincronizada compuesta por 4 semáforos vehiculares y 4 peatonales, y además el diseño del sistema de semáforos interconectados de manera inalámbrica. Resulta importante esta investigación, gracias a que su enfoque tecnológico tiene relación con la investigación a desarrollar “Semáforos Autónomos

para control de tráfico en la ciudad de Neiva”.

Los resultados obtenidos fueron de las diez intersecciones estudiados, el 70% de las mismas presentan niveles de servicio tipo F resultando viable la utilización de sensores de tráfico en las mismas basándose en los resultados que arrojaron los módulos de ajuste de volúmenes, flujos de saturación, análisis de capacidad y módulos de niveles de servicio para los “Semáforos Autónomos para control de tráfico en la ciudad de Neiva”.

La investigación de “Semáforos Autónomos para control de tráfico en la ciudad de Neiva”. es del tipo descriptiva de acuerdo con un diseño no experimental, transaccional y de campo cuyas técnicas de recolección de datos se basaron en la observación directa y la utilización de planillas de conteo e inventarios de campo las cuales permitieron recolectar la información requerida directamente de las intersecciones pertenecientes al presente estudio en su entorno diario para posteriormente obtener los datos de variaciones horarias de volúmenes de tránsito, ajuste de volúmenes, módulos de flujo de saturación, análisis de capacidad y módulos de nivel de servicio los cuales sirvieron de parámetros para establecer la factibilidad de la implementación de sensores de tráfico en las mismas.

Este trabajo es considerado un antecedente de esta investigación por que establece ciertas bases teóricas para su desarrollo, aporta ciertos modelos de flujos de vehículos y ajustes de volúmenes para las avenidas y respalda el uso de esta tecnología demostrando que si es viables u uso ya que puede regular el tráfico controlando los semáforos con información del flujo vehicular presente en las intersecciones de la ciudad con datos en vivo.

## **5.1. Bases teóricas**

En la investigación “Semáforos Autónomos para control de tráfico en la ciudad de Neiva”, se vio la necesidad de desarrollar ciertos conceptos y definiciones para un entendimiento y un enfoque más completo del tema para así obtener el sustento teórico para el desarrollo del sistema, los conceptos son los siguientes:

## **5.2. Sistemas**

Un sistema se define como un conjunto de mecanismos y herramientas que permiten la creación e interconexión de componentes de software, junto con una colección de servicios para facilitar las labores de los componentes que residen y se ejecutan en él.

### **5.2.1. Sistemas autónomos}**

Un sistema inteligente es un programa de computación que reúne características y comportamientos asimilables al de la inteligencia humana o animal.

La expresión "sistema inteligente" se usa a veces para sistemas autónomos incompletos, por ejemplo, para una casa autónoma o un sistema experto.

Un sistema inteligente completo incluye "sentidos" que le permiten recibir información

de su entorno. Puede actuar, y tiene una memoria para archivar el resultado de sus acciones. Tiene un objetivo e, inspeccionando su memoria, puede aprender de su experiencia. Aprende cómo lograr mejorar su rendimiento y eficiencia.

### **5.2.2. Sistemas electrónicos**

Un sistema electrónico es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Una forma de entender los sistemas electrónicos consiste en dividirlos en las siguientes partes:

Entradas o Inputs – Sensores (o transductores) electrónicos o mecánicos que toman las señales (en forma de temperatura, presión, etc.) del mundo físico y las convierten en señales de corriente o voltaje. Ejemplo: El termopar, la foto resistencia para medir la intensidad de la luz, etc.

Circuitos de procesamiento de señales – Consisten en consisten en artefactos electrónicos conectadas juntas para manipular, interpretar y transformar las señales de voltaje y corriente provenientes de los transductores.

Salidas (Outputs – Actuadores u otros dispositivos (también transductores) que convierten las señales de corriente o voltaje en señales físicamente útiles. Por ejemplo: una pantalla que nos registre la temperatura, un foco o sistema de luces que se encienda automáticamente cuando esté oscureciendo.

Básicamente son tres etapas: La primera (transductor), la segunda (circuito procesador) y la tercera (circuito actuador).

### **5.3. Semáforos**

Los semáforos son dispositivos eléctricos que tienen como función ordenar y regular el tránsito de vehículos y peatones en calles y carreteras por medio de luces generalmente de color rojo, amarillo y verde, operados por una unidad de control. (Cal y Mayor, 2007).

El semáforo “es un dispositivo luminoso operado eléctricamente por un controlador y que visualmente comunica una acción o acciones preestablecidas al tránsito” (Covenin 2753:1999), ubicados generalmente en intersecciones para poder regular el flujo de vehículos y peatones en calles y carreteras. Tiene como funciones principales:

- Alternar el permiso de circulación entre una corriente vehicular o peatonal y otras.
- Regular la velocidad de los vehículos para mantener la circulación continua sin congestión y armonía.
- Controlar la circulación por canales.
- Reducir el número de accidentes y colisiones, ocasionados generalmente en intersecciones.
- Generar orden en el tráfico de la ciudad de Neiva.

### **5.3.1. Semáforos autónomos**

Un semáforo inteligente es aquel que “detecta” la cantidad de flujo vehicular mediante sensores (que usualmente están colocados en la carpeta asfáltica) y con base a parámetros ya establecidos, van “modificando” los tiempos de paso y/o detención. Es decir, es un sistema de semáforos "programables" vía remota desde un centro de control gobernado por humanos, que definirán cuál parámetro usar, esto quiere decir que nos permite decidir si es necesario modificar el comportamiento de los semáforos mediante la observación en tiempo de ejecución.

### **5.4. Modelo Informático**

Es la representación de la realidad por medio de abstracciones. Los modelos enfocan ciertas partes importantes de un sistema (por lo menos, aquella que le interesan a un tipo de modelo específico), restándole importancia a otras integrados de uno a uno) desde las sentencias mnemónicas a las instrucciones y datos de máquina. Esto está en contraste con los lenguajes de alto nivel, en los cuales una sola declaración generalmente da lugar a muchas instrucciones de máquina.

Muchos sofisticados ensambladores ofrecen mecanismos adicionales para facilitar el desarrollo del programa, controlar el proceso de ensamblaje, y la ayuda de depuración. Particularmente, la mayoría de los ensambladores modernos incluyen una facilidad de macro (descrita más abajo), y son llamados macro ensambladores.

Fue usado principalmente en los inicios del desarrollo de software, cuando aún no se contaba con potentes lenguajes de alto nivel y los recursos eran limitados. Actualmente se utiliza con frecuencia en ambientes académicos y de investigación, especialmente cuando se requiere la manipulación directa de hardware, altos rendimientos, o un uso de recursos controlado y reducido.

Muchos dispositivos programables (como el micro controlador) aún cuentan con el ensamblador como la única manera de ser manipulados.

#### **5.4.1. Programa ensamblador**

Típicamente, un programa ensamblador moderno crea código objeto traduciendo instrucciones mnemónicas de lenguaje ensamblador en OPCODES, y resolviendo los nombres simbólicos para las localizaciones de memoria y otras entidades.

El uso de referencias simbólicas es una característica clave del lenguaje ensamblador, evitando tediosos cálculos y actualizaciones manuales de las direcciones después de cada modificación del programa. La mayoría de los ensambladores también incluyen facilidades de macros para realizar sustitución textual ej. Generar cortas secuencias de instrucciones como expansión en línea en vez de llamar a subrutinas. Los ensambladores son generalmente más simples de escribir que los compiladores para los lenguajes de alto nivel, y han estado disponibles desde los años 1950. Los ensambladores modernos, especialmente para las arquitecturas basadas en RISC, tales como MIPS, SUN SPARC, y HPPA-RISC, así como también para el x 86 (-64), optimizan la planificación de instrucciones para explotar la

segmentación del CPU eficientemente.

En los compiladores para lenguajes de alto nivel, son el último paso antes de generar el código ejecutable.

#### **5.4.1.1. Número de pasos**

Hay dos tipos de ensambladores basados en cuántos pasos a través de la fuente son necesarios para producir el programa ejecutable.

- Los ensambladores de un solo paso pasan a través del código fuente símbolos y sus valores en el primer paso, después usan la tabla en un segundo paso para generar código.
- El ensamblador debe por lo menos poder determinar la longitud de cada instrucción en el primer paso para que puedan ser calculadas las direcciones de los símbolos.
- La ventaja de un ensamblador de un solo paso es la velocidad, que no es tan importante como lo fue en un momento dado los avances en velocidad y capacidades del computador. La ventaja del ensamblador de dos pasos es que los símbolos pueden ser definidos dondequiera en el código fuente del programa.

Esto permite a los programas ser definidos de maneras más lógicas y más significativas, haciendo los programas de ensamblador de dos pasos más fáciles de leer y mantener.

#### 5.4.1.2. Ensambladores de alto nivel

Los más sofisticados ensambladores de alto nivel proporcionan abstracciones del lenguaje tales como:

- Estructuras de control avanzadas
- Declaraciones e invocaciones de procedimientos/funciones de alto nivel.
- Tipos de datos abstractos de alto nivel, incluyendo las estructuras/records, uniones, clases, y conjuntos.
- Procesamiento de macros sofisticado (aunque está disponible en los ensambladores ordinarios desde finales 1960 para el IBM/360, entre otras máquinas).
- Características de programación orientada a objetos.

#### 5.4.1.3. Uso del término

Note que, en el uso profesional normal, el término ensamblador es frecuentemente usado tanto para referirse al lenguaje ensamblador como también al programa ensamblador (que convierte el código fuente escrito en el lenguaje ensamblador a código objeto que luego será enlazado para producir lenguaje de máquina). Las dos expresiones siguientes utilizan el término "ensamblador":

- "El CP/CMS fue escrito en ensamblador del IBM S/360"

- "El ASM-H fue un ensamblador del S/370 ampliamente usado"

La primera se refiere al lenguaje y el segundo se refiere al programa.

### **5.4.2. Lenguaje**

El lenguaje ensamblador refleja directamente la arquitectura y las instrucciones en lenguaje de máquina de la CPU, y pueden ser muy diferentes de una arquitectura de CPU a otra.

Cada arquitectura de microprocesador tiene su propio lenguaje de máquina, y en consecuencia su propio lenguaje ensamblador ya que este se encuentra muy ligado a la estructura del hardware para el cual se programa. Los microprocesadores difieren en el tipo y número de operaciones que soportan; también pueden tener diferente cantidad de registros, y distinta representación de los tipos de datos en memoria. Aunque la mayoría de los microprocesadores son capaces de cumplir esencialmente las mismas funciones, la forma en que lo hacen difiere y los respectivos lenguajes ensamblador reflejan tal diferencia.

#### **5.4.2.1. Instrucciones del CPU**

La mayoría de los CPU tienen más o menos los mismos grupos de instrucciones, aunque no necesariamente tienen todas las instrucciones de cada grupo. Las operaciones que se pueden realizar varían de un CPU a otro. Un CPU particular puede tener instrucciones que no tengan otro y viceversa. Los primeros microprocesadores de 8 bits no tenían operaciones para

multiplicar o dividir números, por ejemplo, y había que hacer subrutinas para realizar esas operaciones. Otros CPU puede que no tengan operaciones de punto flotante y habría que hacer o conseguir bibliotecas que realicen esas operaciones.

Las instrucciones del CPU pueden agruparse de acuerdo a su funcionalidad en:

- Operadores con enteros: (de 8, 16, 32 y 64 bits dependiendo de la arquitectura del CPU).

Estas son operaciones realizadas por la Unidad Aritmético Lógica del CPU:

- Operaciones aritméticas. Como suma, resta, multiplicación, división, módulo, cambio de signo.
  - Operaciones booleanas. Operaciones lógicas bit a bit como AND, OR, XOR, NOT.
  - Operaciones de bits. Como desplazamiento y rotaciones de bits (hacia la derecha o hacia la izquierda, a través del bit del acarreo o sin él).
- Comparaciones.
  - Operaciones de mover datos:
  - Entre los registros y la memoria:

Aunque la instrucción se llama "mover", en el CPU, "mover datos" significa en realidad copiar datos, desde un origen a un destino, sin que el dato desaparezca del origen. Se pueden mover valores:

- Desde un registro a otro.
- Desde un registro a un lugar de la memoria.
- Desde un lugar de la memoria a un registro.
- Desde un lugar a otro de la memoria.
- Un valor inmediato a un registro.
- Un valor inmediato a un lugar de memoria.

Operaciones de stack:

- PUSH (escribe datos hacia el tope del stack).
- POP (lee datos desde el tope del stack).

**Operaciones de entrada/salida:** Son operaciones que mueven datos de un registro, desde y hacia un puerto; o de la memoria, desde y hacia un puerto.

- INPUT Lectura desde un puerto de entrada.
- OUTPUT Escritura hacia un puerto de salida.
- Operaciones para el control del flujo del programa:
- Llamadas y retornos de subrutinas.
- Llamadas y retornos de interrupciones.
- Saltos condicionales de acuerdo al resultado de las comparaciones.
- Saltos incondicionales.

**Operaciones con números reales:** El estándar para las operaciones con números reales en los

CPU está definido por el IEEE 754. Un CPU puede tener operaciones de punto flotante con números reales mediante el coprocesador numérico (si lo hay), como las siguientes:

- Operaciones aritméticas. Suma, resta, multiplicación, división, cambio de signo, valor absoluto, parte entera.
- Operaciones trascendentales:
- Operaciones trigonométricas. Seno, coseno, tangente, arco tangente.
- Operaciones con logaritmos, potencias y raíces.

El lenguaje ensamblador tiene mnemónicos para cada una de las instrucciones del CPU en adición a otros mnemónicos a ser procesados por el programa ensamblador (como por ejemplo macros y otras sentencias en tiempo de ensamblado).

La transformación del lenguaje ensamblador en código máquina la realiza un programa ensamblador, y la traducción inversa la puede efectuar un desensamblador. A diferencia de los lenguajes de alto nivel, aquí hay usualmente una correspondencia 1 a 1 entre las instrucciones simples del ensamblador y el lenguaje de máquina. Sin embargo, en algunos casos, un ensamblador puede proveer "pseudo instrucciones" que se expanden en un código de máquina más extenso a fin de proveer la funcionalidad necesaria y simplificar la programación.

Por ejemplo, para un código máquina condicional como "si X mayor o igual que", un ensamblador puede utilizar una pseudo instrucción al grupo "haga si menor que", y "si = 0" sobre el resultado de la condición anterior. Los Ensambladores más completos también proveen

un rico lenguaje de macros que se utiliza para generar código más complejo y secuencias de datos.

Para el mismo procesador y el mismo conjunto de instrucciones de CPU, diferentes programas ensambladores pueden tener, cada uno de ellos, variaciones y diferencias en el conjunto de mnemónicos o en la sintaxis de su lenguaje ensamblador. Por ejemplo, en un lenguaje ensamblador para la arquitectura x86, se puede expresar la instrucción para mover 5 al registro AL de la siguiente manera: `MOV AL, 5`, mientras que para otro ensamblador para la misma arquitectura se expresaría al revés: `MOV 5, AL`. Ambos lenguajes ensambladores harían exactamente lo mismo, solo que está expresado de manera diferente. El primero usa la sintaxis de Intel, mientras que el segundo usa la sintaxis de AT&T

El uso del ensamblador no resuelve definitivamente el problema de cómo programar un sistema basado en microprocesador de modo sencillo ya que, para hacer un uso eficiente del mismo, hay que conocer a fondo el microprocesador, los registros de trabajo de que dispone, la estructura de la memoria, y muchas cosas más referentes a su estructura básica de funcionamiento.

### **5.4.3. Lenguajes de Alto Nivel**

Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas.

En los primeros lenguajes de nivel bajo la limitación era que se orientaban a un área específica y sus instrucciones requerían de una sintaxis predefinida. Se clasifican como lenguajes procedimentales.

Otra limitación de los lenguajes de nivel bajo es que se requiere de ciertos conocimientos de programación para realizar las secuencias de instrucciones lógicas. Los lenguajes de alto nivel se crearon para que el usuario común pudiese solucionar un problema de procesamiento de datos de una manera más fácil y rápida.

#### **5.4.4. Micro controlador**

Un micro controlador es un circuito integrado programable o chip, capaz de ejecutar las órdenes grabadas en su memoria, que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado. Sus prestaciones son limitadas, pero su característica principal es su alto nivel de especialización. Difiere de un CPU normal en que es más fácil convertirlo en un controlador en funcionamiento, con un mínimo de chips externos de apoyo. La idea es que solo sea necesario que el chip se coloque en el dispositivo, conectado a la fuente de energía y de información necesaria. Un microprocesador tradicional no permitiría hacer esto, ya que espera que todas estas tareas sean manejadas por otros chips.

Generalmente este chip incluye un procesador o CPU (Unidad Central de

Procesamiento), una memoria RAM para contener los datos, una memoria tipo ROM/PROM/EPROM/FLASH para el programa, varias líneas de E/S para controlar los periféricos, encargados de manejar: comunicación serial, comunicación paralela, temporizadores, conversores de A/D y D/A, perro guardián

(Watchdog), oscilador externo, comparadores analógicos, entre otros y una unidad lógica aritmética (ALU).

LA arquitectura de un micro controlador puede ser RISC (reducido) o CISC (complejo), lo cual dependerá del número de instrucciones que pueda manejar, siendo definido el conjunto de estas instrucciones por el modelo y por el fabricante.

#### **5.4.5. Microprocesadores**

El microprocesador (o simplemente procesador) es el circuito integrado central y más complejo de un sistema informático; a modo de ilustración, se le suele llamar por analogía el «cerebro» de un computador. Es un circuito integrado conformado por millones de componentes electrónicos. Constituye la unidad central de procesamiento (CPU) de un PC catalogado como microcomputador.

Es el encargado de ejecutar los programas; desde el sistema operativo hasta las aplicaciones de usuario; sólo ejecuta instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar,

dividir, las lógicas binarias y accesos a memoria.

Esta unidad central de procesamiento está constituida, esencialmente, por registros, una unidad de control, una unidad aritmética lógica (ALU) y una unidad de cálculo en coma flotante (conocida antiguamente como co-procesador matemático).

El microprocesador está conectado generalmente mediante un zócalo específico de la placa base de la computadora; normalmente para su consta de un disipador de calor fabricado en algún material de alta conductividad térmica, como cobre o aluminio, y de uno o más ventiladores que eliminan el correcto y estable funcionamiento, se le incorpora un sistema de refrigeración que exceso del calor absorbido por el disipador. Entre el ventilador y la cápsula del microprocesador usualmente se coloca pasta térmica para mejorar la conductividad del calor. Existen otros métodos más eficaces, como la refrigeración líquida o el uso de células peltier para refrigeración extrema, aunque estas técnicas se utilizan casi exclusivamente para aplicaciones especiales, tales como en las prácticas de overclocking.

La medición del rendimiento de un microprocesador es una tarea compleja, dado que existen diferentes tipos de "cargas" que pueden ser procesadas con diferente efectividad por procesadores de la misma gama. Una métrica del rendimiento es la frecuencia de reloj que permite comparar procesadores con núcleos de la misma familia, siendo este un indicador muy limitado dada la gran variedad de diseños con los cuales se comercializan los procesadores de una misma marca y referencia. Un sistema informático de alto rendimiento puede estar equipado con varios microprocesadores trabajando en paralelo, y un microprocesador puede, a su vez,

estar constituido por varios núcleos físicos o lógicos. Un núcleo físico se refiere a una porción interna del microprocesador cuasi-independiente que realiza todas las actividades de una CPU solitaria, un núcleo lógico es la simulación de un núcleo físico a fin de repartir de manera más eficiente el procesamiento. Existe una tendencia de integrar el mayor número de elementos dentro del propio procesador, aumentando así la eficiencia energética y la miniaturización. Entre los elementos integrados están las unidades de punto flotante, controladores de la memoria RAM, controladores de buses y procesadores dedicados de video.

#### **5.4.6. Arduino**

Arduino es una plataforma de hardware libre, basada en una placa con un micro controlador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un micro controlador Atm el AVR y puertos de entrada/salida. Los micros controladores más usados son el Atmega168,

Atmega328, Atmega1280, Atmega8 por su sencillez y bajo coste permiten el desarrollo de múltiples diseños. Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que corre en la placa.

Desde octubre de 2012, Arduino se usa también con micro controladoras

CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las micro controladoras CortexM3 usan 3.3V, a diferencia de la mayoría de las placas con AVR que usan mayormente 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atm el AVR a 3.3V como la Arduino Fio y existen clones o imitaciones de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Al ser open-hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

#### 5.4.6.1. **Lenguaje de programación Arduino**

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el popular lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino. Algunos ejemplos son:

- Java
- Flash (mediante Action Script)

- Processing
- Pure Data
- Max MSP (entorno gráfico de programación para aplicaciones musicales, de audio y multimedia)
- VVVV (síntesis de vídeo en tiempo real)
- Adobe Director
- Python
- Ruby
- C
- C++ (mediante lib Serial o en Windows)
- C#
- Cocoa/Objective-C (para Mac OS X)
- Linux TTY (terminales de Linux)
- 3DVIA Virtools (aplicaciones interactivas y de tiempo real)
- Super Collider (síntesis de audio en tiempo real)
- Instant Reality (X3D)
- Liberlab (software de medición y experimentación)
- Blitz Max (con acceso restringido)
- Squeak (implementación libre de Smalltalk)
- Mathematica
- Matlab
- Minibloq (Entorno gráfico de programación, corre también en OLPC)
- Isadora (Interactividad audiovisual en tiempo real)

- Perl
- Physical Etoys (Entorno gráfico de programación usado para proyectos de robótica educativa)
- Scratch for Arduino (S4A) (entorno gráfico de programación, modificación del entorno para niños Scratch, del MIT)
- Visual Basic .NET
- VBScript
- Gambas
- Php

Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida.

Es bastante interesante tener la posibilidad de interactuar con Arduino mediante esta gran variedad de sistemas y lenguajes ya que dependiendo de cuales sean las necesidades del problema que vamos a resolver podremos aprovecharnos de la gran compatibilidad de comunicación que ofrece.

#### **5.4.7. Sensores**

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una TD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

Áreas de aplicación de los sensores: Industria automotriz, robótica, industria aeroespacial, medicina, industria de manufactura, etc. Los sensores pueden estar conectados a un computador para obtener ventajas como son el acceso a una base de datos, la toma de valores desde el sensor, etc.

#### 5.4.7.1. **Características de un sensor:**

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: es el error de medida máximo esperado.

- Offset o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- Repetitividad: error esperado al repetir varias veces la misma medida.

Un sensor es un tipo de transductor que transforma la magnitud que se quiere medir o controlar, en otra, que facilita su medida. Pueden ser de indicación directa o pueden estar conectados a un indicador (posiblemente a través de un convertidor analógico a la digital, un computador y una pantalla (display)) de modo que los valores detectados puedan ser leídos por un humano.

Por lo general, la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento, como por ejemplo un puente de Wheatstone, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para el resto de los circuitos.

### 5.4.7.2. Resolución y precisión

En la siguiente Tabla 1 se presentan la resolución y precisión de sensores

**Tabla 1. Resolución y precisión de sensores**

Magnitud	Transductor	Característica
Posición lineal o angular	Potenciómetro	Analógica
	Encoder	Digital
	Sensor Hall	Digital
Desplazamiento y deformación	Transformador diferencial de variación lineal	Analógica
	Galga extensiométrica	Analógica
	Magnetostrictivos	A/D
	Magnetorresistivos	Analógica

	LVDT	Analógica
Velocidad lineal y angular	Dinamo tacométrica	Analógica
	Encoder	Digital
	Detector inductivo	Digital
	Servo-inclinómetros	A/D

1. Fuente: Lenguaje ensamblador. Extraído el 20 de marzo del 2017  
[http://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador)

La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida. Sin embargo, la precisión es el máximo error esperado en la medida.

La resolución puede ser de menor valor que la precisión. Por ejemplo, si al medir una distancia la resolución es de 0,01 mm, pero la precisión es de 1 mm, entonces pueden apreciarse variaciones en la distancia medida de 0,01 mm, pero no puede asegurarse que haya un error de medición menor a 1 mm. En la mayoría de los casos este exceso de resolución conlleva a un

exceso innecesario en el coste del sistema. No obstante, en estos sistemas, si el error en la medida sigue una distribución normal o similar, lo cual es frecuente en errores accidentales, es decir, no sistemáticos, la repetitividad podría ser de un valor inferior a la precisión.

Sin embargo, la precisión no puede ser de un valor inferior a la resolución, pues no puede asegurarse que el error en la medida sea menor a la mínima variación en la magnitud de entrada que puede observarse en la magnitud de salida.

#### 5.4.7.3. Tipos de sensores

En la siguiente **¡Error! No se encuentra el origen de la referencia.** se indican algunos tipos y ejemplos de sensores electrónicos.

**Tabla 2 Tipos de Sensores**

	RVDT	Analógica
	Giróscopo	
Aceleración	Acelerómetro	Analógico
	Servo-accelerómetros	

Fuerza y por (deformación)	Galga extensiométrica	Analógica
	Triaxiales	A/D
Presión	Membranas	Analógica
	Piezoeléctricos	Analógica
	Manómetros Digitales	Digital
Caudal	Turbina	Analógica
	Magnético	Analógica
Temperatura	Termopar	Analógica
	RTD	Analógica

Notas: Tipos de Sensores Fuente: Lenguaje ensamblador. Extraído el 20 de marzo del 2017

[http://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador)

Algunas magnitudes pueden calcularse mediante la medición y cálculo de otras, por

ejemplo, la velocidad de un móvil puede calcularse a partir de la integración numérica de su aceleración. La masa de un objeto puede conocerse mediante la fuerza gravitatoria que se ejerce sobre él en comparación con la fuerza gravitatoria ejercida sobre un objeto de masa conocida (patrón).

#### **5.4.8. Tráfico**

Es el fenómeno causado por la congestión vehicular en una vía o intersección. (Cal y Mayor, 2007). Se presenta también con muchas similitudes en otros fenómenos como el flujo de partículas (líquidos, gases o sólidos) y el de peatones.

##### **5.4.8.1. Congestión vehicular**

La congestión vehicular o vial se refiere tanto urbana como interurbanamente, a la condición de un flujo vehicular que se ve saturado debido al exceso de demanda de las vías, produciendo incrementos en los tiempos de viaje y atascamientos. Este fenómeno se produce comúnmente en las horas punta u horas pico, y resultan frustrantes para los automovilistas, ya que resultan en pérdidas de tiempo y con sumo excesivos de combustible.

Las consecuencias de las congestiones vehiculares denotan en accidentes, a pesar que los automóviles no pueden circular a gran velocidad, ya que el automovilista pierde la calma al encontrarse estático por mucho tiempo en un lugar de la vía. Esto también deriva en violencia vial, por otro lado, reduce la gravedad de los accidentes ya que los vehículos no se desplazan a

una velocidad importante para ser víctima de daños o lesiones de mayor gravedad. También, los vehículos pierden innecesariamente combustible debido a que se está inactivo por mucho tiempo en un mismo lugar, sin avanzar en el trayecto de un punto a otro.

#### **5.4.8.2. Causas**

La congestión del tráfico se produce cuando el volumen de tráfico o de la distribución normal del transporte genera una demanda de espacio mayor que el disponible en las carreteras. Hay una serie de circunstancias específicas que causan o agravan la congestión, la mayoría de ellos reducen la capacidad de una carretera en un punto determinado o durante un determinado periodo, o aumentar el número de vehículos necesarios para un determinado caudal de personas o mercancías. En muchas ciudades altamente pobladas la congestión vehicular es recurrente, y se atribuye a la gran demanda del tráfico, la mayoría del resto se atribuye a incidentes de tránsito, obras viales y eventos climáticos. La velocidad y el flujo también pueden afectar la capacidad de la red, aunque la relación es compleja. Es difícil predecir en qué condiciones un "atasco" sucede, pues puede ocurrir de repente. Se ha constatado que los incidentes (tales como accidentes o incluso un solo coche frenado en gran medida en un buen flujo anteriormente) pueden causar repercusiones (un fallo en cascada), que luego se difunde y crear un atasco de tráfico sostenido, cuando, de otro modo, el flujo normal puede haber continuado durante algún tiempo más.

#### **5.4.8.3. Efectos negativos**

- Perdida del tiempo de los automovilistas y pasajeros oportunidad").
- Como una actividad no productiva para la mayoría reduce la salud económica regional.
- Retrasos, lo cual puede resultar en la hora atrasada de llegada para el empleo, las reuniones, y la educación, lo que al final resulta en pérdida de negocio, medidas disciplinarias u otras pérdidas personales.
- Incapacidad para predecir con exactitud el tiempo de viaje, lo que lleva a los conductores la asignación de más tiempo para viajar "por si acaso", y menos tiempo en actividades productivas.
- Desperdicio de combustible, aumenta la contaminación en el aire y las emisiones de dióxido de carbono (que puede contribuir al calentamiento global), debido al aumento de ralentización, aceleración y frenado. Aumento del uso de combustibles, en teoría, también puede causar un aumento de los costes de combustible.

Un porcentaje menor de ellos se debe a fallas de fabricación de vehículos, lo cual no excluye atribuirles un "error humano consciente". Posteriores investigaciones de estos "incidentes" han corroborado esta afirmación.

Los accidentes de tráfico tienen diferentes escalas de gravedad, el más grave se considera aquel del que resultan víctimas mortales, bajando la escala de gravedad cuando hay heridos graves, heridos leves, y el que origina daños materiales a los vehículos afectados.

Siempre hay una causa desencadenante que produce un accidente, que se puede agravar de forma considerable si por él resultan afectadas otras personas, además de la persona que lo desencadena.

Asimismo, un accidente puede verse agravado si no se ha hecho uso adecuado de los medios preventivos que no lo evitan, pero reducirían su gravedad. Por ejemplo, no llevar ajustado el cinturón de seguridad o no llevar puesto el casco si se conduce una motocicleta.

Los accidentes de tráfico suelen ocurrir principalmente por los siguientes factores:

- Factor humano: Los factores humanos son la causa del mayor porcentaje de accidentes de tránsito. Pueden convertirse en agravantes a la culpabilidad del conductor causante, según la legislación de tránsito de cada país.
- Conducir bajo los efectos del alcohol (mayor causalidad de accidentes), medicinas y estupefacientes.
- Realizar maniobras imprudentes y de omisión por parte del conductor:
- Efectuar adelantamientos en lugares prohibidos (Choque frontal muy grave).
- Atravesar un semáforo en rojo, desobedecer las señales de tránsito.
- Circular por el carril contrario (en una curva o en un cambio de rasante).
- Conducir a exceso de velocidad (produciendo vuelcos, salida del automóvil de la carretera, derrapes).
- Usar inadecuadamente las luces del vehículo, especialmente en la noche.
- Condiciones no aptas de salud física y mental/emocional del conductor o del peatón (ceguera, daltonismo, sordera, etc.).
- Peatones que cruzan por lugares inadecuados, juegan en carreteras, lanzan objetos resbaladizos al carril de circulación (aceites, piedras).
- Inexperiencia del conductor al volante.

- Fatiga del conductor como producto de la apnea o falta de sueño. Factor mecánico:
- Vehículo en condiciones no adecuadas para su operación (sistemas averiados de frenos, dirección o suspensión).
- Niebla, humedad, derrumbes, zonas inestables, hundimientos.
- Semáforo que funciona incorrectamente o por daño.

#### **5.4.9. Misiones de CO<sub>2</sub>**

Las emisiones de dióxido de carbono tienen dos orígenes, naturales y antropogénicas, teniendo estas últimas un fuerte crecimiento en las últimas décadas. El promedio actual de emisiones de CO<sub>2</sub> en el aire oscila alrededor de 380 ppm, o 0,038%, con algunas variaciones día-noche, estacionales (por la parte antrópica) y con picos de contaminación localizados.

Con grandes variaciones, el incremento anual en la concentración de CO<sub>2</sub> en la atmósfera ha pasado de 0,5 ppm/año en 1960 a 2 ppm/año en año 2000, con un mínimo de 0,43 en 1992 y un máximo de 3 ppm en 1998. Desde 2000, la tasa anual apenas ha cambiado. La concentración de CO<sub>2</sub> en la atmósfera es actualmente de 387 ppm, o sea el 0,0387% de la atmósfera.

#### **5.4.10. Peajes**

Se denomina peaje al pago que se efectúa como derecho para poder circular por un

camino. En la antigüedad, se llamaba portazgo a la suma que debía pagarse para cruzar cierto límite (puerta) entre dos zonas territoriales.

En términos más generales se asocia el concepto de peaje a la tasa o tarifa que se cobra a un medio de transporte terrestre, fluvial o marítimo como derecho de tránsito para utilizar la infraestructura de la respectiva vía de comunicación; por ejemplo, a los automóviles para poder circular por una autopista, o a los barcos para poder atravesar por un canal de navegación o una hidrovía. En la mayoría de los casos la vía o ruta marítima sujeta a peaje permite a los usuarios ahorrar tiempo de viaje y reducir sus costos de operación, con respecto al tránsito por vías o rutas alternas libres de peaje.

#### **5.4.10.1. Tipos**

- Peaje abierto: cada cierta distancia hay una caseta de peaje, donde se abona una cantidad.
- Peaje cerrado: al entrar en la carretera de peaje, se registra la entrada y se abona a la salida, según la longitud recorrida, sin más paradas intermedias (con peaje manual recoge una tarjeta a la entrada y cuando se usa la tele peaje no se recoge tarjeta, pues el aparato se encarga de registrar la entrada).
- Peaje anual: en algunos países, como en Suiza, los usuarios pagan anualmente una cantidad, que se acredita mediante una pegatina en el parabrisas, que les permite circular por todas las autopistas libremente.
- Aquellos que solamente la utilizan ocasionalmente (turistas), tienen que pagar la misma cantidad.

- **Peaje urbano de congestión:** Tasa, cargo o impuesto que se cobra en algunas ciudades bajo la política de tarifas de congestión, como los implantados en Buenos Aires, Estocolmo, Londres, Milán y Singapur, con el propósito de disminuir la cantidad de vehículos que acceden a una determinada zona del centro para reducir la congestión de tránsito, y en los programas más recientes, también tienen el objetivo de disminuir las emisiones de gases de efecto invernadero. Aunque no se trata de un peaje en el sentido tradicional, los medios de comunicación con frecuencia lo llaman "Peaje urbano".

Además, hay otro medio curioso de financiar las carreteras:

**Peaje sombra:** construye y financia la autopista una empresa comercial y la Administración paga el peaje de los vehículos que circulan por ella, de modo que se financia con los impuestos, pero se evita el endeudamiento de la administración a corto plazo, aunque la experiencia indica que a largo plazo puede suponer una quiebra económica.

#### **5.4.11. Multas**

Una multa es la sanción administrativa o penal consistente en un pago en dinero, a veces expresado como días de multa (cuando su pago redime la reclusión por el número correspondiente de días).

Se denomina multa coercitiva a la que se reitera por plazos determinados si personal frente al incumplimiento por parte de la persona obligada, la administración puede tomar diversas medidas: ejecución subsidiaria, multa coercitiva, compulsión sobre las personas, y la más

## CAPITULO III

### 6. MARCO METODOLOGICO

En este capítulo se describirán y analizarán el diseño y las acciones destinadas a resolver el problema planteado “Semáforos Autónomos para control de tráfico en la ciudad de Neiva”. El objetivo general es determinar el tipo de investigación, mientras que el diseño se basa en los procedimientos a realizar. Además, se determinará la unidad de estudio y las técnicas de recolección de datos.

#### 6.1. Tipo de Investigación

En este primer paso del marco metodológico se establece el tipo de investigación a realizar, este se define basándose en el objetivo general del proyecto, este planteamiento es de suma importancia ya que influye en los instrumentos para la recaudación de datos y la manera en la que se analizaran estos, existen varios tipos pero los más comunes son los siguientes: investigación descriptiva, exploratoria, analítica, comparativa, explicativa, predictiva, proyectiva, interactiva, confirmatoria y evaluativa. Para cada tipo de investigación existe un planteamiento diferente por cada autor que ha escrito sobre estas.

La Investigación descriptiva según Hurtado (2010, pág. 101) “Tiene como objetivo la descripción precisa del estudio. Este tipo de investigación se asocia al diagnóstico. En la investigación descriptiva se hace enumeración detallada de las características del evento. Las investigaciones descriptivas trabajan con uno o con más eventos de estudio”.

Según Van Dalen y Meyer (1991).

“El objetivo de la investigación descriptiva consiste en llegar a conocer las situaciones, costumbres y actitudes predominantes a través de la descripción exacta de las actividades, objetos, procesos y personas.

Su meta no se limita a la recolección de datos, sino a la predicción e identificación de las relaciones que existen entre dos o más variables”.

Según lo previamente descrito se puede establecer que la investigación realizada es del tipo descriptiva ya que al principio del proyecto fueron planteados los requerimientos para el desarrollo del proyecto y la función que tendrán los semáforos para la mejoría del flujo del tráfico vehicular en la ciudad.

A su vez, será una Investigación proyectiva según Hurtado (2010, pág. 114)

“Este tipo de investigación, consiste en la elaboración de una propuesta, un plan, un programa o un modelo, como solución a un problema o necesidad de tipo práctico, ya sea

de un grupo social, o de una institución, o de una región geográfica, en un área particular del conocimiento, a partir de un diagnóstico preciso de las necesidades del momento, los procesos explicativos o generadores involucrados y de las tendencias futuras, es decir, con base en los resultados de un proceso investigativo”.

La investigación proyectiva no solo se basa en una propuesta, también proponen alternativas de cambio y tampoco está desligada de un proceso de recolección de información y diseño ya que estos pasos son esenciales para que el resultado del proyecto permita obtener el evento deseado.

Según la UPEL, (2003, pág. 16) proyecto factible:

“Consiste en la investigación, elaboración y desarrollo de un modelo operativo viable para solucionar problemas, requerimientos necesidades de organizaciones o grupos sociales que pueden referirse a la formulación de políticas, programas, tecnologías, métodos, o procesos. El proyecto debe tener el apoyo de una investigación de tipo documental, y de campo, o un diseño que incluya ambas modalidades”

Como podemos ver este tipo de proyectos solo pasa por una fase de documentación que llamaremos descriptiva y otra fase de diseño y desarrollo que llamaremos proyectiva.

Basados en las definiciones anteriores, se puede establecer esta investigación como un proyecto factible ya que el objetivo principal es el desarrollo de un sistema que dará solución a

un problema de un grupo de usuarios de la ciudad. También es importante resaltar que esta investigación cumple con los pasos de este tipo de proyectos que son analizar, recopilar información, establecer diseño, desarrollar y evaluar, estos se ven reflejados en los objetivos del proyecto.

La presente investigación corresponde a un diseño no experimental debido a que en ningún momento se manipularon las variables en estudio. Según

Fernández, Hernández y Baptista (2006) el diseño no experimental es aquel que se realiza sin manipular deliberadamente las variables, sino que se tratan los fenómenos tal y como se dan en el contexto natural para después ser analizado.

Esta investigación coincide con un diseño no experimental longitudinal de tendencia. Según Fernández, Hernández y Baptista son aquellos que analizan cambios a través del tiempo dentro de alguna población en general. Su característica distintiva es que la atención se centra en una población.

Según esta definición es entonces no experimental longitudinal de tendencia porque no se crea ningún evento, los eventos se dan por si solos en la naturaleza y se obtienen los datos a través del tiempo observando si se reducen o no los accidentes de tráfico y otras incidencias.

## 6.2. Técnica de recolección de datos.

En este tema se establece la técnica de recolección de datos a utilizar, este paso detalla que fuente se debe utilizar para obtener los datos y como deben ser analizados, hay varios tipos de recolección entre los más populares están observación, entrevistas, sesiones de grupos de enfoque, documentos, registros, materiales y artefactos, biografías e historias.

Según Hernández, Fernández & Baptista, 2006,

“Una fuente muy valiosa de datos cualitativos son los documentos, materiales y artefactos. Nos pueden ayudar a entender el fenómeno central de estudio. Prácticamente la mayoría de las personas, grupos, organizaciones, comunidades y sociedades los producen y narran, o delinear sus historias y estatus actuales, le sirven al investigador cualitativo para conocerlos antecedentes de un ambiente, las experiencias, vivencias o situaciones y su funcionamiento cotidiano.” (Hernández, et al, 2006)

Por lo tanto, se ha decidido usar como fuente documentos y registros ya que estos pueden darnos información acerca de los problemas con la tecnología actual usada en la ciudad y poder encontrar una mejor solución a estos problemas.

A su vez Según Hernández, Fernández & Baptista, 2006, estos documentos entran en la categoría de grupales y en la subcategoría de registros en archivos públicos ya que lo

describen de la siguiente manera

“En estos podemos encontrar muchos de los documentos, materiales y artefactos generados para fines públicos (catastros, registros de la propiedad intelectual...). Los archivos pueden ser gubernamentales (nacionales o locales) o privados (por ejemplo, de fundaciones).”

Como se puede ver esta categoría es la que más encaja ya que el problema se genera en el área de lo público, siendo la ciudad la fuente principal de registros y datos para la mejor evaluación, como se evidencia según todas estas definiciones, la técnica de recolección de datos a utilizar será documentos y registros ya que se trata un problema acerca de las tecnologías actuales para manejar el flujo de tráfico vehicular en la ciudad siendo fuente registros públicos y estos a su vez de fácil obtención ya que puede tenerse acceso sin utilizarse mediante distintos medios como periódicos, reportes entre otros

### **6.3. Procedimiento metodológico.**

Esta fase consiste en acordar el proyecto a construir, la funcionalidad y los requisitos del proyecto, en ella se encuentran varias sub fases de las cuales se aplicarán algunas de ellas que fueron necesarias para la elaboración del proyecto.

### **6.3.1. Analizar el funcionamiento actual de los semáforos.**

Los semáforos trabajan mediante las señales de luz que cambian en un tiempo predefinido, algunas de estas señales son las siguientes:

- **Verde:** los vehículos tienen derecho al paso.
- **Amarillo:** Advierte a los conductores de vehículos que el estado verde está a punto de cambiar al estado rojo y por lo tanto deben asumir una conducta de prevención y reducir su velocidad,
- **Rojo:** Los vehículos deben detenerse a una distancia de dos metros del semáforo.

Flechas para giros a la izquierda o a la derecha: Los conductores de los vehículos giran a la izquierda o a la derecha, según lo indique la flecha, y de acuerdo con el color que exhiban.

### **6.3.2. Establecer los requerimientos técnicos para la realización del sistema de semáforos autónomos.**

Estos semáforos autónomos necesitan de un sistema que será el que se encargue de la toma de decisiones y del funcionamiento del mismo, para ello se hará el uso de un micro controlador Arduino, el cual nos permitirá su programación con un lenguaje de alto nivel y aprovechamos de sus bondades entre las cuales se encuentran una fácil implementación y bajo consumo eléctrico, además también es muy importante destacar que este sistema utilizara

diodos led para representar las luces del semáforo ya que estas cumplen con las capacidades energéticas, económicas, y físicas del proyecto.

#### **6.4. FASE II: ELABORACIÓN**

Esta metodología tiene como meta el desarrollo de proyectos de manera rápida y con documentación precisa y resumida por lo que esta y las siguientes fases son las más importantes.

##### **6.4.1. Recopilar los elementos necesarios para cumplir cada uno de los objetivos en la elaboración y diseño del sistema de control de tráfico vehicular.**

Sistema de toma de decisiones o microcontrolador el cual se encargará por medio de programación, de verificar el estado del sistema y tomar la mejor decisión al respecto.

- Tablero para circuitos en el cual se realizará la muestra y se aplicará el diseño planteado para el sistema.
  
- Resistencias, diodos y cableado que nos servirán en conjunto al tablero para la implementación del circuito y comprobar el funcionamiento del programa.

**6.4.2. Establecer la base de la elaboración de la arquitectura para el sistema de control de tráfico vehicular.**

**6.5. FASE III: CONSTRUCCIÓN**

Esta fase es donde se implementa el diseño establecido y se realiza la construcción del proyecto por completo.

**6.5.1. Desarrollar el sistema de control de tráfico vehicular en base a la arquitectura definida.**

- Diseñar un algoritmo que guie el desarrollo del programa.
  
- Programar el micro controlador siguiendo las ideas planteadas en el algoritmo y verificarlo con corridas en frio.

**6.6. FASE IV: TRANSICIÓN**

Esta fase final está enfocada a las pruebas y toma de notas para mejoras a futuro, además se analiza que el proyecto haya cumplido todas las expectativas iniciales.

**6.6.1. Evaluar la funcionalidad y performance del sistema construido.**

- Luego del desarrollo del programa realizar evaluaciones constantes de su funcionalidad.
  
- Tomar los resultados analizarlos y mejorar el programa desarrollado y así eliminar posibles errores.
  
- Analizar los resultados y definir mejoras para el futuro del proyecto.
  
- Analizar que el proyecto haya cumplido con las expectativas.

## CAPITULO IV

### 7. ANÁLISIS DE RESULTADOS

En el desarrollo del sistema de control de tráfico mediante semáforos autónomos se utilizó las metodologías open up. Este tipo de metodología permite el desarrollo de proyectos de manera participativa y complementaria entre los integrantes del grupo de desarrollo. La misma cuenta con 4 fases en las que se desarrolla y explica de manera resumida y directa el proceso de concepción, elaboración, construcción y transición que fueron ejecutadas según el orden especificado.

#### 7.1. Concepción del sistema

En la fase de concepción se establece el sistema a desarrollar tomando en cuenta los problemas y soluciones, se definen las expectativas y se establecen los requerimientos en torno al hardware y el software, de manera que permitan el desarrollo exitoso del sistema. A continuación en la **¡Error! No se encuentra el origen de la referencia.** Se aprecian los resultados en los siguientes gráficos Tabla 3 y Tabla 4.

**Tabla 3 Requerimientos del Hardware**

Leds	18
Diodos de Silicio	15
Resistencias 1K	6
Interruptores	6
Sensores de ultrasonido	1
Kit Arduino	1

Notas: Requerimientos del Hardware: Lenguaje ensamblador. Extraído el 20 de marzo del 2017 [http://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador)

**Tabla 4 Requerimientos de Software.**

Sistema Operatives	Windows o Linux.
Lenguaje	Todos los soportados por el arduino.

Notas: Requerimientos de Software: Lenguaje ensamblador. Extraído el 20 de marzo del 2017 [http://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador)

Dentro de las expectativas del hardware, se plantea el desarrollo de un sistema que, permita el control del tráfico vehicular mediante el uso de semáforos que detecten el nivel flujo vehicular utilizando sensores de ultra sonido.

## 7.2. Diseño del sistema

Estructura del sistema: La estructura de un sistema inteligente es muy importante, ya que nos permite identificar el propósito de su desarrollo y a su vez la estructura de cada uno de sus componentes, el fin de cada uno de estos y por qué son indispensables en el desarrollo del proyecto

Dentro del propósito del diseño del sistema de semáforo inteligente, se busca desarrollar un sistema que sin ninguna o poca interacción humana tome decisiones que le permita controlar el flujo de tráfico vehicular, en los distintos tipos de intersecciones que se pueden encontrar en las ciudades. El sistema deberá permitir la detección de vehículos y conteo de los mismos, dicha información debe ser almacenada en una memoria, para que posteriormente sea analizada por el dispositivo ya que esto le permitirá tomar las decisiones sobre el control del tráfico. Una vez tomada la decisión el dispositivo debe controlar una serie de luces las cuales indicaran el curso que debe tomar el tráfico, además debe ser capaz de manejar un sensor ultrasónico que le permitirá detectar los vehículos en las intersecciones.

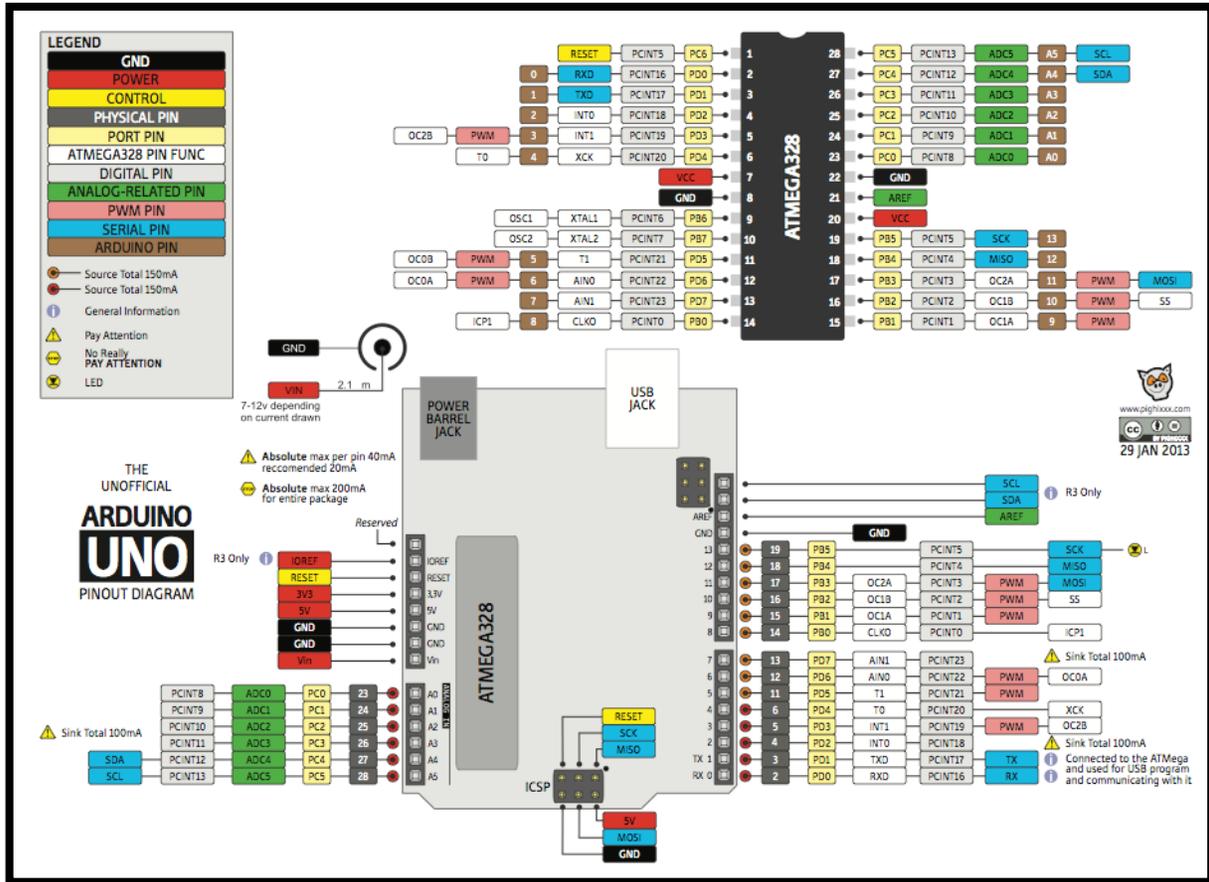
La estructura del dispositivo de control es muy importante y a que a través de este se maneja todo el sistema y se toman todas las decisiones necesarias para el control del tráfico vehicular, para este proyecto se empleó el uso del microcontrolador ATmega328, mediante el kit de distribución de arduino, sin embargo la estructura del dispositivo empleado es libre ya

que es un dispositivo de diseño y programación libre, la estructura empleada en el kit arduino se puede observar en los **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** .



**Ilustración 1 Placa Arduino UNO R3**

Notas: Ilustración 2 Placa Arduino UNO R3: Electrónica. Extraído el 3 de abril del 2017  
<http://es.wikipedia.org/wiki/Electrónica>.



**Ilustración 3 Diagrama de Conexiones Arduino UNO R3.**

Notas: Ilustración 4 Diagrama de Conexiones Arduino UNO R3: Electrónica. Extraído el 3 de abril del 2017 <http://es.wikipedia.org/wiki/Electrónica>.

Este también tiene una serie de especificaciones que también es importante resaltar ya que estas nos definen las limitaciones en cuanto a la programación y procesos soportados por el dispositivo, estas se observan en él **¡Error! No se encuentra el origen de la referencia.**

**Tabla 5 Características de Arduino UNO R3**

Micro controlador	ATmega328
Voltaje de Operación	5 V
Voltaje de Entrada (Recomendado)	7-12 V
Voltaje de Entrada (Limites)	6-20 V
Entradas y Salidas Digitales	14 (6 de estas son salidas de 8bit)
Entradas Analógicas	6
Corriente DC soportada por cada conexión de entrada y salida	40mA
Corriente DC soportada por la conexión 3.3V	50mA
Memoria Flash	32KB Pertencientes al micro controlador, 0.5KB usados por el cargador de inicio.
Memoria SRAM	2KB Pertencientes al micro controlador
EEPROM	1KB Pertenciente al micro controlador
Velocidad de reloj	16MHz

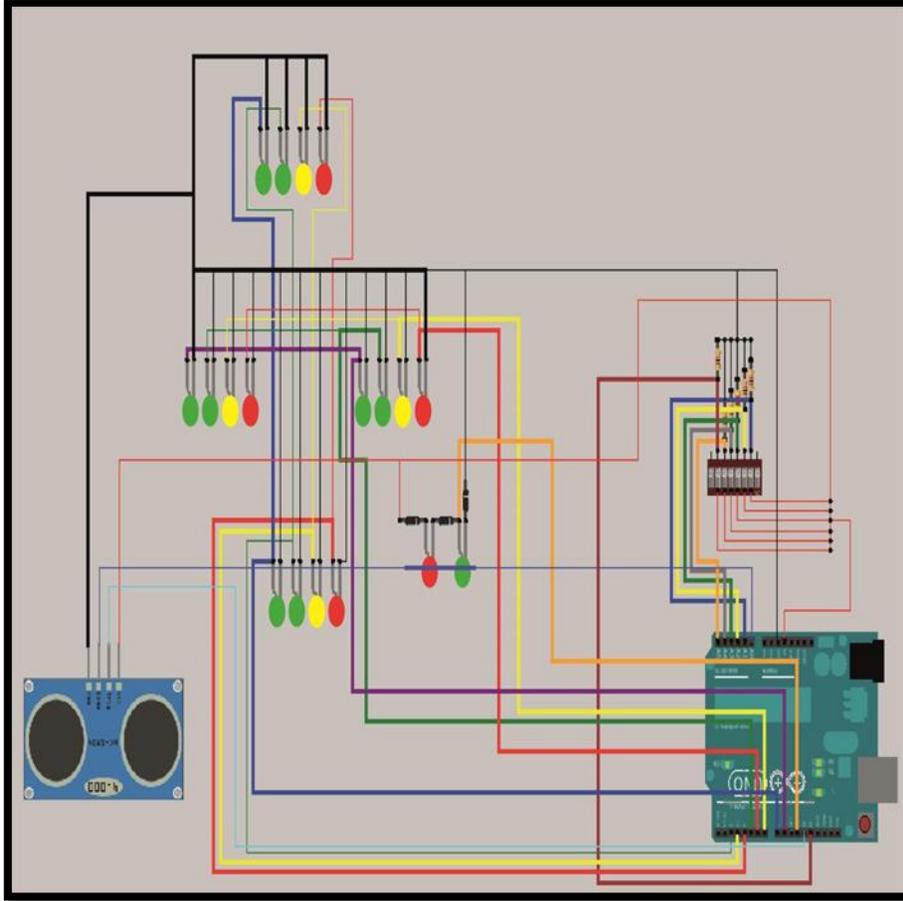
Notas: Características de Arduino UNO R3: Electrónica. Extraído el 3 de abril del 2017

<http://es.wikipedia.org/wiki/Electrónica>.

### 7.3. Diseño y estructura del circuito de semáforos

El diseño del circuito es muy importante ya que este nos ayuda a identificar el funcionamiento y la forma en la que se implantara el sistema de semáforos autónomos, este fue diseñado utilizando el programa virtual bread board, el cual permite el diseño y la programación de circuitos micro controlados.

Este circuito nos muestra cuales conectores fueron utilizados del micro controlador arduino, los componentes utilizados y la forma en la que se conectaron, el circuito puede ser observado en la **¡Error! No se encuentra el origen de la referencia.¡Error! No se encuentra el origen de la referencia.**



**Ilustración 5 Circuito del Semáforo Autónomo**

Notas: Ilustración 6 Circuito del Semáforo Inteligente: Electrónica. Extraído el 3 de abril del 2017 <http://es.wikipedia.org/wiki/Electrónica>

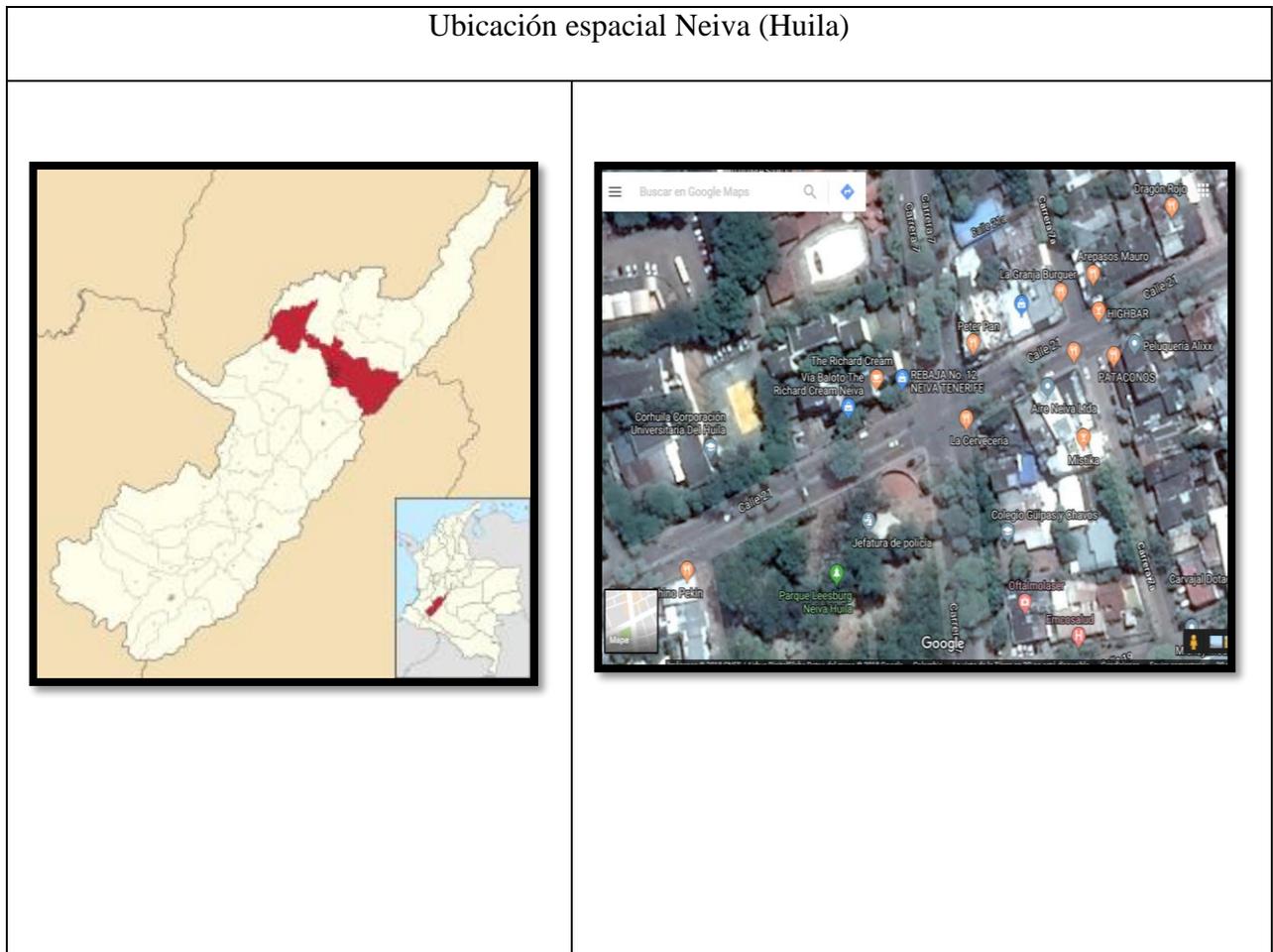
#### **7.4. Diseño y solución para las distintas intersecciones**

En las distintas intersecciones existe una estructura distinta para los semáforos, ya que en una ciudad todas las intersecciones no están estructuradas de la misma manera, es importante resaltar las posibles soluciones que tendría este sistema para cada una de ellas.

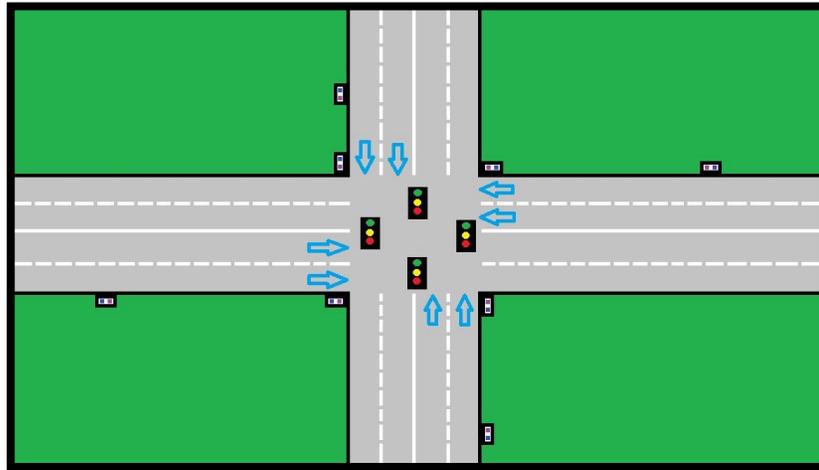
El uso de sensores de ultra sonido es una de las piezas más importantes del sistema, es por eso que lo que más resalta en la solución a las distintas intersecciones es el modo de empleo y la funcionalidad de estos.

### 7.5. Funcionamiento:

En el siguiente dibujo se muestra la posición de los sensores, dos para cada sentido el primero para leer la entrada del vehículo en la cola y el segundo la salida del vehículo de la cola, dependiendo de qué cola sea más prolongada el semáforo tomará sus decisiones para vaciarlas. Imagen 4



Fuente de elaboración google maps



**Ilustración 7 Intersección Sencilla**

Fuente: de elaboración propia.

**7.6. Intersección con desvío:**

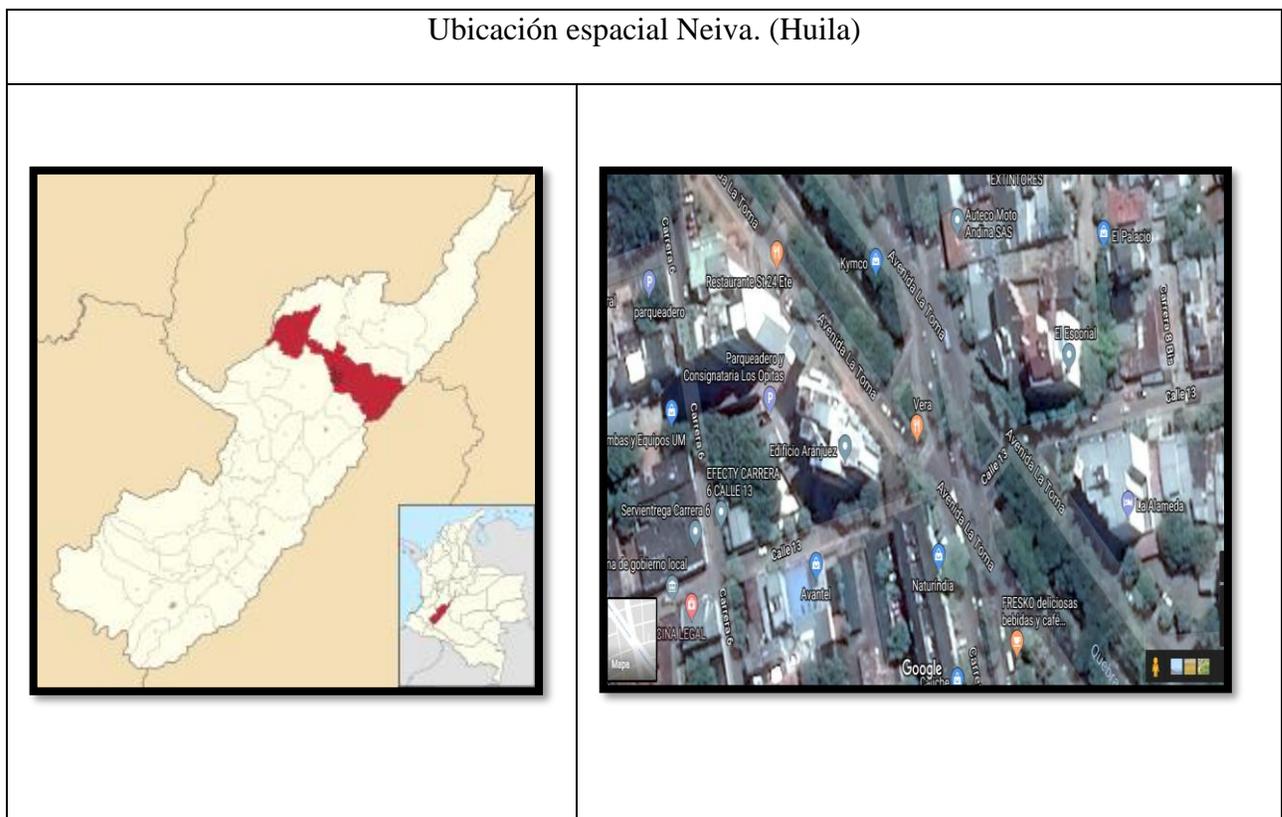
**Tabla 6 Dirección y Sensores con desvío.**

Dirección	Norte	Sur	Este	Oeste
Sensores	4	4	4	4

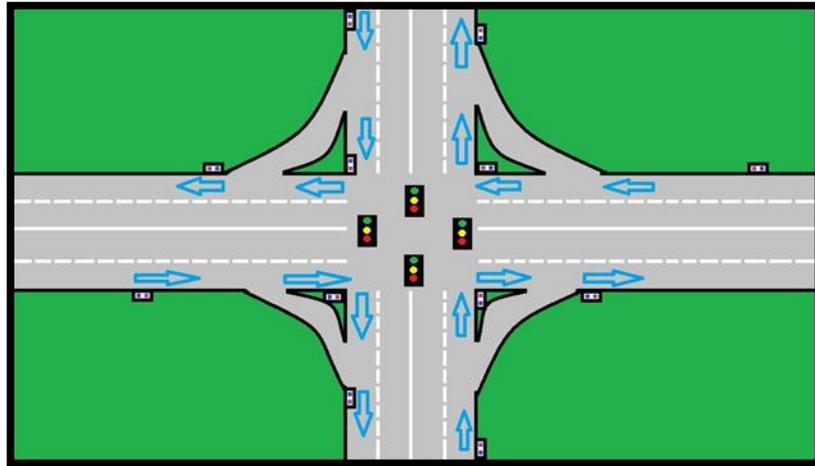
Fuente: de elaboración propia.

### 7.6.1. Funcionamiento:

Este es el caso más usual, en el gráfico se muestra la posición de los sensores, un sensor para leer la entrada del vehículo en la cola y otro la salida del vehículo de la cola, dependiendo de qué cola sea más prolongada el semáforo tomará sus decisiones para vaciarlas, en este caso se agrega un tercer sensor en cada sentido ya que los vehículos pueden tomar el desvío se coloca uno al final del desvío para tomarlo como salida de la cola ya que al no pasar sin tomar el desvío no se podría leer su salida **¡Error! No se encuentra el origen de la referencia..**



Fuente de elaboración google maps



**Ilustración 8 Intersección con Desvió.**

Fuente: de elaboración propia.

### 7.7. Intersección con desvío en forma de Y:

**Tabla 7 Dirección y Sensores Y.**

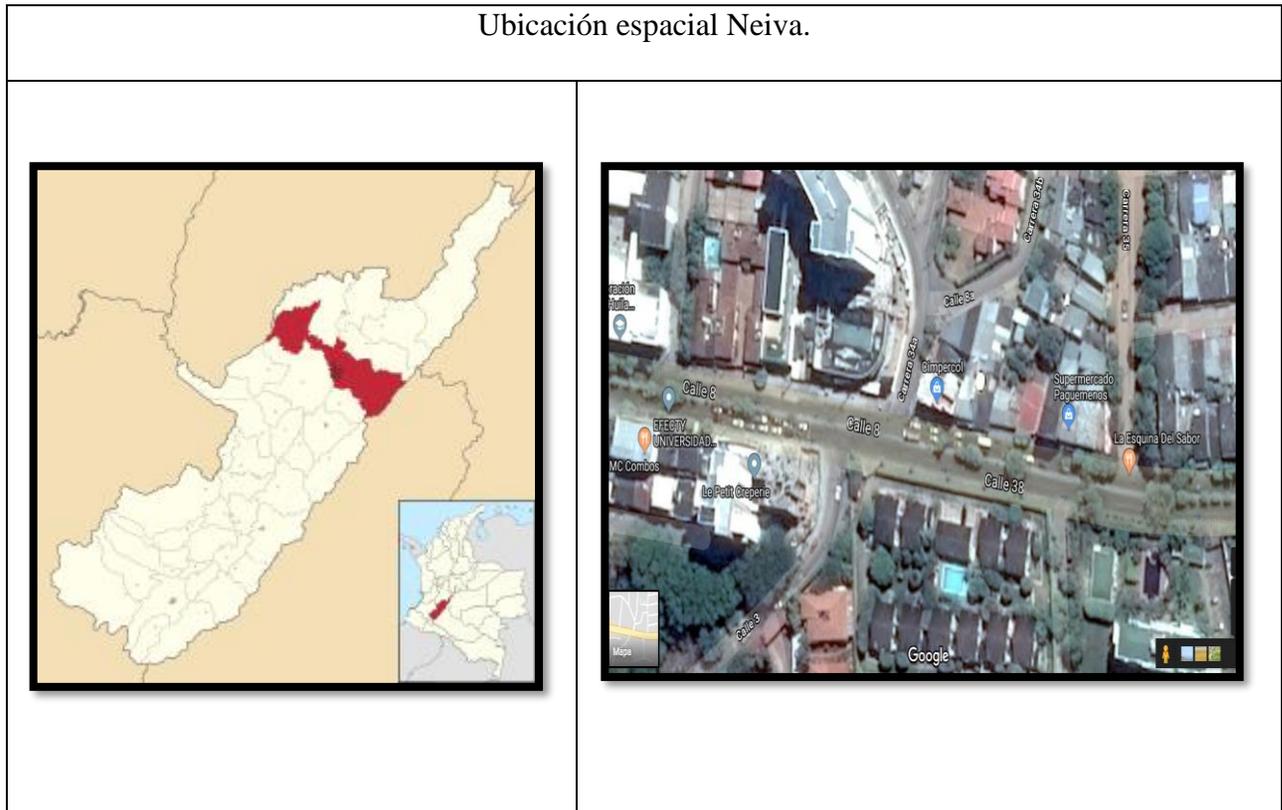
Dirección	Norte	Sur	Este	Oeste
Sensores	4	0	2	0

Fuente: de elaboración propia.

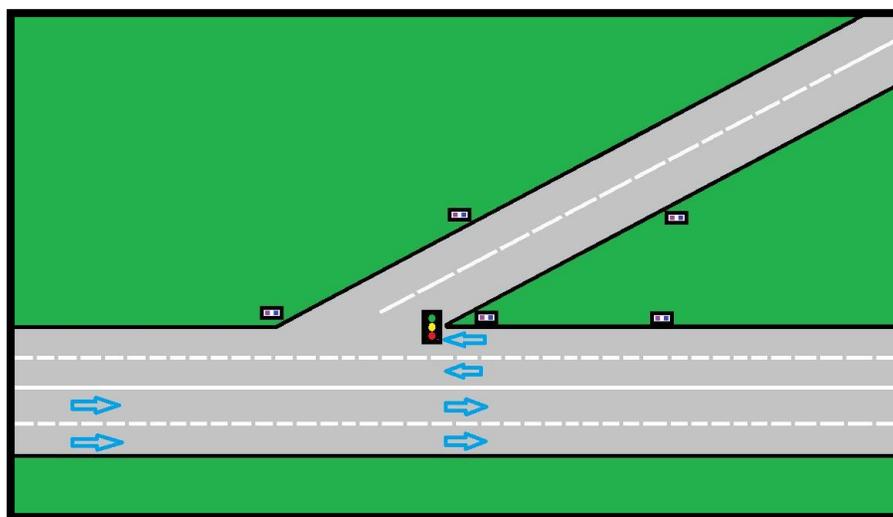
#### 7.7.1. Funcionamiento:

En el siguiente gráfico se muestra la posición de los sensores, dos sensores para los que vienen desde el este, el primero para hacer una entrada del vehículo en la cola y el otro para la salida, en el norte se colocan 4 sensores con el mismo propósito para garantizar que no ocurran accidentes por puntos ciegos Ilustración 6.

### Ubicación espacial Neiva.



Fuente de elaboración google maps



**Ilustración 9 Intersección con Desvío en Forma de Y**

Fuente: de elaboración propia.

## 7.8. Intersección en forma de T:

**Tabla 8 Dirección Y Sensores en forma T**

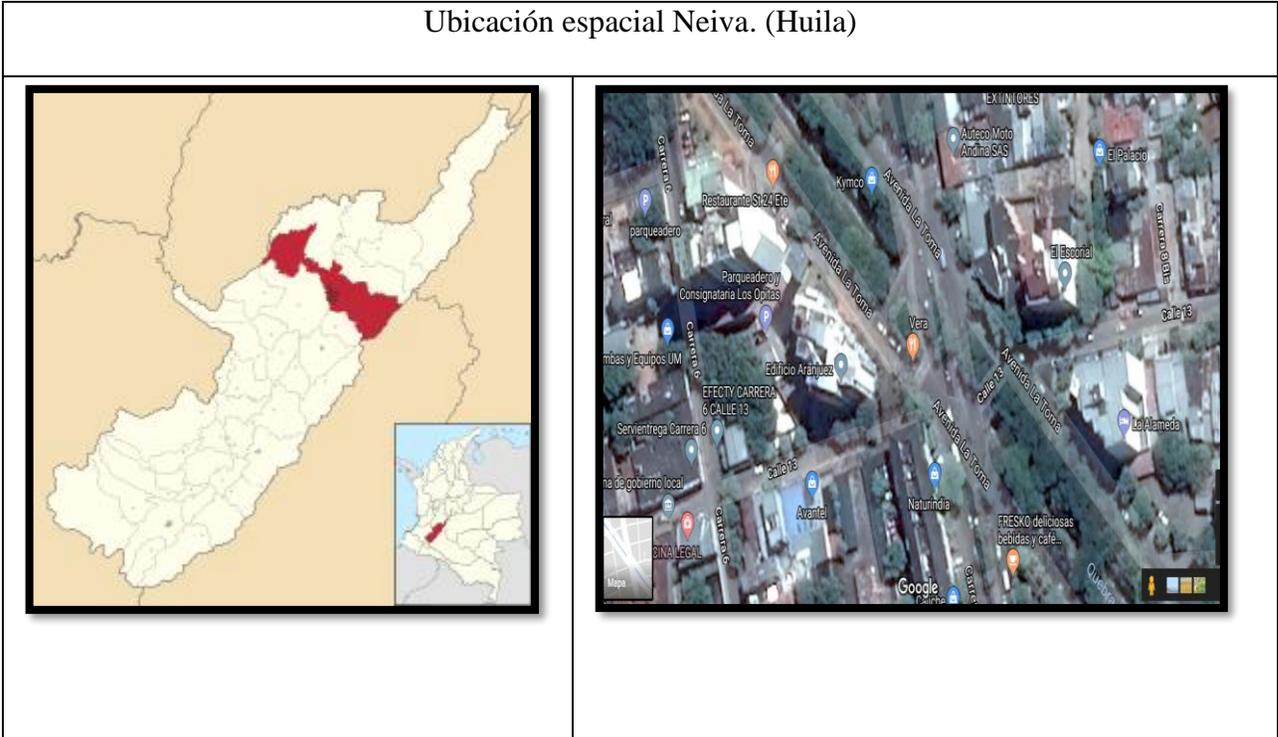
Dirección	Norte	Sur	Este	Oeste
Sensores	0	2	2	2

Fuente: de elaboración propia.

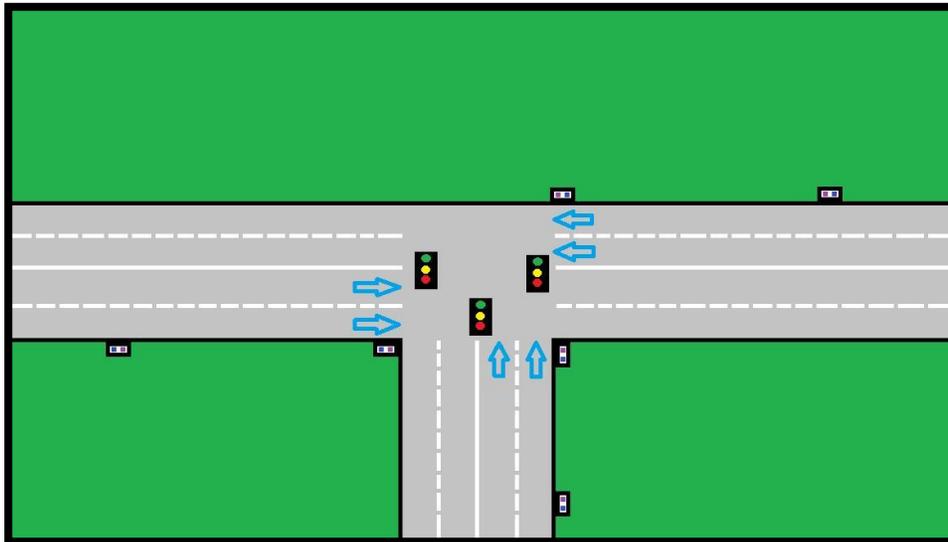
### 7.8.1. Funcionamiento:

En el siguiente gráfico se muestra la posición de los sensores, 6 sensores en total, 2 para el oeste, 2 para el este y 2 para el sur, un sensor lee la entrada del vehículo en la cola y el otro su salida de la cola, dependiendo de qué cola sea más prolongada el semáforo tomará sus decisiones para vaciarla **¡Error! No se encuentra el origen de la referencia..**

### Ubicación espacial Neiva. (Huila)



Fuente de elaboración google maps



**Ilustración 10 Intersección en Forma de T**

Fuente de elaboración propia.

## 7.9. Construcción o desarrollo del sistema inteligente

En esta fase del proyecto, se procede a llevar a cabo la creación del código y su aplicación al micro controlador con su respectivo circuito. Fueron realizadas pruebas a medida que se iba escribiendo el código para así, reducir el número de errores finales dejando como resultado final el sistema inteligente de semáforos.

En primer lugar, se nombraron los puertos a utilizar para el desarrollo y las variables necesarias, estas nos ayudaran a identificar los estados de los sensores y a almacenar el conteo de automóviles por intersección.

```
1 //Sensors
2 // (n)orth
3 // (s)outh
4 // (e)ast
5 // (w)est
6 // (p)eatonal
7 const int nbuttonPin = A0;
8 const int sbuttonPin = A1;
9 const int ebuttonPin = A2;
10 const int wbuttonPin = A3;
11 const int CnsbuttonPin = A4;
12 const int CewbuttonPin = A5;
13 const int pbuttonPin = 13;
14 //Peatonal Ligh
15 const int grpLedPin = 11;
16 //North, South Lights
17 const int gnsLedPin = 2;
18 const int ynsLedPin = 3;
19 const int rnsLedPin = 4;
20 //East, West Lights
21 const int gewLedPin = 5;
22 const int rewLedPin = 6;
23 const int yewLedPin = 7;
24 //Turn Lights
25 const int tnsLedPin = 8;
26 const int tewLedPin = 9;
27 //Trigger Pin
28 const int trigPin = 10;
29
```

**Ilustración 11 Nombrando puertos a utilizar.**

```
1  long duration, cm;  
2  int i;  
3  int lastnbuttonState = 0;  
4  int lastsbuttonState = 0;  
5  int lastebuttonState = 0;  
6  int lastwbuttonState = 0;  
7  int nbuttonState = 0;  
8  int nbuttonCounter = 0;  
9  int sbuttonState = 0;  
10 int sbuttonCounter = 0;  
11 int ebuttonState = 0;  
12 int ebuttonCounter = 0;  
13 int wbuttonState = 0;  
14 int wbuttonCounter = 0;  
15 int CnsbuttonState = 0;  
16 int CewbuttonState = 0;  
17 int pbuttonState = 0;
```

### Ilustración 12 Nombrado puertos a utilizar

Fuente: Captado del L lenguaje de programación C. junio 2017

#### 7.10. Declaración de variables

Luego de establecer estos parámetros procedemos a configurar de qué manera funcionara cada uno de los puertos, esto es muy importante ya que sin esta configuración inicial tendríamos que declarar el funcionamiento cada vez que se haga uso de los puertos, con esta función setup se ha ahorrado tiempo y líneas de código.

```
1 void setup() {  
2  
3     for (int pin = 2; pin <=11; pin++) {  
4         pinMode(pin, OUTPUT);  
5     }  
6     pinMode(grpLedPin, OUTPUT);  
7  
8     pinMode(nbuttonPin, INPUT);  
9     pinMode(sbuttonPin, INPUT);  
10    pinMode(ebuttonPin, INPUT);  
11    pinMode(wbuttonPin, INPUT);  
12    pinMode(CnsbuttonPin, INPUT);  
13    pinMode(CewbuttonPin, INPUT);  
14    pinMode(pbuttonPin, INPUT);  
15  
16    digitalWrite(rnsLedPin, HIGH);  
17    digitalWrite(rewLedPin, HIGH);  
18    digitalWrite(grpLedPin, LOW);  
19    Serial.begin(9600);  
20 }
```

**Ilustración 13 Declaración de variables.**

Fuente: Captado del L lenguaje de programación C. junio 2017

### **7.11. Configuración inicial**

Para trabajar con el sensor de ultrasonido se estableció una función que activa el sensor y envía un pulso de sonido cada 2 o más microsegundos, luego se analiza la duración y se calcula la distancia de acuerdo a una fórmula obtenida de parallax quien fabrica estos sensores.

La fórmula para calcular la distancia desde el sensor hasta el objetivo, teniendo en cuenta que la velocidad del sonido en una temperatura de 20 grados centígrados es 343 m/s se establece de la siguiente manera.

- Pasamos la velocidad a centímetros por segundo, se obtiene  $V = 34.400 \text{ cm/s}$  .

- Calculamos el tiempo con la ecuación  $Distancia = velocidad/tiempo$ , despejamos tiempo y tenemos la ecuación  $t = d/v$  reemplazamos por los valores  $t = 1cm/ (34400cm/s)$  luego  $t = 0,000029155s$ .
- Como la señal va y vuelve hace el doble de la distancia que se desea medir, por lo tanto tomara el doble de tiempo, así que mediremos el doble de tiempo luego  $t = 0,000058309 s$ , finalmente pasamos este valor a microsegundos  $t = 58,309 us$  que para efecto práctico lo aproximaremos a  $t=58us$   $t = 58 us$  en el programa.
- Tener en cuenta que la velocidad del sonido es mayor cuando aumenta la temperatura, aproximadamente  $0.6m/s$   $0.6 m/s$  por cada grado centígrado.

Entonces para encontrar la distancia del objetivo se toma la mitad de la distancia viajada, es decir primero se calcula la distancia tomando en cuenta el tiempo de retorno del pulso y luego esta se divide entre 2.

$$Cm = (t/29)$$

A demás este proceso se establece como una función en la cual se hace un print en la pantalla de serial que nos permitirá monitorear a medida que se corre el código las distancias capturadas por el dispositivo.

```
1 void SensorUltrasonido() {
2 //Enviar pulso
3 digitalWrite(trigPin, LOW);
4 delayMicroseconds(2);
5 digitalWrite(trigPin, HIGH);
6 delayMicroseconds(5);
7 digitalWrite(trigPin, LOW);
8
9 //Leer pulso
10 duration = pulseIn(nbuttonPin, HIGH);
11
12 //Convertir en distancia
13 cm = microsecondsToCentimeters(duration);
14 Serial.print(cm);
15 Serial.print("cm");
16 Serial.println();
17
18 }
19 long microsecondsToCentimeters(long microseconds)
20 {
21 // The speed of sound is 340 m/s or 29 microseconds per centimeter.
22 // The ping travels out and back, so to find the distance of the
23 // object we take half of the distance travelled.
24 return microseconds / 29 / 2;
25 Serial.print(cm);
26 Serial.print("cm");
27 Serial.println();
28 }
```

**Ilustración 14 Programación sensor ultrasonido.**

Fuente: Captado del L lenguaje de programación C. junio 2017

### **7.12. Sensor Ultrasonido**

Esta función de la mano con los condicionales nos permite detectar el paso de los automóviles a través del sensor de ultrasonidos con lo cual establecimos un tope de 15 cm para la detección.

```

1  if (cm <= 15) {
2      CruceNorteSur ();
3      north ();
4  }
  
```

**Ilustración 15 Activar a los 15 cm.**

Fuente: Captado del L lenguaje de programación C. junio 2017

## 8. Evaluación de resultados

Al realizar la evaluación del sistema se pudo concluir, que en ciertas situaciones el sistema inteligente reduce los tiempos de espera, lo que se traducen un mejor flujo de vehículos a través de las intersecciones, además este sistema se adapta a los eventos, tomando decisiones que le permiten establecer una rutina para el cambio de luces distinta, gracias a su capacidad de detección de vehículos a través de sensores, estos resultados obtenidos nos permitieron realizar un pequeño cuadro comparativo en el cual demostramos las ventajas de este sistema sobre los tradicionales .

**Tabla 9 Ventajas del sistema sobre el tradicional.**

Sistema Tradicional	Sistema Autónomo
Tiempo de espera dependiente del establecido por el fabricante.	Tiempo de espera variable y en algunos casos reducidos, debido a que el sistema realiza los cambios de luces dependiendo de la cantidad de vehículos.

<p>Rutinas para el cambio de luces siempre en funcionamiento, sin tomar en cuenta el tráfico.</p>	<p>Rutinas para el cambio de luces en espera de eventos (Todas las luces en rojo), lo que permite reducir algunos accidentes y además adaptar los cambios de luces al estado del tráfico.</p>
<p>No se detecta el paso de los automóviles.</p>	<p>Detección y conteo de automóviles que transitan cada una de las vías en la intersección, lo que permitirá una toma de decisiones por parte del dispositivo, que le permita reducir el congestionamiento.</p>

Fuente de elaboración propia.

También es muy importante destacar que este sistema nos permite realizar auditorías en vivo, lo que nos ayuda a su mejoría y reestructuración de los algoritmos, además es posible agregar nuevos dispositivos que contribuyan con una toma de decisiones más precisa.

## CONCLUSIONES

El presente trabajo de investigación obtuvo como resultado la creación de un sistema de control de semáforos autónomos para el control del tráfico vehicular en la ciudad de Neiva, permitiendo una mejoría del flujo del tráfico gracias a la capacidad de detección de vehículos y toma de decisiones que dan prioridad a las avenidas con mayor flujo de tráfico.

Dando respuesta a los objetivos que en primer lugar se establecieron, sustentándonos en la metodología Open Up compuesta por cuatro fases, se llevó a cabo un análisis de los elementos necesarios para crear el sistema de semáforos autónomos y se determinó la funcionalidad del mismo. Posteriormente llevándonos por la elaboración de un diseño y una planificación base de todos los requerimientos del sistema. Dando paso a la fase de construcción donde se genera el sistema que será probado para verificar que está listo para ser implementado, tomando en cuenta el diseño base generado en la fase previa y el objetivo del proyecto.

Una vez construido el sistema de semáforos autónomos se hicieron una serie de pruebas en un ambiente de intersecciones simulado para determinar si se han alcanzado las expectativas del proyecto. Luego de haber culminado el desarrollo del proyecto se citaron ciertas recomendaciones, basándonos en las pruebas simuladas que ayudaran a mantener el sistema funcionando óptimamente y a su futura mejoría.

## RECOMENDACIONES

Luego de la culminación del sistema de semáforos autónomos para el control del tráfico vehicular, es pertinente citar algunas recomendaciones que permitirán que este se aproveche al máximo y sea posible su ampliación, para las intersecciones de las ciudades y sus usuarios.

- Se recomienda interconectar los semáforos de las intersecciones en red, mediante el uso del shield Arduino para redes o a través de sus mismos puertos, para así tener un mayor y mejor control del tráfico en las intersecciones seguidas o paralelas.
- Se recomienda el mantenimiento periódico a las conexiones y leds del sistema de semáforos, para que se pueda brindar a la ciudad un servicio de calidad, donde los semáforos estén siempre en óptimas condiciones.
- Se recomienda la implementación propuesta con respecto a los sensores para cada intersección, y a que esto permitirá que el conteo de los automóviles sea siempre preciso y reduce las posibilidades de error.
- Se recomienda la ampliación del sistema, para la inclusión de artefactos de detección de imágenes y detectores de velocidad, esto permitirá aplicar un sistema de monitoreo al tráfico que facilitará la aplicación de infracciones.
- Utilizar el presente proyecto de investigación como antecedente de otras investigaciones u proyectos similares, logrando así su actualización y mejora.

## BIBLIOGRAFIA

2. D. Vexelman (2011). Lima tiene 800 semáforos inteligentes. Extraído el 9 de noviembre de 2012 <http://peru21.pe/noticia/1323755/lima-cuenta-800-semaforos-inteligentes>
3. Yaiza Martínez (2008). Semáforos inteligentes reducen la contaminación y agilizan el tráfico. Extraído el 9 de noviembre de 2012 [http://www.tendencias21.net/Semaforos-inteligentes-reducen-la-contaminacion-y-agilizan-el-trafico\\_a2074.html](http://www.tendencias21.net/Semaforos-inteligentes-reducen-la-contaminacion-y-agilizan-el-trafico_a2074.html)
4. Eduardo Solís (2011). Ventas al Público. Extraído el 10 de noviembre de 2012 <http://www.amia.com.mx/ventas.html>
5. La historia del control del tráfico. Extraído el 10 de noviembre de 2012 <http://www.trafficlogix.com.mx/historia.php>
6. Lenguaje ensamblador. Extraído el 5 de diciembre del 2012 [http://es.wikipedia.org/wiki/Lenguaje\\_ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador)
7. Tránsito Vehicular. Extraído el 5 de diciembre del 2012 [http://es.wikipedia.org/wiki/Tránsito\\_vehicular](http://es.wikipedia.org/wiki/Tránsito_vehicular)
8. Gestión de tráfico. Extraído el 5 de diciembre del 2012 [http://es.wikipedia.org/wiki/Gestión\\_de\\_tráfico](http://es.wikipedia.org/wiki/Gestión_de_tráfico)
9. Información Peaje. Extraído el 5 de diciembre del 2016 <http://es.wikipedia.org/wiki/Peaje>
10. Información Multa. Extraído el 5 de diciembre de 2016

<http://es.wikipedia.org/wiki/Multa>

11. Semáforo. Extraído el 3 de abril del 2016 <http://es.wikipedia.org/wiki/Semáforo>
12. Congestión vehicular. Extraído el 3 de abril del 2017  
[http://es.wikipedia.org/wiki/Congestión\\_vehicular](http://es.wikipedia.org/wiki/Congestión_vehicular)
13. Sensor. Extraído el 5 de abril del 2017 <http://es.wikipedia.org/wiki/Sensor>
14. Accidente de tráfico. Extraído el 3 de abril del 2013  
[http://es.wikipedia.org/wiki/Accidente\\_de tráfico.](http://es.wikipedia.org/wiki/Accidente_de_tráfico)
15. Emisiones de CO2. Extraído el 3 de abril del 2017  
[http://es.wikipedia.org/wiki/Emisiones\\_de\\_CO2](http://es.wikipedia.org/wiki/Emisiones_de_CO2)
16. Microprocesador. Extraído el 3 de abril del 2017  
<http://es.wikipedia.org/wiki/Microprocesador>
17. Micro controlador Extraído el 3 de abril del 2017  
<http://es.wikipedia.org/wiki/Microcontrolador>
18. Electrónica. Extraído el 3 de abril del 2017  
<http://es.wikipedia.org/wiki/Electrónica>
19. Sistema Inteligente. Extraído el 3 de abril del 2013  
[http://es.wikipedia.org/wiki/Sistema inteligente.](http://es.wikipedia.org/wiki/Sistema_inteligente)
20. David Rodríguez Julián Rincón (2011). Modelo, modelar, modelo del sistema viable, factibilidad y viabilidad. Extraído el 11 de febrero de 2013  
<http://www.slideshare.net/toofymen/modelo-informtico#btnNext>
21. Teoría del tráfico vehicular. Extraído el 11 de Febrero de 2017  
<http://www.conocimientosweb.net/portal/article2733.html>

22. Daniel Seijo (2008). Semáforos Inteligentes. Extraído el 12 de Noviembre de 2017

<http://www.motorpasion.com/tecnologia/semaforos-inteligentes>

23. Semáforos Inteligentes. Extraído el 13 de Noviembre de 2017

<http://trabajosi.wikispaces.com/SEMAFOROS+INTELIGENTES>

24. Swarco's American Holding. Gestión de tráfico urbano. Extraído el 13 de

Noviembre de 2017 <http://www.swarco.com/americanas->

[es/Productos/Gesti%C3%B3n-de-tr%C3%A1fico-urbano](http://www.swarco.com/americanas-es/Productos/Gesti%C3%B3n-de-tr%C3%A1fico-urbano)

25. Hernández Sampieri, Fernández Calos y Baptista (2003), Metodología de la Investigación, editorial McGraw-Hill, Bogotá.