

**DISEÑO DE UN SISTEMA DE SEMAFORIZACIÓN INTELIGENTE EN EL
SECTOR DE LA FUENTE LUMINOSA EN CÚCUTA**

CARLOS ANDRES PEREZ COMBARIZA

UNIVERSIDAD SIMON BOLIVAR

PROGRAMA DE INGENIERIA DE SISTEMAS

SAN JOSE CUCUTA

2018

**DISEÑO DE UN SISTEMA DE SEMAFORIZACIÓN INTELIGENTE EN EL
SECTOR DE LA FUENTE LUMINOSA EN CÚCUTA**

CARLOS ANDRES PEREZ COMBARIZA

Trabajo de investigación presentado en la asignatura de Formación Investigativa III

JOSÉ GERARDO CHACÓN RANGEL

TUTOR

Línea de investigación:

Innovación y Nuevas Tecnologías

UNIVERSIDAD SIMON BOLIVAR

PROGRAMA DE INGENIERIA DE SISTEMAS

SAN JOSE CUCUTA

2018

CONTENIDO

	Página
TÍTULO	14
INTRODUCCION	14
CAPÍTULO 1. PROBLEMA DE INVESTIGACIÓN	15
Planteamiento del Problema	15
Formulación del Problema	17
Justificación	18
Objetivos	20
<i>General</i>	20
<i>Específicos</i>	20
CAPÍTULO 2. MARCO REFERENCIAL	21
Antecedentes	21
Marco Teórico	25
Marco Conceptual	28
Marco Legal	30
CAPÍTULO 3. DISEÑO METODOLÓGICO	33

	4
Paradigma	33
Enfoque	33
Nivel y diseño de Investigación	34
Población y Muestra	35
Cuadro de operacionalización de variables	36
Técnicas e instrumentos de recolección de datos	36
Técnicas de procesamiento y análisis de datos	37
CAPÍTULO 4. ANÁLISIS DE LA INFORMACIÓN	38
Resultados	38
Análisis	41
CAPÍTULO 5. SELECCIÓN DEL OPERADOR INTELIGENTE	44
Determinar las técnicas de inteligencia artificial para los semáforos inteligentes	44
<i>Máquinas de soporte vectorial</i>	45
<i>Lógica difusa</i>	46
<i>Redes neuronales</i>	47
Elección de un operador inteligente para la semaforización inteligente	49
CAPITULO 6. SINCRONIZACION NUEVA PARA LOS SEMAFOROS	54
Etiquetado de vías, intersecciones y semáforos de la zona de la fuente luminosa	54
Asignación de prioridades	60
<i>Primer método</i>	61
<i>Segundo método</i>	62

	5
<i>Tercer método</i>	62
Paso a paso de la nueva sincronización	64
<i>Paso 1</i>	64
<i>Paso 2</i>	65
<i>Paso 3</i>	66
<i>Paso 4</i>	67
<i>Paso 5</i>	68
CAPITULO 7. DESARROLLO DEL SIMULADOR DE ESCRITORIO	69
Metodología de desarrollo	69
Sprints	71
<i>Sprint 1</i>	71
<i>Sprint 2</i>	72
<i>Sprint 3</i>	80
<i>Sprint 4</i>	86
<i>Sprint 5</i>	90
<i>Sprint 6</i>	95
<i>Sprint 7</i>	100
<i>Sprint 8</i>	103
Pruebas	113
CONCLUSIONES	118
RECOMENDACIONES	120
REFERENCIAS BIBLIOGRÁFICAS	121
ANEXOS	130

LISTA DE TABLAS

	Página
Tabla 1. Operacionalización de variables	36
Tabla 2. Datos obtenidos de la espera en rojo entre la Diagonal Santander y calle octava	38
Tabla 3. Datos obtenidos de la espera en rojo entre la avenida cero y calle octava	39
Tabla 4. Datos obtenidos de la espera en rojo entre la Diagonal Santander y avenida cero	39
Tabla 5. Datos obtenidos del paso en verde entre la Diagonal Santander y calle octava	39
Tabla 6. Datos obtenidos del paso en verde entre la avenida cero y calle octava	40
Tabla 7. Datos obtenidos del paso en verde entre la Diagonal Santander y avenida cero	40
Tabla 8. Datos obtenidos sobre los semáforos	41
Tabla 9. Comparativa de operadores inteligentes	50
Tabla 10. Normalización de las entradas y salidas	92
Tabla 11. Máxima cantidad de automóviles en las horas del día	94
Tabla 12. Matriz modelo de aprendizaje para las neuronas de la capa de entrada	96

Tabla 13. Matriz modelo de aprendizaje para las neuronas de la capa de salida	97
Tabla 14. Pesos sinápticos de las neuronas de las capas de entrada	98
Tabla 15. Pesos sinápticos de las neuronas de la capa de salida del paso 1	98
Tabla 16. Pesos sinápticos de las neuronas de la capa de salida del paso 2	99
Tabla 17. Pesos sinápticos de las neuronas de la capa de salida del paso 3	99
Tabla 18. Pesos sinápticos de las neuronas de la capa de salida del paso 4	99
Tabla 19. Pesos sinápticos de las neuronas de la capa de salida del paso 5	99
Tabla 20. Primera prueba del simulador	113
Tabla 21. Segunda prueba del simulador	114
Tabla 22. Tercera prueba del simulador	114
Tabla 23. Cuarta prueba del simulador	114
Tabla 24. Quinta prueba del simulador	115
Tabla 25. Sexta prueba del simulador	115
Tabla 26. Séptima prueba del simulador	115
Tabla 27. Octava prueba del simulador	116
Tabla 28. Novena prueba del simulador	116
Tabla 29. Decima prueba del simulador	116

LISTA DE FIGURAS

	Página
Figura 1. Composición de una neurona	51
Figura 2. Neurona artificial	52
Figura 3. Enumeración del sector 1	57
Figura 4. Enumeración del sector 2	58
Figura 5. Enumeración del sector 3	59
Figura 6. Primer paso de la sincronización	63
Figura 7. Segundo paso de la sincronización	64
Figura 8. Tercer paso de la sincronización	65
Figura 9. Cuarto paso de la sincronización	66
Figura 10. Quinto paso de la sincronización	67
Figura 11. Fases de la metodología de desarrollo planteada para el proyecto	70
Figura 12. Prototipo del inicio del simulador	73
Figura 13. Prototipo simulación del primer paso	74
Figura 14. Prototipo resultados del primer paso	74
Figura 15. Prototipo simulación del segundo paso	76

Figura 16. Prototipo resultados del primer paso	76
Figura 17. Prototipo simulación del tercer paso	77
Figura 18. Prototipo resultados del primer paso	77
Figura 19. Prototipo simulación del cuarto paso	78
Figura 20. Prototipo resultados del cuarto paso	78
Figura 21. Prototipo simulación del quinto paso	79
Figura 22. Prototipo resultados del quinto paso	79
Figura 23. Triangulo de distancia, velocidad y tiempo	81
Figura 24. Prototipo definitivo del simulador	84
Figura 25. Modelo de redes neuronales del primer paso	87
Figura 26. Modelo de redes neuronales del segundo paso	87
Figura 27. Modelo de redes neuronales del tercero paso	88
Figura 28. Modelo de redes neuronales del cuarto paso	88
Figura 29. Modelo de redes neuronales del quinto paso	89
Figura 30. Elementos naturales de una neurona artificial	90
Figura 31. Función de activación sigmoidea	91
Figura 32. Función de activación lineal	91
Figura 33. Normalización por rangos	91

Figura 34. Despeje de X en la normalización

92

LISTA DE IMÁGENES

	Página
Imagen 1. Mapa de las vías de la zona de la fuente luminosa	55
Imagen 2. Sector 1 avenida Diagonal Santander con calle 8	57
Imagen 3. Sector 2 avenida 0 con calle 8	58
Imagen 4. Sector 3 avenida Diagonal Santander con avenida 0	59
Imagen 5. Interface del simulador	104
Imagen 6. Semáforo en verde	105
Imagen 7. Semáforo en rojo	105
Imagen 8. Semáforo en amarillo	105
Imagen 9. Semáforo en rojo y amarillo	105
Imagen 10. Inicio del simulador	106
Imagen 11. Paso uno del simulador	107
Imagen 12. Entre el paso uno y el paso dos del simulador	107
Imagen 13. Paso dos del simulador	108
Imagen 14. Entre el paso dos y el paso tres del simulador	108
Imagen 15. Paso tres del simulador	109

Imagen 16. Entre el paso tres y el paso cuatro del simulador	109
Imagen 17. Paso cuatro del simulador	110
Imagen 18. Entre el paso cuatro y el paso quinto del simulador	110
Imagen 19. Paso quinto del simulador	111
Imagen 20. Entre el paso quinto y el paso uno del simulador	111

LISTA DE ANEXOS

	Página
Anexo 1. Código de aprendizaje de las neuronas de la capa de entrada	130
Anexo 2. Código de la clase principal del aprendizaje de las neuronas de la capa de salida	132
Anexo 3. Código de la clase objeto del aprendizaje de las neuronas de la capa de salida	132
Anexo 4. Código de la clase de las capas de entrada del simulador	134
Anexo 5. Código de la clase de las capas de salida del simulador	136
Anexo 6. Código de la clase principal del simulador	137

TITULO

Diseño de un sistema de semaforización inteligente en el sector de la fuente luminosa en
Cúcuta.

INTRODUCCION

En el mundo hoy en día vive un gran auge de la tecnología, la ciencia, la medicina y un largo fin de áreas. La inteligencia artificial, ha sido en parte una de las herramientas con más desempeño en estas áreas, se ve un gran mundo de posibilidades en que poder ser utilizadas, en el caso particular de los semáforos, la inteligencia artificial es una de los mejores complementos a la hora de poder controlar el flujo vehicular en las ciudades metropolitanas de un país, como Colombia en vía de poder mejorar su infraestructura en materia tecnológica el auge de los semáforos inteligentes en las ciudades es cada vez una realidad, en la que poco a poco se suman ciudades como lo es Cúcuta, la cual tiene un gran potencial en materia de emprendimiento y poder así utilizar la inteligencia artificial para el beneficio de los ciudadanos en materia de movilidad, en la cual los semáforos inteligentes puedan descongestionar las vías de la ciudad de Cúcuta que se vive una gran cantidad de automóviles los cuales la infraestructura vial se queda cada día más pequeña, siendo los semáforos inteligentes una de las mejores soluciones en materia de infraestructura tecnológica en la ciudad. El diseño de un nuevo sistema de semaforización en la zona de la fuente luminosa ayudará enormemente la transitabilidad en dicha zona por lo cual se busca realizar este proyecto.

CAPITULO 1. PROBLEMA DE INVESTIGACIÓN

Planteamiento del problema

En las grandes ciudades del mundo se ve una problemática con respecto al tráfico vehicular y tráfico peatonal, se tiene una alta demanda de nuevas tecnologías que ayuden a disminuir las congestiones vehiculares que se generan en dichas ciudades, Roble, Ñañez y Quijano (2009) afirman

Debido al rápido crecimiento de los sistemas de tráfico urbano en términos de demanda y complejidad, el estudio de este tipo de sistemas ha adquirido cada vez más importancia tanto en entornos académicos, como por parte de administraciones gubernamentales alrededor del mundo. (p.60)

Este proyecto nace de la problemática que se percibe en las ciudades metropolitanas, dicha problemática trata de la gran congestión vehicular a las horas de las 12:00 M y las horas 06:00 PM. Salcedo, Pedraza y Hernández (2006) definen

(...) como consecuencia la necesidad de un óptimo sistema de control que permita un ágil desplazamiento y que reduzca entre otras el alto consumo de combustibles, la alta contaminación ambiental causada por el desprendimiento de monóxidos propio de los automotores y el nivel de accidentalidad. (p.61)

En la ciudad se vive un alto crecimiento de vehículos, según La Opinión (2016)

En Cúcuta y municipios del área metropolitana las autoridades de Tránsito no conocen una cifra real del parque automotor venezolano que circula por las calles. En las cuentas

de los empresarios figura que en Cúcuta hay 185 mil carros, y de estos 120 mil son venezolanos. (p.1)

Donde se ve, aunque sea solo una aproximación que, Cúcuta posee una cantidad bastante grande de automóviles no incluyendo motocicletas, que vendría a ser un tema aparte se puede definir la avenida Diagonal Santander en la ciudad de Cúcuta, es la arteria principal de la ciudad lo cual debe lidiar con la cantidad masiva de autos, donde se vive en las horas 12:00 M y 06:00 PM un gran flujo de vehículos. Sean colombianos o venezolanos.

La problemática de la contaminación es un tema serio en la ciudad, según Redacción Nacional (2015)

los resultados del Índice de Calidad Ambiental Urbana (ICAU) (...) evidenció que Cúcuta –con una población superior a 500 mil habitantes– obtuvo el índice de calidad ambiental más bajo, al ‘rajarse’ en los indicadores de estado y gestión que reflejan la sostenibilidad ambiental de las poblaciones. (p.1)

Sabiendo que la contaminación ambiental engloba muchas ramas, como por basura, auditiva, hídrica etc. Se centra como punto importante la contaminación auditiva, la cual no afecta de manera física a una persona, sino que afecta al estado de la persona, su ser, ya que en los momentos de atascos o en la propia vía, se vive la alta contaminación auditiva, en gritos, en bocinas de los carros, en motores, música a alto volumen, debido a la congestión vehicular se pueden presentar este tipo de contaminación, en la ciudad de Cúcuta.

También la sincronización de los semáforos influye en toda esta problemática, ya que cada intersección de la ciudad de Cúcuta que posea semáforos, estos funcionan con tiempos fijos, lo cual en la fuente luminosa, al tener calles relativamente cortas, se nota la falta de una

sincronización para esa zona, ya que pueden tener los semáforos un alto tiempo en verde, pero si avanzan los automóviles, encontrándose con un semáforo en rojo en la siguiente intersección, lo cual hace que se genere atascos en esta zona

Las limitaciones del proyecto se pueden denominar en el espacio de la ciudad de Cúcuta donde los semáforos son un tema complicado porque no se escoge ideas innovadoras, ya por presupuesto o por simple falta de información y también por problemas políticos, en todo lo que se dice de transporte puede ser bastante delicado ya que el alcalde o gobernantes que están actualmente en el mandato son los que aprueban cambios o si dejan todo como está. En el tema conceptual, la ciudad de Cúcuta vive un problema bastante grande, que es sobre los refugiados de la crisis de Venezuela que, al ser una ciudad de frontera, también se vive esa situación. En Cúcuta también se vive que hay un número considerable de motos, automóviles registrados en Venezuela, al no estar la frontera abierta para automotores, entonces están represados los autos en la ciudad, haciendo congestión vehicular en la ciudad. En la parte de temporal se vive la situación de que el proyecto debe ser terminado en el año 2018.

Formulación del problema

Por faltas de buena estructuración en las vías y por falla o falta de una buena semaforización para esta ciudad se ve reflejado una gran duda, la cual se plantea en la pregunta: ¿Qué sistema de semaforización se puede diseñar para el sector de la fuente luminosa en la ciudad de Cúcuta que mejore las condiciones de transitabilidad vehicular?

Justificación

Con este proyecto se busca diseñar un sistema de semaforización que permita mejorar el flujo vehicular en la zona de la fuente luminosa en la ciudad de Cúcuta para así evitar congestiones y mejorar de cierto modo la calidad de vida tanto de los transeúntes como de los conductores.

La congestión, sobre todo en la parte central de la ciudad, es algo de todos los días, y se tiene conocimiento de que la labor de controlar el flujo vehicular es muy tediosa y dura para mano de obra humana debido a las condiciones de trabajo tales como el abrasante calor del medio día lo cual puede afectar incluso la calidad de vida e integridad de la persona que lleva a cabo dicho trabajo, así que se tomó la iniciativa de gestionar un ente no humano que llevara a cabo dicha labor.

Este proyecto parte de la necesidad de gestionar e implementar nuevas tecnologías en la ciudad para no solo mejorar la calidad de vida, sino que también se busca mejorar la eficiencia en cuanto a flujo y tiempo de los conductores permitiéndoles llevar a cabo sus labores diarias a tiempo y reduciendo a la larga costos en cuanto a reparación de semáforos y la contratación de personal humano para controlar la congestión de vehículos.

Este dispositivo de semaforización debe cumplir con todas las necesidades viales que tiene la ciudad, como autobuses, buses intermunicipales. Y a veces vehículos de tracción animal y/o humana.

La ciudad necesita un cambio urgente del sistema de semaforización, al ser una ciudad que se puede considerar mediana, pero su estructura vial está muy desfasada a su tiempo, por falta de espacio, por lo ambiental, por paradigmas que se ven en que, si hay más calles, más

gente optara por comprar un vehículo para poder transportarse y así la ciudad pueda colapsar por tantos vehículos.

El proyecto buscará la mejor manera por opinión del autor, de diseñar un nuevo sistema de semaforización inteligente en esta zona, y que a su vez este proyecto puede hacerse realidad en un prototipo más adelante que se pueda realizar.

Con este proyecto se busca realizar un diseño de un sistema de semaforización inteligente que permita mejorar el flujo vehicular en la ciudad de Cúcuta en el sector de la fuente luminosa para así evitar congestiones y mejorar de cierto modo la calidad de vida tanto de los ciudadanos como de los conductores.

Este proyecto significa un nuevo entorno de paradigmas. De dejar de pensar en micro soluciones, a una solución mayor para una problemática que afecta a la ciudad, para tener un mejor orden en la ciudad y querer un buen futuro para la ciudad de Cúcuta, se ve que en para las ciudades metropolitanas de los países se está implementado una muy buena tecnología, ya que solucionan un gran problema vial que son los atascos, se utiliza una rama de la ingeniería de sistemas llamada inteligencia artificial(IA), la cual demuestra ser muy fiable en el ámbito de tener múltiple opción para asignar los tiempos en cada semáforo.

Objetivos

General

Diseñar un sistema de semaforización inteligente en el sector de la fuente luminosa en Cúcuta.

Específicos

Seleccionar el operador inteligente a implementar haciendo revisión documental en literatura especializada que se ajuste a la zona de la fuente luminosa

Estructurar una sincronización válida para el sistema de semaforización para la zona de la fuente luminosa

Desarrollar software de escritorio fundamentado en metodología Scrum para la simulación de la zona de la fuente luminosa.

CAPÍTULO 2. MARCO REFERENCIAL

Antecedentes

Bances, M., y Ramos. M., (2014). Semáforos inteligentes para la regulación del tráfico vehicular. *Ingeniería: Ciencia, Tecnología e Innovación*, 1(1), 37-45.

La Investigación desarrolla un Sistema de Semáforo Inteligente (SSI), basado en lógica difusa, que, según la densidad vehicular capturada por cámaras web, permiten organizar los cambios de luces en función de las condiciones que se presenten en la zona.

La razón de este trabajo es regular el tráfico vehicular con un Sistema de Semáforos Inteligentes que otorguen tiempos variables dependiendo de la densidad vehicular en tiempo real. De esta manera el estudio disminuirá en gran medida la congestión vehicular que actualmente genera caos vehicular en la Ciudad de Chiclayo. Este trabajo es una aplicación dentro del campo de la Inteligencia Artificial, específicamente dentro de Lógica difusa, está basado en visión por computador, cámaras web que permiten la entrada de datos, lenguaje de programación Python, para el procesamiento de imágenes algoritmos de visión, como es OpenCV y Highgui, así como del Microcontrolador PIC 18F2550 que permiten en gran medida disminuir la congestión como principal propósito de la investigación.

Se escoge esta referencia porque demuestra que los semáforos que existen actualmente son, de alguna manera poco eficientes y que a su vez se intenta mejorar estos sistemas de semaforización ya que ayuda bastante en materia de contaminación y de congestión

vehicular. En referente a el diseño del sistema de semaforización demuestra hacia donde debe ir enfocado el proyecto.

Salcedo, O., Pedraza, L., Hernández, C., (2006). *Modelo de Semaforización Inteligente para la Ciudad de Bogotá*. Revista científica y tecnológica de la facultad de ingeniería, universidad distrital francisco José de caldas 11(2), 61-69

Se presenta el diseño de un modelo de tráfico vehicular, el cual examina el trafico existente en una vía principal de Bogotá a través de una serie de semáforos, y a partir de esto se sincroniza el tiempo de duración y de desfase de los semáforos, utilizando para ello el Sistema de Inferencia Difusa Basado en Redes Adaptativas e intentando mantener a su vez la velocidad máxima de los vehículos permitida en la vía. El modelo es simulado en el software Matlab y se evalúan los resultados a nivel macroscópico con el modelo de temporización fija que funciona actualmente en Bogotá.

Se ve que en Colombia el cambio del sistema de semaforización es urgente y se presenta ya como un reto en la ciudad de Bogotá y se ve como un gran aporte en materia de infraestructura y en tema de tecnología. En el proyecto ayuda mucho en el reto que es diseñar un nuevo sistema de semaforización para Cúcuta, viendo la necesidad de una ciudad capital.

Guzmán, R., & Álvaro, W. (2013). *Elaboración e implementación de una maqueta prototipo de semáforo inteligente para la intersección de dos avenidas*. Quito, 2013.

En el presente proyecto se elaboró una maqueta prototipo de semáforo inteligente, en el cual se indica el principio de funcionamiento de dichos dispositivos, empleando para tal finalidad una réplica de las Avenidas Naciones Unidas y Amazonas a escala 1:500. El proyecto emplea sensores de tipo infrarrojo identificados como receptores y emisores, estos

se encuentran ubicados en la maqueta a lo largo de las avenidas que forman la intersección; cada receptor y emisor constan de línea de vista, los mismos que son controlados a través de circuitos independientes dotados con microcontroladores. El circuito principal recibe y administra las señales provenientes de los circuitos de recepción y emisión, este último circuito emite una señal a la frecuencia de 38Khz, la misma que es recibida por los distintos receptores, los cuales muestran en un display LCD el estado en el que encuentran las avenidas, es decir si existe o no presencia vehicular, para posteriormente encender las luces de los semáforos en función del flujo vehicular presente en ese momento. La información que se muestra en el LCD esta manejada en base a códigos, los cuales representan cada situación vehicular. El sistema adicionalmente funciona en modo temporizado, el cual activa su funcionamiento de manera manual o automática.

De aquí se puede visualizar que existen tiempos estándares máximos y mínimos los cuales se deben incluir en la elaboración del algoritmo de repartos de tiempo, después de todo la meta de los semáforos inteligentes es mejorar la calidad de vida de las personas. No hacer los tiempos de cada semáforo no sea muy exagerado a las horas más críticas de la ciudad de manera automovilística.

Celis, J., & Escobar, C., y Sepúlveda, S., y Castro, S., & Medina, B., y Ramírez, J. (2016). Control adaptativo para optimizar una intersección semafórica basado en un sistema embebido. *Ingeniería y Ciencia*, 12 (24), 169-193.

Para optimizar el flujo vehicular en una intersección vial se diseñaron un algoritmo de control adaptativo y una base de datos que apoya la depuración del rendimiento del controlador, ambos alojados en el sistema embebido Raspberry Pi B+. El desempeño del

algoritmo fue evaluado con un instrumento virtual, que emuló una intersección semafórica de la ciudad de Cúcuta, esto es, los sensores magnetorresistivos, el proceso de encendido en las luces de los semáforos y el flujo vehicular. La manipulación de los tiempos de encendido en las luces de los semáforos, aumentó el flujo vehicular hasta 5.5 % y, disminuyó el tiempo máximo de espera del vehículo para avanzar hasta 28 segundos y el largo de fila hasta un 18 %. Con base en el caso de estudio, se puede inferir que es posible integrar el control adaptativo y los sistemas embebidos como herramientas de software y hardware para mejorar el funcionamiento en los sistemas de regulación vial.

Este artículo viene muy de la mano de que sea en Cúcuta donde, ya ha habido estudios para la implementación de semáforos inteligentes y la ayuda de cómo utilizar la inteligencia artificial como método de ayuda al flujo vehicular para la depuración y el desempeño del tráfico.

Pedraza, L., Hernández, C., López, D., (2012). *Control de tráfico vehicular usando ANFIS*. Revista chilena de ingeniería 20(1) p.79-88.

Diferentes estrategias para el control del tráfico urbano se han presentado a lo largo del tiempo. Este artículo presenta el diseño de un modelo de tráfico vehicular, el cual examina el tráfico existente en una vía a través de una serie de semáforos. A partir de este modelo se sincronizan los tiempos de duración y de desfase de los semáforos, utilizando para ello el Sistema de Inferencia Difusa Basado en Redes Adaptativas (ANFIS). El modelo es simulado y los resultados se evalúan a nivel macroscópico con el modelo de tiempos fijos, que funciona actualmente en Bogotá-Colombia.

Con este artículo se tiene una referencia que fue gestionada para la ciudad de Bogotá y que al parecer es una de las óptimas, utilizando redes neuronales, ya que al usar este método fue probado en una ciudad tan grande y poblada como lo es la capital de Colombia.

Marco teórico

Para el proyecto se toman muchas teorías significativas en torno a la inteligencia artificial, la cual se toma la teoría de Richard Ernest Bellman que en 1978 dice López (2007) ““La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como la toma de decisiones, resolución de problemas, aprendizaje...” (Bellman, 1978)” (p.1). Lo cual habla sobre la automatización que a veces se ve en la zona de la fuente luminosa, ya que, cuando hay mucho tráfico en la zona, actúan los policías de tránsito para intentar solucionar la congestión en la zona.

Donde se ve reflejado el alto desarrollo por parte de la inteligencia artificial, para el poco tiempo que ha pasado, ha sido muy bien empleado, donde ha tenido un auge en estos años y donde seguramente seguirá creciendo, gracias a esto se fundamenta la razón del proyecto, la automatización de tomar las decisiones en los tiempos de los semáforos, del cual deben generarse unos cálculos, que es eso lo que nos hace razonar y actuar en función de todo. No se debe quedar en solo en la toma de decisiones, sino que se está dando una solución a una gran problemática mundial, la contaminación, la congestión vehicular, etc.

La inteligencia artificial toma un entorno amplio en este caso, ya que los semáforos inteligentes tienen varias ramas para poder tomar en estos casos, en este caso se tomarán varios operadores inteligentes, los cuales cumplen con la teoría de Bellman, ya que se trabaja

de forma que el operador inteligente estimule un tiempo del semáforo en las intersecciones en la fuente luminosa.

En la parte del desarrollo del software se plantea alguna metodología que se acorde con lo planeado en el proyecto, mirando primero las metodologías de desarrollo tradicionales las cuales, según Navarro, Fernández & Morales (2013)

En las metodologías tradicionales se concibe un solo proyecto, de grandes dimensiones y estructura definida; se sigue un proceso secuencial en una sola dirección y sin marcha atrás; el proceso es rígido y no cambia; los requerimientos son acordados de una vez y para todo el proyecto, demandando grandes plazos de planeación previa y poca comunicación con el cliente una vez ha terminado ésta. (p. 31)

Dan una perspectiva de la utilización de las metodologías tradicionales de desarrollo, lo cual, para el proyecto, se necesita una metodología que sea ágil, la cual permita el direccionamiento dinámico en la que se pueda manejar el proyecto, según Leiva & Villalobos (2015)

Estos métodos se centran en otras dimensiones, distintas que, en los métodos tradicionales, como por ejemplo el factor humano o el producto software. Esta es la filosofía de los métodos ágiles, las que dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. (p.475)

Lo cual estos tipos de metodologías para este proyecto se verían como la mejor, para lo cual se debe buscar una metodología de desarrollo ágil según Ávila, Meneses & De la Cruz (2013)

Cada metodología de desarrollo de software, ya sea tradicional o ágil, posee ventajas y desventajas. Se comparte el criterio de que las metodologías tradicionales como RUP no son viables en proyectos donde la satisfacción del cliente a corto tiempo es importante; y en la que dicha satisfacción se logra entregándole una versión ejecutable en un tiempo sumamente corto, aunque no posea todas las funcionalidades solicitadas. (p.60)

Lo cual se ve la clara desventaja de las metodologías de desarrollo de software tradicional, pero eso no quiere decir que la metodología de desarrollo ágil sea la mejor en todos los casos, eso se dará a conocer al final del proyecto, lo cual se va a tomar la metodología de desarrollo ágil para el proyecto. Una de ellas de la que se basa en metodología de desarrollo ágil es la metodología Scrum Sobrevilla, Hernández, Velasco y Soriano (2017) “El proceso de Scrum está definido por una serie de iteraciones de duración fija llamadas sprints, que pueden durar desde una hasta cuatro semanas según las preferencias y/o necesidades del equipo.” (p.3), lo cual conviene al desarrollo del proyecto en manera de distribución de cargo a través del tiempo, lo cual es de la mejor manera la metodología de desarrollo más adecuada. También tenemos la metodología ágil Extreme Programing (XP) según Madariaga, Rivero, Leyva (2016)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, que promueva el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y proporcione un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (p. 5)

También es buena candidata para ser escogida como metodología de desarrollo para el proyecto, ya que se busca rapidez a la hora del desarrollo del software del proyecto, pero se escoge Scrum como metodología, ya que, en la teoría, es la metodología de desarrollo de software ágil, que más se adapta a las necesidades del proyecto, lo cual genera de manera cíclica el desarrollo del software a través del tiempo y con el llevar un control y una serie de objetivos que se cumplen con la ejecución completa del proyecto.

Marco conceptual

Semáforo: Retoma el concepto de Reinoso y Tocaín (2009). “Es el elemento más importante en la planificación, diseño y operación de la ingeniería de tráfico. es un dispositivo que regula el flujo vehicular” (p.6) donde se ve que es un instrumento para mantener el orden y el control de las vías, para que no haya caos, En este proyecto se contextualiza de vital importancia a la hora de emplear este proyecto como los semáforos inteligentes que es una nueva tecnología que busca mejorar la congestión vehicular y flujo vehicular

Semáforo inteligente: los semáforos inteligentes son los mismos semáforos conocidos, pero estos a su vez varían el tiempo de cada luz, según Martínez (2012)

Es considerado semáforo inteligente a todo aquel capaz de tomar decisiones dependiendo de una serie de parámetros de entrada (flujo de vehículos, velocidad media, identificar la calle, entre otros). En otras palabras, el comportamiento del mismo, es de forma dinámica y se ajusta de acuerdo a varios parámetros. Actualmente, existen diferentes tecnologías implementadas, pero aún no se encuentra la mejor fórmula al problema principal de las ciudades, la congestión vehicular. Este enfoque tiene varios aspectos positivos, que se irán

tratando más adelante. Este semáforo se presenta para solucionar varios problemas de tránsito comunes en todo el mundo. (p.2)

Dando esta definición a la perspectiva que se vive, dando la solución más factible para la ciudad de Cúcuta.

Inteligencia artificial: El concepto de la inteligencia artificial es una rama de la ingeniería donde se puede decir que las maquinas son capaces de tomar decisiones propias sin que una persona sea la que la tome por él. Según Hardy (2001)

Las aplicaciones de la IA se sitúan principalmente en la simulación de actividades intelectuales del hombre. Es decir, imitar por medio de máquinas, normalmente electrónicas, tantas actividades mentales como sea posible, y quizás llegar a mejorar las capacidades humanas en estos aspectos. (p.3).

Por medio de esta podremos ceder casi el control de los semáforos así mismos.

Flujo vehicular: Según Reinoso y Tocaín (2009) “es el resultado del movimiento o flujo de vehículos en una determinada vía, calle, o autopista. En las grandes ciudades el diario vivir de la gente se encuentra ligado al tráfico vehicular” (p.3) se puede tener encuentra que el tráfico, hace que presente atasco en las vías, y mucho más grave en accidentes. Según Urrego, Calderón, Forero y Quiroga (2009)” Las ciudades de gran tamaño se asocian con congestiones de tráfico vehicular, ya sea porque la infraestructura no creció o porque no se desarrolló de manera adecuada con respecto al aumento en el número de vehículos.” (p.8) se tienen en cuenta el tamaño de la ciudad, puede afectar el tráfico, se puede entender que un método para poder mejorar el tráfico.

Marco legal

En la parte jurídica se ve grandes aportaciones que da la ley para la identificación de semáforos en la ciudad de Cúcuta, según el código de tránsito de Colombia en capítulo XIII: procedimientos de control de tránsito y en los artículos siguientes. El artículo 117. Clasificación de semáforos. Los semáforos son elementos para regular y ordenar el tránsito y se clasifican en:

- Semáforos para control de vehículos.
- Semáforos para peatones.
- Semáforos especiales.
- Semáforos de aproximación a cruces de transporte masivo, trenes y guardarrieles.
- Semáforos direccionales, intermitentes y otros.

Donde el proyecto utilizara los semáforos para control de vehículos para el diseño y en lo que se basara siempre.

En el artículo 118. Simbología de las señales luminosas. Las señales luminosas para ordenar la circulación son las siguientes:

- Roja: Indica el deber de detenerse, sin pisar o invadir la raya inicial de la zona de cruce de peatones. Si ésta no se encuentra demarcada, se entenderá extendida a dos metros de distancia del semáforo. El giro a la derecha, cuando la luz está en rojo está permitido, respetando la prelación del peatón. La prohibición de este giro se

indicará con señalización especial. Las autoridades de tránsito, en su jurisdicción, podrán autorizarlo.

- Amarilla: Indica atención para un cambio de luces o señales y para que el cruce sea desalojado por los vehículos que se encuentran en él o se abstengan de ingresar en el cruce aun disponiendo de espacio para hacerlo. No debe iniciarse la marcha en luz amarilla, ni incrementarse la velocidad durante ese lapso.

No se debe ingresar en amarillo a la intersección y si un vehículo ya está en la intersección en luz amarilla mantendrá la prelación hasta culminar el cruce.

- Verde: Significa vía libre

Se ve el funcionamiento normal de un semáforo los tipos de luces lo que cual se refleja en las vías, donde en Cúcuta se vive un caso particular, se ve que no se respeta casi este artículo en la vida cotidiana, por afán o por falta de cultura de los ciudadanos.

En el Parágrafo 1°. En ciertas situaciones o en determinados horarios, las autoridades de tránsito, en su jurisdicción y mediante resolución motivada, podrán utilizar la intermitencia de la luz de los semáforos. Esta intermitencia se da en amarillo y en rojo. El amarillo se utilizará para las vías con prelación y el rojo para todas las que acceden a éstas. La señal intermitente roja se asimila a una señal de PARE. Este funcionamiento es esencial en horas de la madrugada, ya que, en los semáforos de tiempo fijo, el conductor debe esperar relativamente mucho tiempo sin haber automóviles en las otras intersecciones.

Y terminando con el artículo 119. Jurisdicción y facultades. Sólo las autoridades de tránsito, dentro del territorio de su jurisdicción, podrán ordenar el cierre temporal de vías, la

demarcación de zonas, la colocación o retiro de señales, o impedir, limitar o restringir el tránsito o estacionamiento de vehículos por determinadas vías o espacios públicos. Esto puede generar bastante congestión en especial en las vías alternas más cercanas a la jurisdicción, en el sector de la fuente luminosa es centro de Cúcuta, por ende, es alta la probabilidad de accidentalidad en la zona y las autoridades pueden influir en este término.

CAPÍTULO 3. DISEÑO METODOLÓGICO

Paradigma

Visto en lo anterior el paradigma que se manejó es el positivista, según Meza (2010)

En el enfoque positivista se parte de teorías previamente seleccionadas de la cual se extraen, por un enfoque hipotético deductivo, hipótesis que se desea contrastar en la investigación para confirmarlas o desecharlas. En la concepción dialéctica del conocimiento la teoría que se acepta en el marco de la investigación es la que tiene relación más directa con el tema investigado o la que se desprende de ello. (p.3)

Es positivista ya que trabajó a partir de conceptos y utilización ya preestablecida.

Enfoque

En la metodología se optó por usar un enfoque cuantitativo de acuerdo a lo descrito por Angulo (2011) "Las características que destacan en la metodología cuantitativa, en términos generales es que esta elige una idea, que transforma en una o varias preguntas de investigación relevantes" (p.115) y se ve necesariamente la aplicación cuantitativa de dicha tecnología en una ciudad con un crecimiento bastante exponencial, al realizar el proyecto podría mejorar el control vehicular por medio de los semáforos inteligentes.

Para el desarrollo de software se tomó la metodología Scrum para el desarrollo del simulador, porque es la mejor se acopla para el desarrollo ágil del software para optimizar los tiempos de trabajo, este se realizó en Sprint de manera semanal.

Nivel y Diseño de investigación

El nivel de esta investigación es descriptivo debido a que, según de Hernández, Fernández y baptista (2006)

miden, evalúan o recolectan datos sobre diversos conceptos (variables), aspectos, dimensiones o componentes del fenómeno a investigar. En un estudio descriptivo se selecciona una serie de cuestiones y se mide o recolecta información sobre cada una de ellas, para así (valga la redundancia) describir lo que se investiga. (p. 102)

El diseño es no experimental en el libro de Hernández, Fernández y baptista (2006)

Podría definirse como la investigación que se realiza sin manipular deliberadamente variables. Es decir, se trata de estudios donde no hacemos variar en forma intencionales variables independientes para ver su efecto sobre otras variables. 10 que hacemos en la investigación no experimental es observar fenómenos tal como se dan en su contexto natural, para después analizarlos. (p.205)

Debido a que, el proceso de recolección no afectó al ambiente, sino que se recolectó la información de manera de que no afecte el entorno. Una vez hecha la toma de mediciones de parte de los tiempos de los semáforos en verde y del flujo vehicular, ya que en la zona central de la metrópolis se tendrán que llevar a cabo correlaciones entre esas dos variables y se deberá hacer un paralelo entre los resultados obtenidos y los que arrojarían los semáforos controlados por Inteligencia Artificial. Las variables son los tiempos de los semáforos y el flujo vehicular.

Población y muestra

La población del proyecto es la cantidad de semáforos que hay en la ciudad de Cúcuta en el año 2015 según el diario la opinión se calcularon unas 104 intercepciones que tienen semáforos, dando como resultado la cantidad de 436 semáforos, en el cual en el diario dice que hay 25 semáforos defectuosos, y la semaforización va de mal en peor. Para eso se seleccionó una muestra intencional Hernández, Fernández y baptista (2006)

El investigador puede elegir ciertos casos, analizarlos, y más adelante seleccionar casos adicionales para confirmar o no los primeros resultados. Posteriormente puede elegir casos homogéneos y luego heterogéneos para probar los límites y alcances de sus resultados. Más adelante, muestras en cadena, luego casos extremos. Finalmente, analizar casos negativos (intencionalmente buscar casos contradictorios, excepciones, que le permitan otros puntos de vista y comparaciones). La riqueza de datos es mayor porque se expresan múltiples "voces". (p. 666)

de dichos semáforos, se tomó los semáforos de la zona de la fuente luminosa la cual tiene 11 semáforos, los cuales se manejan actualmente de manera independiente de su entorno. Se utilizará esta zona porque es un sector muy concurrido de la ciudad de Cúcuta, ya que en ella se genera mucho tráfico vehicular que afecta la movilidad de la ciudad.

Cuadro de operacionalización de variables

Tabla 1. Operacionalización de variables

Variable	Sub variables o dimensiones	Indicadores	Instrumentos
Movilidad	Semáforos	Cantidad de semáforos en cada intersección	Ficha de observación
		Tiempo de cada semáforo en verde	
	Flujo vehicular	Cantidad de vehículos esperando en cada semáforo en rojo en horas valle	
		Cantidad de vehículos que circulan en verde en cada semáforo en horas valle	
		Cantidad de vehículos esperando en cada semáforo en rojo en horas pico	
		Cantidad de vehículos que circulan en verde en cada semáforo en horas pico	

Fuente. Propia

Técnicas e instrumentos de recolección de datos

Se utilizó de tipo de ficha de observación directa en cada intersección que comprende la fuente luminosa, según Benguría, Alarcón, Valdés, Pastellides y Gómez (2010) “son aquellas en las que el observador se pone en contacto directa y personalmente con el hecho o fenómeno a observar.” (p.9) y será una observación directa ya que el integrante del proyecto irá a cada uno de los semáforos de esta zona, el cual tomará: La cantidad de semáforos que hay en cada una de las intersecciones, y a su vez tomará el tiempo de cada uno de los semáforos, y el tiempo de avanzar (verde), y también debe tomar la cantidad de automóviles que llegan por cada uno de los semáforos cuando está en rojo, en diferentes horas de los días laborales (lunes a viernes).

Técnicas de procesamiento y análisis de datos

La interpretación de los factores, se debe tener en cuenta la variable, y medir cada uno de los factores, lo cual depende de la naturaleza de las variables, una vez se interpreten los valores de los factores de la variable, se toma una variable representativa y unas variables suplentes, los cuales en el proyecto lo cual la sub variable representativa es el flujo vehicular y la suplente son los semáforos.

CAPÍTULO 4. ANÁLISIS DE LA INFORMACIÓN

Resultados

En los resultados de la recolección de datos se obtuvieron los según las técnicas escogidas para la apropiación del sector de la fuente luminosa de la ciudad de Cúcuta. Lo cual se utilizaron la ficha de observación para poder la recolección de la información lo cual fue de manera Estructurada o Científica, según Díaz (2011) “Observar un objetivo claro, definido y preciso: el investigador sabe que es lo que desea observar y para que quiere hacerlo, lo cual implica que debe preparar cuidadosamente la observación.” (p.7) Para lo cual se utilizó un formato para la obtención de datos de la variable movilidad, previamente analizado teniendo en cuenta que valor se querían tomar (véase en la tabla 1). Para lo cual se tomó un formato por cada intersección de la fuente luminosa. Lo cual tienen inmersas los espacios solicitados para la investigación (véase en las tablas 2,3,4,5,6,7,8).

Tabla 2. Datos obtenidos de la espera en rojo entre la Diagonal Santander y calle octava

Semáforos de la intersección entre la Diagonal Santander y calle octava.		Itaraciones de los semáforos										Promedios	
		Cantidad de autos esperando en el semaforo en horas valle					Cantidad de autos esperando en el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5	Horas valle	Horas pico
1	Semáforo 1.1	3	2	1	2	6	15	23	19	23	22	2.8	20.4
2	Semáforo 1.2	7	5	2	5	6	7	6	7	7	5	5	6.4
3	Semáforo 1.3	16	11	10	8	15	15	14	19	20	22	12	18
4	Semáforo 1.4	10	11	5	8	14	14	10	13	16	20	9.6	14.6

Fuente. Propia.

Tabla 3. Datos obtenidos de la espera en rojo entre la avenida cero y calle octava

		Itearcciones de los semáforos										Promedios	
Semáforos de la intersección entre la avenida cero y calle octava		Cantidad de autos esperando en el semaforo en horas valle					Cantidad de autos esperando en el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5		
5	Semáforo 2.1	6	11	6	11	7	10	12	15	12	9	8.2	11.6
6	Semáforo 2.2	7	0	2	2	5	16	13	12	17	11	3.2	13.8
7	Semáforo 2.3	16	11	10	8	15	14	15	13	10	11	12	12.6

Fuente. Propia.

Tabla 4. Datos obtenidos de la espera en rojo entre la Diagonal Santander y avenida cero

	Semáforos de la intersección entre la Diagonal Santander y avenida cero.	Itearcciones de los semáforos										Promedios	
		Cantidad de autos esperando en el semaforo en horas valle					Cantidad de autos esperando en el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5	Horas valle	Horas pico
8	Semáforo 3.1	10	14	11	13	11	15	19	21	24	14	11.8	18.6
9	Semáforo 3.2	10	8	11	9	11	27	30	22	21	19	9.8	23.8
10	Semáforo 3.3	16	11	10	8	15	22	30	20	25	18	12	23
11	Semáforo 3.4	10	11	5	8	14	10	12	15	10	21	9.6	13.6

Fuente. Propia.

Tabla 5. Datos obtenidos del paso en verde entre la Diagonal Santander y calle octava

	Semáforos de la intersección entre la Diagonal Santander y calle octava.	Itearcciones de los semáforos										Promedios	
		Cantidad de autos que pasaron el semaforo en horas valle					Cantidad de autos que pasaron el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5	Horas valle	Horas pico
1	Semáforo 1.1	10	20	13	14	7	10	11	9	12	13	12.8	11
2	Semáforo 1.2	12	9	10	7	8	6	5	4	5	9	9.2	5.8
3	Semáforo 1.3	14	15	12	11	13	10	5	7	8	7	13	7.4
4	Semáforo 1.4	13	15	16	12	15	8	6	5	9	10	14.2	7.6

Fuente. Propia.

Tabla 6. Datos obtenidos del paso en verde entre la avenida cero y calle octava

		Itearciones de los semáforos										Promedios	
	intersección entre la avenida cero y calle octava	Cantidad de autos que pasaron el semaforo en horas valle					Cantidad de autos que pasaron el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5		
5	Semáforo 2.1	10	9	11	8	12	3	2	5	2	3	10	3
6	Semáforo 2.2	6	8	10	9	7	5	3	4	6	7	8	5
7	Semáforo 2.3	13	9	12	7	10	10	9	8	7	11	10.2	9

Fuente. Propia.

Tabla 7. Datos obtenidos del paso en verde entre la Diagonal Santander y avenida cero

	intersección entre la Diagonal Santander y avenida cero.	Itearciones de los semáforos										Promedios	
		Cantidad de autos que pasaron el semaforo en horas valle					Cantidad de autos que pasaron el semaforo en horas pico						
		1	2	3	4	5	1	2	3	4	5	Horas valle	Horas pico
8	Semáforo 3.1	15	12	14	16	12	16	14	15	11	10	13.8	13.2
9	Semáforo 3.2	12	15	16	11	12	6	7	7	9	10	13.2	7.8
10	Semáforo 3.3	12	9	7	6	12	9	8	5	9	9	9.2	8
11	Semáforo 3.4	22	12	9	10	11	3	2	5	6	4	12.8	4

Fuente. Propia.

Tabla 8. Datos obtenidos sobre los semáforos

Intersección	Numeración	Semáforos	Tiempo en verde (segundos)
Diagonal Santander y calle octava	1	Semáforo 1.1	65
	2	Semáforo 1.2	20
	3	Semáforo 1.3	40
	4	Semáforo 1.4	35
Avenida cero y calle octava	5	Semáforo 2.1	30
	6	Semáforo 2.2	20
	7	Semáforo 2.3	20
Diagonal Santander y avenida cero	8	Semáforo 3.1	40
	9	Semáforo 3.2	40
	10	Semáforo 3.3	25
	11	Semáforo 3.4	25

Fuente. Propia.

Análisis

Una vez tomado los datos se hizo su respectivo análisis lo cual demostró valores tan interesantes como la mayor cantidad de autos en espera en rojo en hora pico (véase en la tabla 4), lo cual demuestra que es 30 automóviles esperando, tomando como referencia para futuros cálculos y decisiones en las determinaciones de los semáforos, también la mayor cantidad de vehículos en horas valle (véase en las tablas 2 y 4), es de 16, también para tener en cuenta

para su uso, y por ende el valor mínimo de autos que siempre va a ser en los semáforos es de 0, ya en (véase en la tabla 3), hay un semáforo en rojo el cual no tiene automóviles.

Para entender el problema de la diagonal Santander se puede ver (véase en las tablas 2, 3 y 4), la cantidad de vehículos que están esperando en el semáforo rojo en horas valle es menor a la cantidad de automóviles que esperan en el semáforo en rojo cuando es hora pico, lo cual se ve un incremento en sus promedios en los autos que esperan, y el problema radica en que, (véase las tablas 5, 6 y 7) los automóviles que circulan por la diagonal Santander reduce la cantidad de automóviles que circulan en horas pico de manera elevada cambiando drásticamente el panorama de esta zona en cuestión. Dependiendo de que las calle no son lo suficientemente extensas como para cubrir la demanda de espacio por parte de flujo vehicular en esta zona. En horas valle el flujo vehicular actúa de cierta manera, pero cuando es hora pico el flujo vehicular baja, de ello radica el problema, el cual se ve que aumenta en horas pico los automóviles en los semáforos en rojo y a su vez la circulación se torna más lenta de la que es en horas valle, lo cual genera congestión vehicular en esta zona.

Una vez hablado del flujo vehicular, se debe hablar sobre otra problemática de la zona, lo cual son los semáforos de esta, lo que sucede es que estos semáforos (véase en la tabla 8) el tiempo para cada semáforo es constante, quiere decir que no varía a lo largo del día, lo cual es un problema a la hora de tratar con la gran cantidad de flujo vehicular, lo cual se sostiene que las intersecciones no se manejan correctamente, lo que conlleva a que no haya una buena sincronización por parte de los semáforos, esto influye notablemente en la congestión de la zona de la fuente luminosa, ya que los semáforos no actúan de manera óptima para la circulación. Se obtuvo que el número máximo de tiempo en verde es el semáforo 1, denominado semáforo 1.1 con el tiempo de 65 segundos, y el tiempo de más bajos lo tiene

los semáforos 6 y 7, denominados semáforo 2.2 y semáforo 2.3 respectivamente con el tiempo de 20 segundos.

CAPÍTULO 5. SELECCIÓN DEL OPERADOR INTELIGENTE

Determinar las técnicas de inteligencia artificial para los semáforos inteligentes

Se entiende que la entrada de los datos de los semáforos inteligentes para que la maquina pueda procesar los datos obtenidos se deben analizar los datos con unas técnicas de inteligencia artificial, que se verán a continuación. según Leyton, Rodríguez & Correa (2014)

Aunque resulta innegable que la incorporación de nuevas tecnologías dentro del sector industrial es un aporte de gran importancia para el mejoramiento de procesos productivos y la toma de decisiones, es evidente que el principal constituyente de cualquier tipo de organización es el componente humano, el cual es, de hecho, el más susceptible de verse impactado por los cambios tecnológicos en su entorno laboral. (p. 213)

En el tema de los semáforos inteligentes tienen varias ramas en lo que tiene que ver con las técnicas de inteligencia artificial, según Ospino, Robles & Duran (2014) “(...) actualmente se están utilizando técnicas de inteligencia artificial, las cuales son capaces de encontrar correlaciones entre datos diversos y que han mostrado ser útiles en la determinación de parámetros (...)”(p. 226) se vive una amplia forma de semaforizaciones en las grandes ciudades, según Hurtado, Villarreal, Villarreal (2016) “Para sistemas donde es posible recolectar numerosos datos representativos de su operación, tanto en condiciones normales como anómalas, es posible construir un modelo mediante técnicas como redes neuronales o modelos difusos del tipo Takagi-Sugeno.”(p.20) cómo los semáforos que a cierta hora cambian su temporizador, para así intentar eliminar el caos en ciertas horas, y un sin fin de ejemplos, en lo que respecta a este proyecto. Guzmán, & Álvaro (2013) dicen

Independientemente de lo que resulte de los cálculos, la duración del ciclo tiene que estar forzosamente comprendida entre los límites que fija la psicología del conductor. La práctica indica que ciclos menores de 35 segundos o mayores de 120 se acomodan difícilmente a mentalidad del usuario de la vía pública. (p.19)

Se entiende que no pueden ser valores muy exagerados, ya que, los conductores, los transeúntes se mostrarían afectados por la demora o por el corto tiempo de la duración del semáforo, y sería un punto muy negativo en el sistema y en la ciudad, y más que una solución sería un caos vehicular.

Máquinas de soporte vectorial

Los semáforos inteligentes utilizando máquinas de soporte vectorial Molina, Bejarano, Robles & Dávila (2014) dicen

Las SVM, constituyen una de las principales herramientas del aprendizaje estadístico hoy en día. Están basadas en una proyección de los datos en una dimensión mayor, conocida como Dimensión de Vapnik-Chervonenkis, donde el problema a resolver tiene garantizada una solución y pueden ser implantadas usando tanto hardware como software. En el diseño de una SVM se plantea el uso de Hiperplanos como elementos discriminadores lineales para separar las clases dadas con el mayor margen posible lo cual hace que su algoritmo de aprendizaje sea más eficiente y presente mejores tiempos de respuesta a la hora de clasificar y cuantificar imágenes (p.30)

Se determina que las máquinas de soporte vectorial, son en teoría mejores en problemas de clasificación que las redes neuronales, se puede notar que en esta técnica también sería

una buena opción ya que puede manejar problemas de regresión que funcionan perfectamente con los semáforos inteligentes.

Lógica difusa

Se debe hablar de la lógica difusa para los sistemas de semáforos inteligentes es esencial, porque es la base de los tiempos de los cambios de semáforos, según Montenegro, Santos, Garmendia (2014) “Los problemas que pretenden reflejar un comportamiento medianamente realista suelen requerir un elevado número de reglas mientras que los problemas simples no siempre llegan a tener aplicaciones en el mundo cotidiano, etc.” (p.1) se refiere a la cantidad alta de reglas que se pueden utilizar en los semáforos para el cambio de luz, entonces según Mantilla y Márquez (2014) “Una vez que se definen las variables de entrada, salida y MF, tenemos que diseñar la base de reglas (o matriz de decisión de la base de conocimientos difusa)” (p.36) Las reglas definen el comportamiento del sistema de semaforización, como un ejemplo es que dos vías perpendiculares no pueden tener el color verde como activos. Ya que generaría conflicto, y podría generar accidentes.

Se entiende que este proyecto solo trabajará con la congestión vehicular, en el que se tomaría la congestión vehicular como la variable de entrada y las variables de salida sea el tiempo en verde, el tiempo de espera del verde será calculado, a través de la técnica de inteligencia artificial, y el color rojo, será la sumatoria de la duración del color verde de cada semáforo, es decir, que al principio de cada iteración, dando a entender que la iteración sea la circulación de todos los carriles, el rojo se debería calcular después de haber calculado el color verde.

Redes neuronales

También se ven los tipos de semáforos inteligentes utilizando redes neuronales, Pedraza, Hernández, & López (2012) dicen

Diseño de un modelo de tráfico vehicular, el cual examina el tráfico existente en una vía a través de una serie de semáforos. A partir de este modelo se sincronizan los tiempos de duración y de desfase de los semáforos, utilizando para ello el Sistema de Inferencia Difusa Basado en Redes Adaptativas (ANFIS). El modelo es simulado y los resultados se evalúan a nivel macroscópico con el modelo de tiempos fijos, que funciona actualmente en Bogotá-Colombia. (p.79)

Se ve reflejado un modelo de tráfico vehicular que puede funcionar en la ciudad de Bogotá, que es una ciudad mucho más grande que Cúcuta al ser el distrito capital de Colombia, se puede vivir en la ciudad de Bogotá una cantidad inmensa de automóviles en la ciudad, por ende, utilizan la medida del pico y placa, algo que en la ciudad de Cúcuta ya se está ejecutando, pero solo en la parte central de la ciudad de Cúcuta. Es un sistema mucho más complicado, al tratarse más de la técnica de las redes neuronales, que sería una buena técnica para los semáforos inteligentes.

Utilizando redes neuronales se debe tener una regulación del tiempo presentado en la luz roja, y en la luz verde, ya que si no se pone un margen, que significa el máximo expresado de espera para el cambio de luz, se verían afectados las personas de manera psicológica, porque se haría más tedioso, y más desesperante para los conductores que transiten la avenida, entonces las redes neuronales se deberían trabajar con la función lineal mixta, según Natareno (2013)“Función de activación lineal-mixta: con esta función, el estado de activación de la unidad está obligado a permanecer dentro de un intervalo de valores reales

prefijados” (p.26) se entiende que es la función más óptima para los semáforos en este caso, ya que al ser una de las avenidas más importantes de Cúcuta, la congestión sea mayor, por lo tanto los números de tiempo de espera serán altos, para satisfacer de manera correcta la circulación del tráfico, también se debe establecer un mínimo de tiempo, ya que , en las horas normales de la noche o madrugada, podrían, mostrar un cambio de luces , bastante alterado , ya que normalmente no circulan carros y sería un caos completo en cambios de luces en los semáforos.

Son una de las primeras técnicas de la inteligencia según Mercado, Pedraza y Martínez (2015)

Las Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. En los últimos años se ha masificado el uso de las RNA para la predicción de series temporales gracias a su capacidad generalizadora por el hecho de aprender a partir de ejemplos, tal y como lo hace el cerebro humano. (p.89)

Se puede identificar como maquinas inteligentes capaces de tomar decisiones en el menor tiempo posible para la gran congestión vehicular que hay en la ciudad de Cúcuta en ciertas horas del día. Porque es la que más beneficios da a la hora de infraestructura en la avenida diagonal Santander de la ciudad de Cúcuta, al ser óptimo para la avenida se refleja que podría ser un gran aporte en la ciudad, entonces se podría determinar la viabilidad de un sistema de semaforización con ese tipo de técnica de inteligencia artificial. Según Natareno (2013) “las redes neuronales son más que otra forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos” (p.8) una emulación o

simulación de las redes neuronales, se entiende para poder controlar como sería una persona para controlar los semáforos. Según Mercado, Pedraza y Martínez (2015)

Cuando se trabaja con series temporales, una de las tareas más importantes es la de predecir los datos futuros de la serie, es decir, a partir de los datos del pasado proyectar los valores que tomará la variable determinada. Para llevar a cabo esta actividad se construye un modelo matemático que capture, total o parcialmente las características de esta. (p.89)

Se concluye que para la mayor eficiencia se toma a partir de la predicción de los nuevos tiempos ya habiendo calculado correctamente los datos de prueba con los cuales la inteligencia artificial aprende.

Elección de un operador inteligente para la semaforización inteligente

Para poder elegir un operador inteligente se debe tener en cuenta los valores agregados que nos dan los operadores inteligentes, para lo cual se comparó cada técnica.

Tabla 9. Comparativa de operadores inteligentes

OPERADORES INTELIGENTES		
MAQUINAS DE SOPORTE VECTORIAL	LOGICA DIFUSA	REDES NEURONALES
SVM se plantea el uso de Hiperplanos como elementos discriminadores lineales para separar las clases dadas con el mayor margen posible lo cual hace que su algoritmo de aprendizaje sea más eficiente y presente mejores tiempos de respuesta a la hora de clasificar y cuantificar imágenes	Los problemas que pretenden reflejar un comportamiento medianamente realista suelen requerir un elevado número de reglas mientras que los problemas simples no siempre llegan a tener aplicaciones en el mundo cotidiano	Cuando se trabaja con series temporales, una de las tareas más importantes es la de predecir los datos futuros de la serie, es decir, a partir de los datos del pasado proyectar los valores que tomará la variable determinada.
Las SVM su propósito son resolver problemas de clasificación y de estimación, para lo que constituye en que es uno de los mejores operadores inteligentes para problemas de clasificación	Una vez que se definen las variables de entrada, salida y MF, tenemos que diseñar la base de reglas (o matriz de decisión de la base de conocimientos difusa)	las redes neuronales son más que otra forma de emular ciertas características propias de los humanos, como la capacidad de memorizar y de asociar hechos

Fuente. Propia

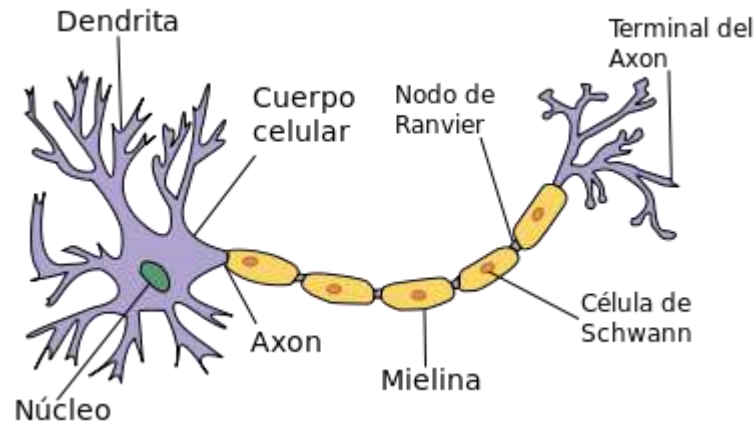
Este proyecto trabajará con la técnica de inteligencia artificial de redes neuronales, se escoge ya que es para la zona de la fuente luminosa una de los mejores operadores en materia de selección y estimación de los tiempos de cada semáforo.

Para el análisis de las redes neuronales artificiales como mejor técnica para el diseño de un sistema de semaforización, se debe escoger una función, en este caso una función que se acople mejor para un semáforo.

Para entender el cómo funcionan la técnica de inteligencia artificial de redes neuronales primero se debe comprender como funciona una de las células más importantes del ser humano, las neuronas. Las neuronas como parte del sistema nervioso de los seres humanos son la célula que produce reacciones eléctricas y químicas, son especializadas en recibir

estímulos e impulsos eléctricos. Las neuronas presentan características morfológicas de las cual se divide en:

Figura 1. Composición de una neurona



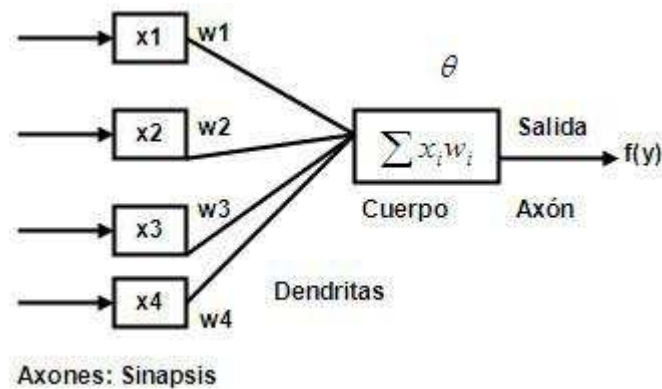
Fuente. <https://www.definicionabc.com/salud/neurona.php>

La cual la neurona funciona de la siguiente manera:

1. Donde las entradas de estímulos, es en sus dendritas.
2. En el cuerpo se realizan los estímulos correspondientes,
3. Después de ser procesado el estímulo, se es enviado al terminal del axón donde genera la salida de la estimulación resultante.

Para el estudio del proyecto también se tiene un modelo de una neurona artificial, la cual, trabaja de la misma manera que la neurona cerebral.

Figura 2. Neurona artificial



Fuente. <https://advancedtech.wordpress.com/page/6/>

La cual se tienen estímulos, transformados en entradas, se tienen los pesos para cada entrada, la cual en el cuerpo se es calculado mediante una sumatoria y en su mismo cuerpo realiza la respuesta que se es obtenido mediante la función de activación que se es escogida, la más adecuada para el proyecto. En el axón se es transmitida la respuesta obtenida En la zona de la fuente luminosa basado en Celis, Escobar, Sepúlveda, Castro, Medina y Ramírez (2016)

Por otro lado, la emulación y el modelado del comportamiento de los vehículos a través de una carretera ha servido de apoyo para plantear una solución a la problemática presentada en el flujo vehicular en una secuencia de semáforos, teniendo en cuenta los parámetros de velocidad y el ciclo de los semáforos con la finalidad de reducir el tiempo de viaje debido a la configuración de los mismos. (p.173)

Significa mucho que el sensor óptico a utilizar debe ser capaz de identificar autos en movimientos para así poder determinar cálculos y poder realizar los respectivos cambios de luces.

De acuerdo con Pedraza, Hernández & López (2012)

Al analizar los resultados se observa que el control de los semáforos con el modelo ANFIS es más óptimo, ya que la densidad vehicular se reduce, permitiendo atender una mayor cantidad de vehículos en una misma distancia al compararse con el sistema de tiempos fijos. A su vez el modelo ANFIS aumenta la velocidad promedio de los vehículos a través de la vía, lo cual aumenta la movilidad vehicular, y también se presenta una disminución del promedio de vehículos en cola sobre todas las colas, reduciendo la posibilidad de generar una congestión vehicular por causa de los semáforos. (p.86)

Demostrando, así como la mejor técnica para estos casos y para el de la fuente luminosa son las redes neuronales, lo cual realizan en mejor medida la optimización de tiempos y la optimización del flujo vehicular en esta zona, la fuente luminosa.

CAPITULO 6. SINCRONIZACION NUEVA PARA LOS SEMAFOROS

Etiquetado de vías, intersecciones y semáforos de la zona de la fuente luminosa

En la sincronización de los semáforos de la fuente luminosa es prácticamente independiente cada intersección de las otras, lo cual es causa de la congestión vehicular en horas pico, y aunque se supondría que en horas valle el flujo vehicular debería ser constante y rápido, en esta zona se torna pesado y lento, lo cual genera atascos a horas imprevistas o que menos se imaginan, basados en Valencia (2000)

La coordinación de semáforos consiste en la sincronización de los programas de tiempo de los semáforos con el propósito de favorecer la progresión del tránsito, es decir, que los grupos de vehículos (pelotones) avancen a lo largo de la vía o rutas (sucesión de vías) manteniendo una velocidad compatible con las características geométricas de la vía que recorre y el nivel de servicio del tránsito, experimentando el mínimo de demoras y detenciones. La coordinación exige que todas las intersecciones tengan la misma duración de ciclo. (p.90)

Se tiene en como consecuencia el uso dinámico de los tiempos de los semáforos de una manera no tan sencillo, puesto que no puede ser los semáforos tener un tiempo de libre albedrio, sino que deben trabajar todos en conjuntos si es que en verdad se quiere dar solución a los problemas de atascos en la zona.

Contrastando con lo escrito en la señalización capítulo 7mo de Los semáforos la finalidad de un sistema de semáforos sólo se cumple si es operado de una manera consistente y se acondiciona a las necesidades y requerimientos del tránsito. Los ciclos excesivamente largos

y la división impropia de los mismos ocasionan faltas de respeto y desobediencia a las indicaciones de los semáforos.

Imagen 1. Mapa de las vías de la zona de la fuente luminosa

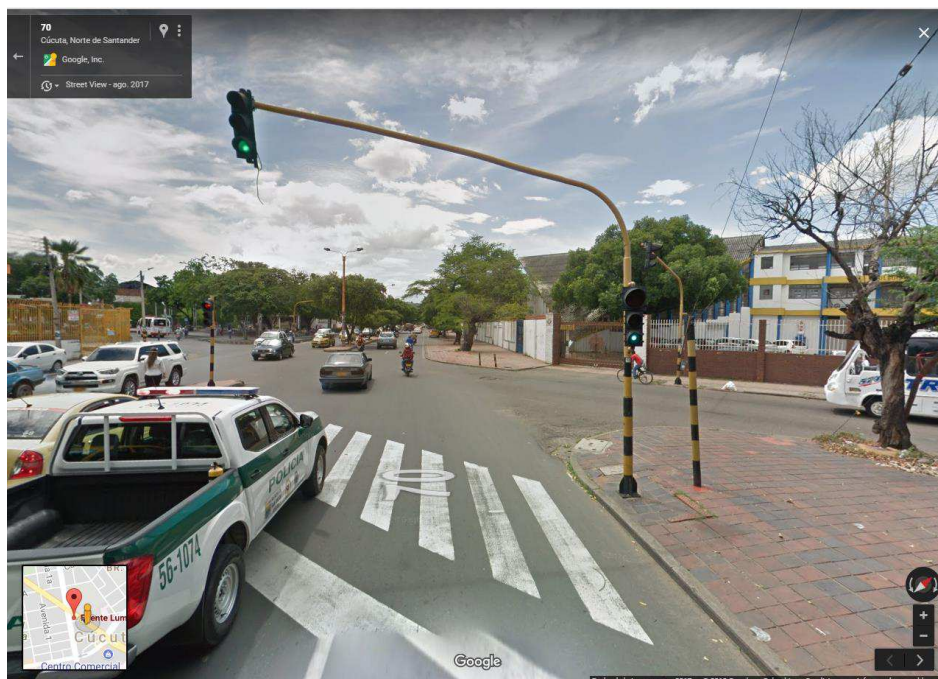


Fuente. Google Maps

Primero se deben etiquetar cada uno de los dispositivos de los semáforos para tener un orden, y así una manipulación de que dispositivos se tienen control sobre los semáforos, para el desarrollo del proyecto se debe tomar en cuenta que en la zona de la fuente luminosa hay 3 intersecciones en la cual todas tienen semáforos, de la cual las vamos a separar en tres sectores:

1. El primer sector será la intersección de la avenida diagonal Santander con la calle 8
 - 1) El primer semáforo de ese sector es el de la diagonal Santander en hacia el noroeste
 - 2) El segundo es el en la misma calle, pero es un cruce a la izquierda para entrar en la calle 8, donde siempre trabajara con el primer semáforo
 - 3) El tercero es el semáforo de la diagonal Santander hacia el sureste
 - 4) El ultimo es el de la calle 8
2. El segundo sector será la intersección de la avenida 0 con la calle 8
 - 1) El primer semáforo será el de la calle 8
 - 2) El segundo es el de la avenida 0 hacia el norte
 - 3) El tercero semáforo será el de la avenida 0 hacia el sur
3. El tercer sector será la intersección de la avenida diagonal Santander y la avenida 0
 - 1) El primer semáforo será el de la el de la avenida diagonal Santander hacia el noroeste
 - 2) El segundo será el de la avenida diagonal Santander hacia el sureste
 - 3) El tercer semáforo será el de la avenida 0 hacia el sur
 - 4) El cuarto semáforo será el de la avenida 0 hacia el norte

Imagen 2. Sector 1 avenida Diagonal Santander con calle 8



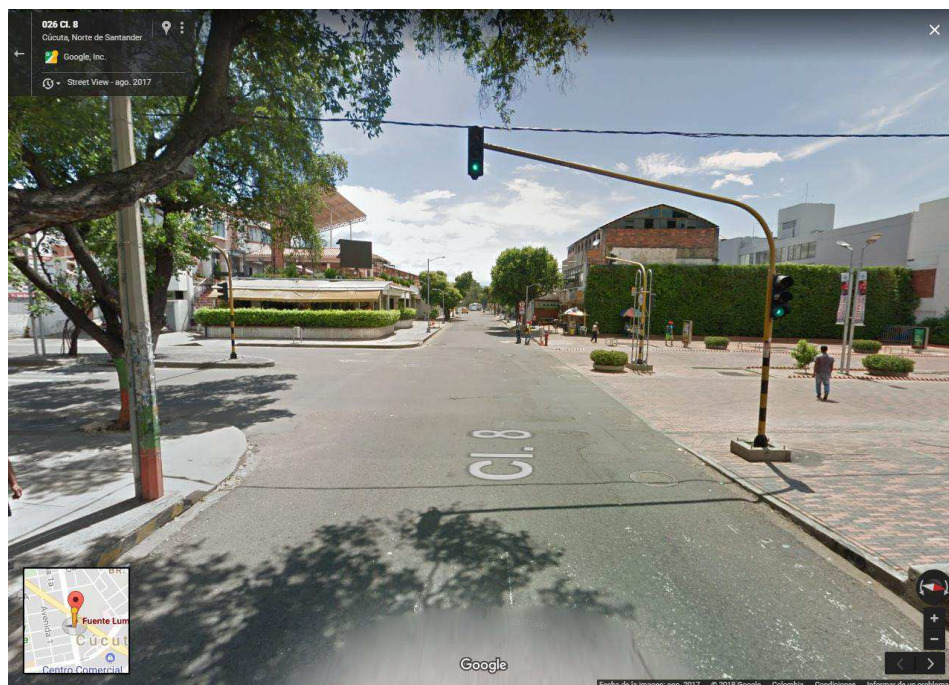
Fuente. Google Maps

Figura 3. Enumeración del sector 1



Fuente. Propia

Imagen 3. sector 2 avenida 0 con calle 8



Fuente: Google Maps.

Figura 4. Enumeración del sector 2



Fuente. Propia

Imagen 4. Sector 3 avenida diagonal Santander con avenida 0



Fuente. Google Maps

Figura 5. Enumeración sector 3



Fuente. Propia

Asignación de prioridades

Sabiendo esto, se quiere implementar una sincronización a los semáforos de ese sector, ya sea, para una mejor circulación en los ciertos casos en las calles que interceden en la fuente luminosa, vienen de otros semáforos que actúan independientes de ellos, y esto a su vez genera destiempo en la llegada de los automóviles a los semáforos de la zona de la fuente luminosa, ya que en ciertos casos en el semáforo puede durar hasta 20 segundos en verde sin ningún carro que pase por esa calle.

Para la sincronización el simulador en vez de trabajar un solo semáforo y este a su vez esté aislado, se tomará un cierto tipo de sincronización para los semáforos, y se asignara los turnos para que funcionen de manera ordenada.

Se escoge la sincronización de los semáforos de manera que sea la manera más fluida en teoría

De esta manera se tomará acción sobre los semáforos haciendo primero su cálculo en el en cada uno de los sectores asignados anteriormente y de ahí consultar con sus colores para así hacer un promedio de las secciones para que haya una sincronización en todo el sector, de esa manera no será independiente cada semáforo de sus vías aledañas, sino que, actúen según su entorno. se debe tener en cuenta el tiempo de cambio de los semáforos actualmente, que, en la recolección de datos se obtuvieron estos resultados en cuestión del dicho tiempo:

- sector 1 es de 100 segundos para dar un ciclo completo de semaforización
- sector 2 es de 50 segundos para dar un ciclo completo
- sector 3 es de 90 segundos para dar un ciclo completo

lo cual funcionan de manera cíclica, sin importar las adversidades de las calles, o sobre la congestión de cada una, lo cual influye en la congestión en las intersecciones, lo cual en este momento se deben tomar ciertas importancias en que vías son las más concurridas en el sector de la fuente luminosa, lo cual se puede tomar de tres métodos:

1. Determinando actualmente quien posee mayor tiempo en verde en sus semáforos.
Dando así el nivel de importancia en materia de tiempos.
2. Determinando según el flujo vehicular por el cual transitan en las diferentes vías de la zona de la fuente luminosa.
3. Determinando según la cantidad de automóviles que se encuentran esperando en cada semáforo

Primer método (véase en la tabla 8):

Sector 1

En el sector 1 lo cual muestra la clara vía principal que transita, es la Diagonal Santander y la vía secundaria es la calle 8.

Sector 2

En el sector 2 lo cual muestra la clara vía principal que transita, es la calle 8 y la vía secundaria es la avenida 0.

Sector 3

En el sector 3 lo cual muestra la clara vía principal que transita, es la Diagonal Santander y la vía secundaria es la avenida 0.

Segundo método (véase en las tablas 5, 6 y 7):

Sector 1

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 5):

Lo cual se ve como vía importante y más transitada la calle 8 y secundaria la Diagonal Santander.

Sector 2

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 6):

Lo cual se ve como vía importante y más transitada la calle 8 y secundaria la avenida 0.

Sector 3

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 7):

Lo cual se ve como vía importante y más transitada Diagonal Santander y secundaria la avenida 0.

Tercer método (véase en las tablas 2, 3 y 4)

Sector 1

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 2):

Lo cual se ve como vía importante y más transitada la Diagonal Santander y secundaria la calle 8.

Sector 2

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 3):

Lo cual se ve como vía importante y más transitada la avenida 0 y secundaria la calle 8.

Sector 3

Siguiendo los promedios de la cantidad de autos en horas valle se tiene (véase en la tabla 4):

Lo cual se ve como vía importante y más transitada la avenida 0 y secundaria la Diagonal Santander.

Cualesquiera de los tres métodos son válidos, y arrojan resultados similares lo cual muestra cual la principal vía para tomar como principales y secundarias, las tres arrojan resultados iguales, todos los sectores varían, pero no significativas para la sincronización, son un caso que se debe tomar con cuidado.

No hay ningún método ganador, ya que todos los métodos arrojan resultados diferentes, así que se realizará la sincronización con las vías según se cumpla una buena circulación.

Paso a paso de la nueva sincronización

Para dar un orden en la salida de los semáforos, se debe dar privilegios a una de las avenidas principales de Cúcuta, el cual se compondrá de 5 pasos secuenciales para ser completado:

Paso 1

La avenida Diagonal Santander donde se tiene el semáforo 1.1 y 1.3, la cual se activará al tiempo, para descongestionar rápidamente las vías en esa avenida, al tiempo se activa los semáforos 2.2 y 2.3 en el sector 2 de la fuente luminosa, y en el sector 3 se activarán los semáforos 3.1 y 3.2.

Figura 6. Primer paso de la sincronización



Fuente. Propia

Paso 2

Después en el sector 1 de la fuente luminosa seguirá encendido el semáforo 1.1 y se activará el semáforo 1.2, en el sector 2 se activará el semáforo 2.1 y en el sector 3 seguirá activado el semáforo 3.1.

Figura 7. Segundo paso de la sincronización

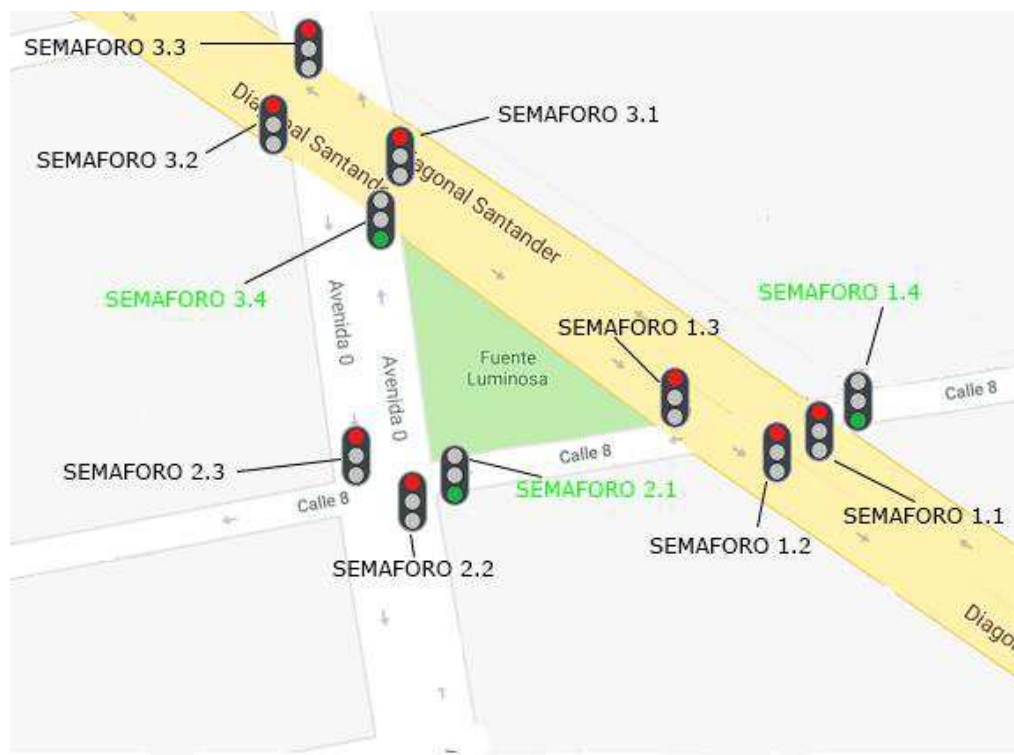


Fuente. Propia

Paso 3

Seguidamente en el sector 1 de la fuente luminosa se activará el semáforo 1.4, en el sector 2 seguirá encendido el semáforo 2.1 y en el sector 3 se activará el semáforo 3.4.

Figura 8. Tercer paso para la sincronización

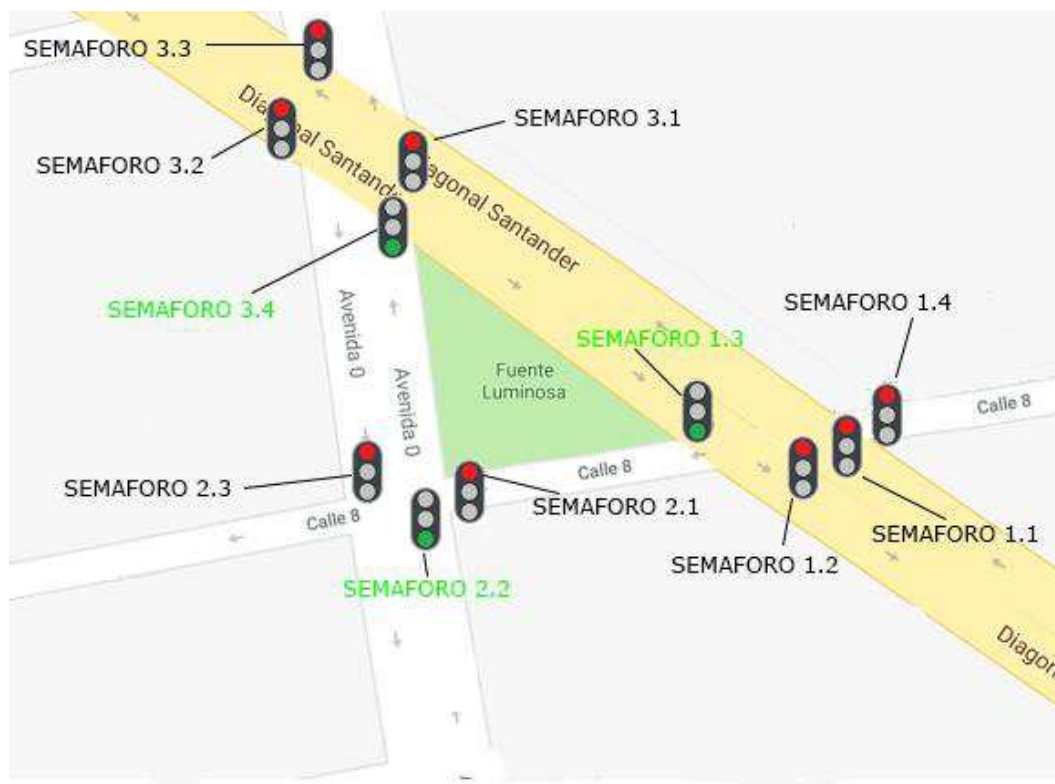


Fuente. Propia

Paso 4

En este paso en el sector 1 de la fuente luminosa se activará el semáforo 1.3, en el sector 2 se activará el semáforo 2.2 y en el sector 3 seguirá encendido el semáforo 3.4.

Figura 9. Cuarto paso para la sincronización



Fuente. Propia

Paso 5

Seguidamente en el sector 1 de la fuente luminosa se activarán los semáforos 1.1 y 1.2, en el sector 2 se activará el semáforo 2.3 y en el sector 3 se activará el semáforo 3.3.

Figura 10. Quinto paso para la sincronización



Fuente. Propia

Una vez terminado el ciclo se debe mantener los semáforos en los que se están seguidos para así completar el ciclo, y seguir con el paso 1

Esta es una de las maneras más eficientes que puede llegar a dar los semáforos para poder hacer el flujo óptimo de vehículos y evitar la congestión vehicular en esa zona, aparte de esa solución habría un problema en la sincronización de los otros semáforos porque puede que funcione correctamente esta sincronización, pero en los otros semáforos puede haber más congestión lo cual influye, un choque o un siniestro.

CAPITULO 7. DESARROLLO DEL SIMULADOR DE ESCRITORIO

Metodología de desarrollo

El desarrollo del software hará con metodología de desarrollo ágil según Mitre, Ortega & Lemus (2014)

Las metodologías ágiles toman su nombre después de que en 2001 un grupo de 17 expertos en desarrollo de software reunieron sus ideas y crearon el manifiesto ágil, estableciendo doce principios y en donde cada metodología debía cumplir con los siguientes preceptos: individuos e interacciones sobre procesos y herramientas, software funcionando sobre documentación extensiva, colaboración con el cliente sobre negociación contractual, respuesta ante el cambio sobre seguir un plan. (p. 404)

Siguiendo la metodología SCRUM según Sobrevilla, Hernández, Velasco, & Soriano (2017)

Scrum es uno de los métodos ágiles más populares en la actualidad y se ha convertido en el método ágil de preferencia de equipos de desarrollo de software superando a otros (...)

El proceso de Scrum está definido por una serie de iteraciones de duración fija llamadas sprints, que pueden durar desde una hasta cuatro semanas según las preferencias y/o necesidades del equipo. (p.2)

desarrollado en 8 Sprints que se realizó por cada semana, cada sprint se comprende tres fases; la primera que es la planificación donde se realizaran tareas como la elaboración de análisis, elaboración de propuestas, maquetación, etc. En la segunda fase se llevará a cabo la revisión, en esta fase se pone a prueba los modelos propuestos para su aceptación por parte

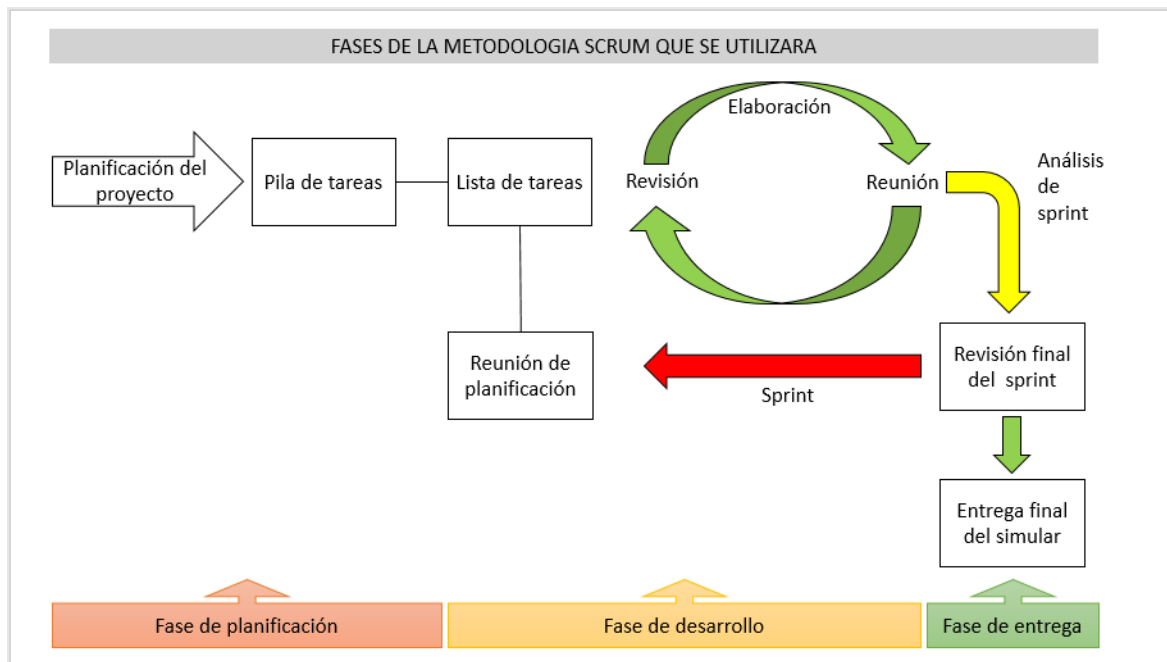
del director del proyecto. En la tercera fase tenemos la retrospectiva, seguidamente de la revisión en esta etapa lo que se hace es analizar la respuesta del director del proyecto y teniendo en cuenta se realiza ajustes al modelo que se presentaron en la segunda fase.

Según lo propuesto se toma en cuenta que los procesos de desarrollo con esta metodología de desarrollo ágil según Arana, Ruiz y La Serna (2015)

La metodología desarrollada para nuestro sistema será el Scrum que define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral, prototipos, etc.) mediante la aceptación de la naturaleza caótica del desarrollo de software. (p. 151)

Lo cual muestra la gran calidad de esta metodología en el momento de desarrollar un simulador, ya que un simulador necesita de un desarrollo estable, cíclico y retroalimentativo, para que haya un funcionamiento óptimo y una eficiencia bastante alta, lo que quiere decir que plantea una retroalimentación sobre la inteligencia artificial y el desarrollo de software. Se toma también para evitar conflictos entre los tiempos de entrega de resultados, saber que se está desarrollando, como lo está realizando y porqué se lo realiza de cierta manera. Cumpliendo con las dos ramas de la ingeniería de sistemas se puede realizar de la manera más óptima y correcta en el uso de la metodología, para lo cual se toma Scrum como metodología de desarrollo.

Figura 11. Fases de la metodología de desarrollo planteada para el proyecto



Fuente. Propia

Sprints

Sprint 1

Planificación

Se llevó a cabo todo lo escrito en la sincronización, la selección de la técnica, la elección del lenguaje de programación que es un lenguaje de alto nivel, llamado Python, lo cual se utilizó la versión 3.6.5 y el paradigma de programación que se va a utilizar es el de programación orientado a objetos (POO), según Challenger, Díaz, & Becerra (2014). “Python cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas.” (p.2), en el cual se

trabajaré de manera significativa, ya que es un lenguaje interpretado, que para las pruebas sería muy recomendable, según García, Rodríguez & Taboada (2017)

Estos métodos se destacan por su rapidez, que es uno de los parámetros que se tiene siempre en cuenta en las investigaciones. Se propone el desarrollo de aplicaciones con los distintos algoritmos implementados en lenguajes de programación tales como Python o Android, que son lenguajes ampliamente utilizados actualmente y permiten el uso de las mismas tanto en el computador como en dispositivos móviles. (p.119)

y en la edición o actualización de código en un futuro, se plantea que pueda reutilizarse el código, y ser utilizado como código libre para su uso.

Revisión

En la reunión se acordó concluir el análisis de que se plantea para el simulador, ya que no hay una base sólida para el desarrollo, y de cómo se va a realizar.

Retrospectiva

Se da la pauta para poder realizar un modelo de la ingeniería de requisitos, lo cual dará paso al desarrollo del simulador, el contexto donde se va a implementar, que se va a utilizar y en el próximo sprint revisar los requisitos, y maqueta de esos requisitos.

Sprint 2

Planificación

La ingeniería de requisitos es muy importante en este punto de planeación del simulador, lo cual infiere notoriamente lo que necesita el software, para que cumpla una calidad alta a

la hora de realizar un desarrollo correcto, realizando la elicitación del proyecto según Buitrón & Pino (2015)

La elicitación de requisitos (ER) implica un proceso de captura y entendimiento de las necesidades de los usuarios que serán usadas en las etapas siguientes del desarrollo del producto software. Dentro de las necesidades se deben incluir tanto las necesidades funcionales (RF) como las restricciones (conocidas como requisitos no funcionales, RNF), para así determinar la calidad del producto. (p.130)

En la planeación se debe tener en cuenta los requisitos necesarios para el funcionamiento del simulador, y también determinar unos requisitos que no sean parte del proyecto, sino que son externos

Requisitos funcionales:

RF1: Se desea que se simule todas las intersecciones a la vez para su verificar su tiempo de respuesta.

RF2: Se debe ejecutar de manera cíclica la simulación.

RF3: Se debe tener el resultado del tiempo en verde en cada paso de la sincronización.

RF4: Se debe simular paso a paso como está estipulado en la sincronización previamente hecha, la cual se deben tomar la cantidad de automóviles en cada momento del paso a paso.

RF5: El usuario podrá salirse de la simulación cuando él lo desee, lo cual lo re direccionará a la pantalla de inicio.

Requisitos no funcionales:

RFN1: El sistema no debe demorar más de 5 segundos en resolver los cálculos de las redes neuronales en cualquiera de los resultados.

Después de analizar todos los requisitos después de la reunión el día 27 de febrero del 2018, en la cual se entiende del sistema para la cual se deben representar en los prototipos que se muestra a continuación, En los cuales son muy importantes a la hora de simular el tiempo de respuesta del programa para su implementación, y su viabilidad, de los cuales los requisitos son esenciales para el diseño del simulador, los cuales podemos tomar los requisitos y desarrollar su primer prototipo:

El inicio será muy sencillo ya que no requiere de más pasos en la implementación:

Figura 12. Prototipo del inicio del simulador



Fuente. Propia

En el siguiente prototipo se tiene diez figuras de las cuales donde se especifican la inserción de la cantidad de automóviles en cada semáforo, según el paso de la sincronización que debe realizar.

Figura 13. Prototipo simulacion del primer paso

SIMULACION DEL PRIMER PASO

Ahora escoge la cantidad de carros que hay en cada semaforo:

semaforo 1.1	
semaforo 1.3	
semaforo 2.2	
semaforo 2.3	
semaforo 3.1	
semaforo 3.2	



El mapa muestra un sector urbano con varias calles y puntos de interés. Se han marcado los semáforos 1.1, 1.3, 2.2, 2.3, 3.1 y 3.2. El sector escogido está resaltado en amarillo. Los puntos de interés incluyen 'Fuente Luminosa', 'Almacén Vidrios Casadiego' y 'Calle 6'.

Fuente. Propia

Figura 14. Prototipo resultados del primer paso

RESULTADOS DEL PRIMER PASO

Este es el resultado de la simulacion del primer paso, el tiempo que se le asignará a todos los semaforos

El tiempo calculado es de:

XX SEGUNDOS



El mapa muestra el mismo sector urbano que en la Figura 13, con las calles y puntos de interés marcados. El sector escogido está resaltado en amarillo.

Fuente. Propia

Figura 15. Prototipo simulación del segundo paso

SIMULACION DEL SEGUNDO PASO

Ahora escoge la cantidad de carros que hay en cada semaforo:

semaforo 1.2

semaforo 2.1

semaforo 3.1



El mapa muestra un sector urbano con varias calles y puntos de interés. Se han marcado tres semáforos: SEMAFORO 1.2, SEMAFORO 2.1 y SEMAFORO 3.1. El área seleccionada para la simulación está sombreada en verde y etiquetada como 'mapa del sector escogido'. Los puntos de interés incluyen Fuente Luminosa, Calle 8, Almacén Vidrios Casadiego, Hotel Suite, MacPolo Avenida Caro, y Hotel Zuri 1.

Fuente. Propia

Figura 16. Prototipo resultados del segundo paso

RESULTADOS DEL SEGUNDO PASO

Este es el resultado de la simulacion del segundo paso, el tiempo que se le asignará a todos los semaforos

El tiempo calculado es de:

XX SEGUNDOS



El mapa es idéntico al de la Figura 15, mostrando el sector urbano con las calles y puntos de interés, y los tres semáforos marcados. El área seleccionada para la simulación sigue sombreada en verde.

Fuente. Propia

Figura 17. Prototipo simulación del tercer paso

SIMULACION DEL TERCER PASO

Ahora escoge la cantidad de carros que hay en cada semaforo:

semaforo 1.4

semaforo 2.1

semaforo 3.4



El mapa muestra un sector urbano con varias calles y puntos de interés. Se han marcado tres semáforos: SEMAFORO 1.4, SEMAFORO 2.1 y SEMAFORO 3.4. El mapa incluye etiquetas como 'Fuente Luminosa', 'Almacén Vidrios Casadiego', 'MacPolo Avenida Cero', 'Avenida B', 'Calle 8', 'Calle 9', 'Calle 10', 'Calle 11', 'Calle 12', 'Calle 13', 'Calle 14', 'Calle 15', 'Calle 16', 'Calle 17', 'Calle 18', 'Calle 19', 'Calle 20', 'Calle 21', 'Calle 22', 'Calle 23', 'Calle 24', 'Calle 25', 'Calle 26', 'Calle 27', 'Calle 28', 'Calle 29', 'Calle 30', 'Calle 31', 'Calle 32', 'Calle 33', 'Calle 34', 'Calle 35', 'Calle 36', 'Calle 37', 'Calle 38', 'Calle 39', 'Calle 40', 'Calle 41', 'Calle 42', 'Calle 43', 'Calle 44', 'Calle 45', 'Calle 46', 'Calle 47', 'Calle 48', 'Calle 49', 'Calle 50', 'Calle 51', 'Calle 52', 'Calle 53', 'Calle 54', 'Calle 55', 'Calle 56', 'Calle 57', 'Calle 58', 'Calle 59', 'Calle 60', 'Calle 61', 'Calle 62', 'Calle 63', 'Calle 64', 'Calle 65', 'Calle 66', 'Calle 67', 'Calle 68', 'Calle 69', 'Calle 70', 'Calle 71', 'Calle 72', 'Calle 73', 'Calle 74', 'Calle 75', 'Calle 76', 'Calle 77', 'Calle 78', 'Calle 79', 'Calle 80', 'Calle 81', 'Calle 82', 'Calle 83', 'Calle 84', 'Calle 85', 'Calle 86', 'Calle 87', 'Calle 88', 'Calle 89', 'Calle 90', 'Calle 91', 'Calle 92', 'Calle 93', 'Calle 94', 'Calle 95', 'Calle 96', 'Calle 97', 'Calle 98', 'Calle 99', 'Calle 100'.

Fuente. Propia

Figura 18. Prototipo resultados del tercer paso

RESULTADOS DEL TERCER PASO

Este es el resultado de la simulacion del tercer paso, el tiempo que se le asignará a todos los semaforos

El tiempo calculado es de:

XX SEGUNDOS



El mapa muestra el mismo sector urbano que en la Figura 17, con los mismos semáforos y puntos de interés. El mapa incluye etiquetas como 'Fuente Luminosa', 'Almacén Vidrios Casadiego', 'MacPolo Avenida Cero', 'Avenida B', 'Calle 8', 'Calle 9', 'Calle 10', 'Calle 11', 'Calle 12', 'Calle 13', 'Calle 14', 'Calle 15', 'Calle 16', 'Calle 17', 'Calle 18', 'Calle 19', 'Calle 20', 'Calle 21', 'Calle 22', 'Calle 23', 'Calle 24', 'Calle 25', 'Calle 26', 'Calle 27', 'Calle 28', 'Calle 29', 'Calle 30', 'Calle 31', 'Calle 32', 'Calle 33', 'Calle 34', 'Calle 35', 'Calle 36', 'Calle 37', 'Calle 38', 'Calle 39', 'Calle 40', 'Calle 41', 'Calle 42', 'Calle 43', 'Calle 44', 'Calle 45', 'Calle 46', 'Calle 47', 'Calle 48', 'Calle 49', 'Calle 50', 'Calle 51', 'Calle 52', 'Calle 53', 'Calle 54', 'Calle 55', 'Calle 56', 'Calle 57', 'Calle 58', 'Calle 59', 'Calle 60', 'Calle 61', 'Calle 62', 'Calle 63', 'Calle 64', 'Calle 65', 'Calle 66', 'Calle 67', 'Calle 68', 'Calle 69', 'Calle 70', 'Calle 71', 'Calle 72', 'Calle 73', 'Calle 74', 'Calle 75', 'Calle 76', 'Calle 77', 'Calle 78', 'Calle 79', 'Calle 80', 'Calle 81', 'Calle 82', 'Calle 83', 'Calle 84', 'Calle 85', 'Calle 86', 'Calle 87', 'Calle 88', 'Calle 89', 'Calle 90', 'Calle 91', 'Calle 92', 'Calle 93', 'Calle 94', 'Calle 95', 'Calle 96', 'Calle 97', 'Calle 98', 'Calle 99', 'Calle 100'.

Fuente. Propia

Figura 19. Prototipo simulación del cuarto paso

SIMULACION DEL CUARTO PASO

Ahora escoge la cantidad de carros que hay en cada semaforo:

semaforo 1.3

semaforo 2.2

semaforo 3.4



Mapa del sector escogido

El mapa muestra un sector urbano con varias calles y puntos de interés. Se destacan tres semáforos: SEMAFORO 1.3 (verde), SEMAFORO 2.2 (rojo) y SEMAFORO 3.4 (verde). Las calles visibles son Calle 8, Calle 9 y Calle 10. Los puntos de interés incluyen Fuente Luminosa, Almacén Vidrios Castañedo, Belleza Impulso 1, y Cucuta.

Fuente. Propia

Figura 20. Prototipo resultados del cuarto paso

RESULTADOS DEL CUARTO PASO

Este es el resultado de la simulacion del cuarto paso, el tiempo que se le asignará a todos los semaforos

El tiempo calculado es de:

XX SEGUNDOS



Mapa del sector escogido

El mapa muestra el mismo sector urbano que en la Figura 19, con las mismas calles y puntos de interés. Los semáforos SEMAFORO 1.3, SEMAFORO 2.2 y SEMAFORO 3.4 están nuevamente marcados.

Fuente. Propia

Figura 21. Prototipo simulación del quinto paso

SIMULACION DEL QUINTO PASO

Ahora escoge la cantidad de carros que hay en cada semaforo:

semaforo 1.1

semaforo 1.2

semaforo 2.3

semaforo 3.3

SIMULAR **SALIR DE LA SIMULACION**

Fuente. Propia

Figura 22. Prototipo resultados del quinto paso

RESULTADOS DEL QUINTO PASO

Este es el resultado de la simulacion del quinto paso, el tiempo que se le asignará a todos los semaforos

El tiempo calculado es de:

XX SEGUNDOS

SEGUIR AL PRIMER PASO **REPETIR EL QUINTO PASO** **SALIR DE LA SIMULACION**

Fuente. Propia

Al completar el quinto paso, el programa sigue cíclicamente los pasos, pasando del quinto paso al primer paso, Los cuales completan los requisitos previamente estipulados.

Revisión

Se revisaron los requisitos cumplen con lo esperado a lo que se busca en el simulador, la maquetas, no es cómoda y no cumplen con los requisitos para el desarrollo del simulador, pero no es visualmente agradable, es ambiguo y no se entiende los pasos.

Retrospectiva

Se desea detallar una sincronización sobre los semáforos pendiente del análisis de esos casos para su respectiva revisión. También observar las horas en las que se implementaría, en especial los casos de la noche-madrugada, y en la tarde, donde, aunque no sea hora pico, sigue habiendo una gran cantidad de vehículos en esas vías.

Se deben reorganizar el maqueto, siguiendo como tal la simulación de los semáforos.

Se debe escoger los tipos de redes neuronales para el simulador en cada paso y como se obtendrán los datos de las redes neuronales, según el tipo de red que sea.

Sprint 3

Planeación

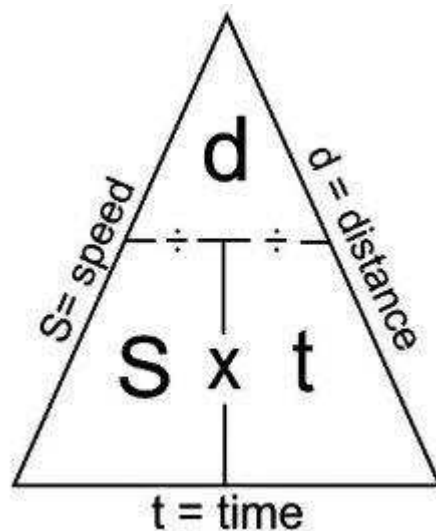
En esta parte del proceso se debe tomar una decisión de cómo va a ser el tipo de redes neuronales la cual van a ser redes perceptrón multicapa según Ceballos, Muñoz & Moreno (2011)

Esta problemática se puede resolver aplicando un modelo de red neuronal del tipo perceptrón multicapa que requiere realizar el proceso de selección del modelo, el cual abarca tres dificultades fundamentales: la selección de las entradas relevantes, la

determinación del número de capas ocultas y la determinación del número de neuronas en cada capa oculta, y finalmente, la estimación de los pesos de las conexiones. (p.110)

Las entradas relevantes van a ser la cantidad de automóviles que se encuentran en cada uno de los semáforos y como segunda entrada relevante son la hora y el minuto en la que se toma la cantidad de los semáforos, no va existir una capa oculta, sino que habrá dos capas, la de entrada y la de salida de la red, la estimación de los pesos será calculado iterativo a través del cual se va a implementar en este proyecto, debe determinar cómo se tomarán los datos de una matriz de aprendizaje, que debe tener para poder realizarse en todas las neuronas, se debe tener unos valores predeterminados los cuales se toman a partir del cálculo por velocidad de un carro utilizando la fórmula de la distancia:

Figura 23. triangulo de distancia, velocidad y tiempo



Fuente. <http://www.skateboardhere.com/skateboarding-physics.html>

De la cual se tomará la fórmula de la distancia, que, aun sabiendo que los automóviles están en estado de reposo en el semáforo según Pérez & Arco (2016)

Muchos de los métodos de aprendizaje automático dependen del cálculo de distancias en un espacio multidimensional para estimar la similitud entre dos ejemplos teniendo en cuenta la estructura de los datos. Está comprobado que se obtienen mejores resultados cuando la métrica se diseña específicamente para un contexto dado, pero esta es una tarea compleja. (p.43)

Se tomará esta fórmula, pero esto posee un problema, ya que los automóviles están en estado de reposo, y no en una velocidad de constante, lo cual influye en el desarrollo de esta fórmula, pero con base ella se determina que todos los datos son de incertidumbre, ya que no se puede obtener la distancia real que ocupan los automóviles estacionados, pero estimando que los automóviles su tamaño es de casi cinco metros de longitud, se toma que por respeto a los otros conductores se deja un espacio entre automóvil de 1 metro por delante y por atrás y siguiendo así la cadena, se obtiene que el espacio por automóvil es de 7 metros, en promedio, utilizando todas estas características, no se tienen en cuenta los buses, transporte público, entonces se podría dar como agregado al tiempo, dar más segundos para este medio de transporte, que es a su vez mucho más grande que el de un automóvil común de la ciudad, se le agregará al tiempo que se calculé a través de la formula (véase en la figura 23) que será 20 segundos, de los cuales se utiliza este tiempo para poder estimar el tiempo cuando un autobús, que ese tiempo es muy relativo, o cualquier otro vehículo que se dispone a parar y dejar un pasajero.

Se debe tener en cuenta que todos los semáforos excepto el semáforo 1.2 poseen 2 carriles, y el semáforo 1.2 posee un solo carril, lo cual influye directamente proporcional al cálculo del tiempo, ya que para todos los semáforos excepto el semáforo 1.2 se toma la cantidad de automóviles y se divide entre 2 (el número de carriles que posea el semáforo) lo cual el

semáforo 1.2 la cantidad de automóviles no se dividirá por la misma razón de que se dividirá entre 1, dando así un mismo resultado.

Un ejemplo de esto puede ser que un auto móvil va a una velocidad de 20 km por hora, expresado en m/s irá a 5.55556 metros por segundos y recorrerá 50 metros en 8.999 segundos, redondeándolos a 9 segundos, si otro automóvil va a una velocidad de 30 km por hora expresado en m/s 8.33333 metros por segundos y recorrerá los mismos 50 metros en 6 segundos exactos. Claramente no es el mismo tiempo, entonces no se puede valorar correctamente al auto más veloz, porque al momento de pasar una situación de estas, al llegar a los 50 metros de la calle el auto con menor velocidad, no alcanzaría a pasar por esa diferencia de segundos.

También se utilizará ese triángulo para el cálculo de los resultados para el aprendizaje de las redes neuronales porque en el desarrollo no debe haber un campo donde diga cuál es el tiempo que se debe colocar en ese caso específico, sino que el mismo programa lo calculará a través de esta fórmula.

Para la capa de salida la red debe tomar como entradas los valores que fueron obtenidos por la capa de entrada y su resultado debe ser un promedio de todos los semáforos en la red neuronal.

Para el maqueteo se re-direccionó el desarrollo de la interface gráfica para que sea más entendible y más gráfica, la parte de los semáforos, en la inserción de datos, en el tiempo y en el conteo. Para lo cual se actualizaron las maquetas:

Figura 24. Prototipo definitivo del simulador



Fuente. Propia

Se modificó en la parte en la que se visualiza el mapa con los semáforos, lo cual se muestra en cada paso los semáforos en verde, y se explica al usuario en la parte superior como son los estados de los semáforos, lo cual verde significa que deben avanzar, el semáforo en rojo significa que debe esperar, el semáforo en amarillo significa que ese semáforo estaba en verde y ahora cambiará a rojo, el semáforo que tiene el color rojo y amarillo encendido, significa que el semáforo estaba en rojo y ahora cambiara a verde.

En la parte del inicio, cambiará según en qué paso está, debajo de ella se colocarán los semáforos en los cuales se realiza el cálculo de las neuronas, la cantidad de autos que obtuvieron, lo cual ya no será digitado por el usuario, sino que se tomara aleatoriamente, ya que en los casos reales no se tiene un control sobre cuantos carros deben estar en cada

semáforo esperando, los semáforos se actualizarán a medida de cual paso se desarrolla en el simulador.

Siguiendo el orden descendente, hay un espacio donde muestra el tiempo que se ha obtenido según la red neuronal, este tiempo es estático, es decir que no cambiará en la simulación, sirve para que el usuario vea el tiempo que se obtuvo, debajo de este hay otro tiempo el cual es un contador hacia abajo lo cual significa que este tiempo se tomará primero por el número que se tiene en el tiempo estático, y va descendiendo su número hasta llegar a cero el cual no mostrará el cero, sino que mostrará una etiqueta que diga “en amarillo” para identificar que el semáforo cambia de paso y primero debe cambiar a amarillo y cambiará la imagen mostrando los semáforos que pasan a amarillo y amarillo rojo.

El botón de “¡simular!” funciona como control de los semáforos, lo cual actúa, cambiando la imagen de muestra del simulador, el paso, los semáforos calculados y los tiempos, el botón se deshabilitará después de ser presionado, y se volverá a activar después de que el contador del tiempo llegue a cero o a “en amarillo”, los semáforos que se encuentren en verde su etiqueta de nombre cambiará a verde.

Revisión

Se revisó satisfactoriamente el procedo de cómo se va a tomar la información y de que se va a realizar en cada caso para cada semáforo, el nuevo maqueteo cumple con los requisitos para el buen funcionamiento del simulador.

Retrospectiva

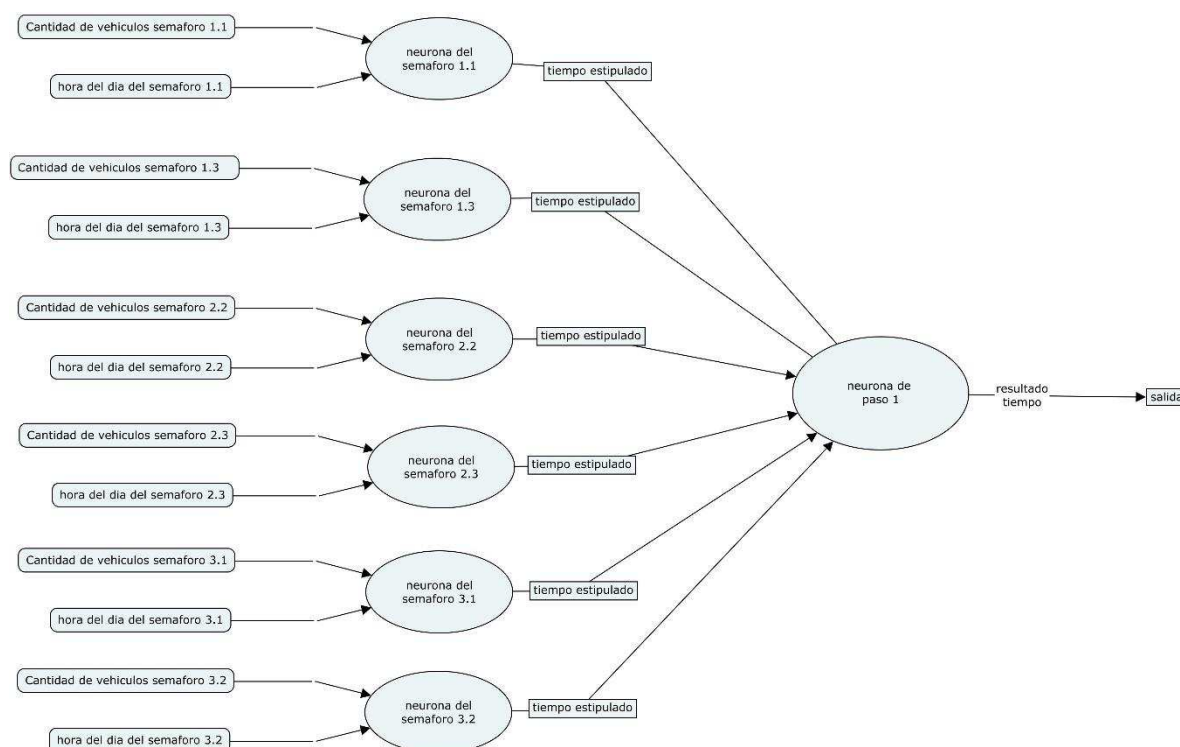
Se pide que se el modelo de red neuronal para su evalúo y aprobación con este tipo de entrada de datos sobre cada semáforo y el arreglo de dichas estructuras, para el desarrollo del aplicativo.

Sprint 4

Planeación

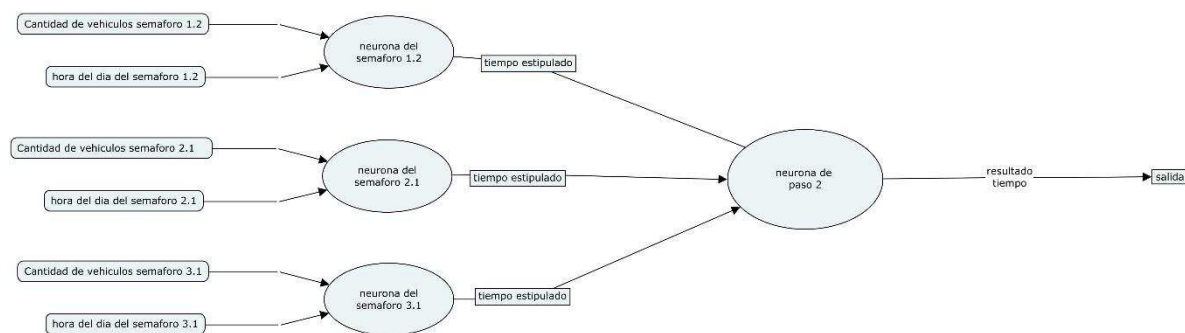
En este momento se plantearon los modelos de redes neuronales para la implementación del sistema de semaforización inteligente para la zona de la fuente luminosa, basados en la sincronización, concordando con Curvelo, de Araújo, Biazus & de Souza (2015) “(...) se propone utilizar una Red Neuronal Artificial (RNA) Perceptrón Multicapa (PMC) para simular la variación de la concentración de proteína de acuerdo con el tiempo y también para determinar la hora final del procedimiento (...)” (p.269) cada paso de la sincronización es un modelo de redes neuronales los cuales ya tenemos las entradas que se aceptaron (cantidad de automóviles, hora del día), para lo cual se diseñó los cinco modelos de las redes neuronales.

Figura 25. Modelo de redes neuronales del primer paso



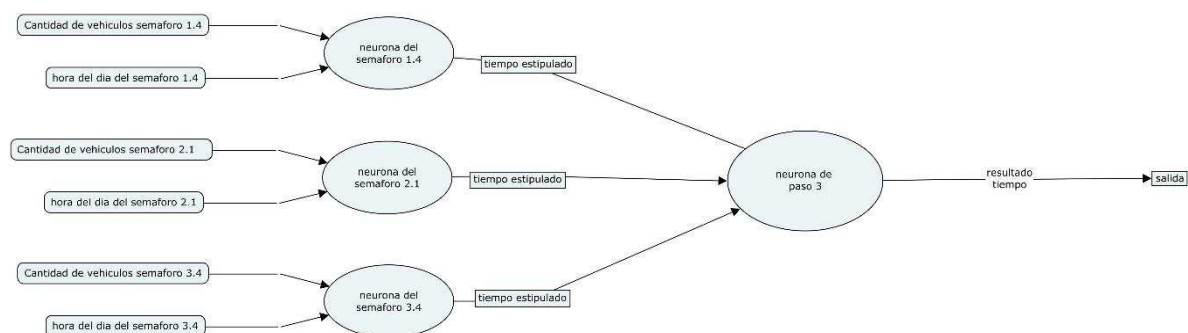
Fuente. Propia.

Figura 26. Modelo de redes neuronales del segundo paso



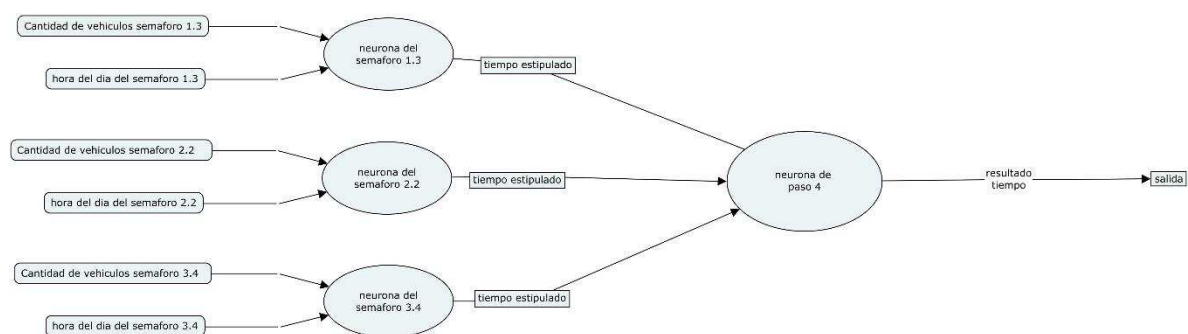
Fuente. Propia

Figura 27. Modelo de redes neuronales del tercer paso



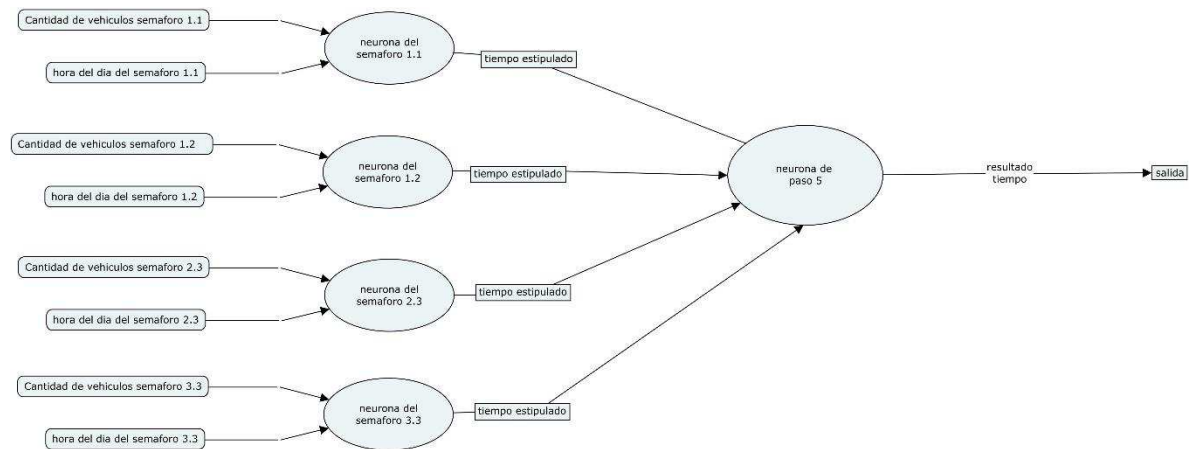
Fuente. Propia

Figura 28. Modelo de redes neuronales del cuarto paso



Fuente. Propia

Figura 29. Modelo de redes neuronales del quinto paso



Fuente. Propia

los cuales se empezarán a entrenar individualmente para después unir todas las neuronas según su modelo. Los cuales cada semáforo poseerá su neurona, la cual puede ser utilizado por los paso a paso de la sincronización.

Revisión

Se revisó satisfactoriamente los modelos de redes neuronales para cada paso de la sincronización del sistema de semaforización, el cual se trabajará satisfactoriamente en la programación según los modelos ya estipulados,

Retrospectiva

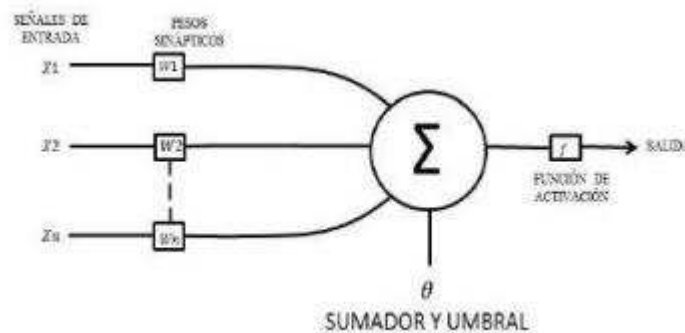
Se debe realizar el código para el aprendizaje de los semáforos, lo cual se debe aprobar, revisar y si se deben hacer correcciones al código, también se debe realizar el código para el aprendizaje de la neurona de salida que está planteado en los modelos.

Sprint 5

Planeación

Para plantear el código del aprendizaje de todas las neuronas, se debe primero saber que se requiere para el desarrollo de una neurona.

Figura 30. Elementos naturales de una neurona



Fuente. Vicario, Oubram, Bassam, Velázquez, & Ordóñez (2017)

Lo cual se necesitan: entradas, pesos sinápticos, la sumatoria, la función de activación, lo cual esta a su vez genera una salida. Se tiene ya conocimiento de las entradas, los pesos sinápticos que se utilizarán de manera que se ajustan según la necesidad del tiempo en verde que se necesita, en la sumatoria se suman las entradas por el peso (que al inicio se asigna aleatoriamente) y una vez sumado el cual, será el valor que entre en la función de activación, según Vicario, Oubram, Bassam, Velázquez, & Ordóñez (2017) “Función de activación o de transferencia: (...) en la activación de la unida se denomina función de activación. (...) Algunas de las funciones más utilizadas son la lineal, sigmoidea (...)” (p.43), el programa de aprendizaje de las redes neuronales utilizará la función de activación sigmoidea para la

neurona de la capa de entrada de cada red neuronal y la función de activación lineal para la capa de salida de cada red neuronal.

Figura 31. Función de activación sigmoidea

$$f(x) = \frac{1}{1 + e^{-x}}$$

Fuente Vicario, Oubram, Bassam, Velázquez, & Ordóñez (2017)

Figura 32. Función de activación lineal

$$f(x) = x$$

Fuente Vicario, Oubram, Bassam, Velázquez, & Ordóñez (2017)

Lo cual se tienen todo lo que integra el aprendizaje de la neurona, pero falta algo importante, lo cual determina que la neurona pueda aprender es sobre sus entradas, como se sabe, cuando entra la cantidad de automóviles entra un valor que no está entre cero y uno, lo cual se deben normalizar los valores de las entradas, lo cual se utiliza la normalización basada en la unidad.

Figura 33. Normalización por rangos

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Fuente. Propia

Teniendo en cuenta la normalización para los datos de entrada, como la cantidad de automóviles, la hora del día y el tiempo en segundos en verde(salida) dando este último con el despeje de X, ya que cada red neuronal da un valor normalizado, es decir que está entre cero y uno, lo cual para su despeje se daría de la siguiente forma:

Figura 34. Despeje de X en la normalización

$$X = X' (X_{\max} - X_{\min}) + X_{\min}$$

Fuente. Propia

Tomando la base de las fórmulas para la realización del aprendizaje de las neuronas, (véase la figura 33 y 34) se debe tomar en cuenta los valores de “Xmax” y “Xmin”, una vez teniendo en cuenta que “X” es el valor que deseamos normalizar y “X'” es el valor ya normalizado, se tienen que normalizar los valores de las entradas, y de la salida de la neurona.

Tabla 10. Normalización de las entradas y salidas

Variables de entrada y salida	Valor mínimo	Valor máximo
Cantidad de automóviles	0 autos	30 autos
Tiempo	0 horas	2359 horas
Resultado tiempo en verde	20 segundos	65 segundos

Fuente. Propia

En los valores (véase la tabla10) se definen el número mínimo y el número máximo de las entradas (cantidad de automóviles y tiempo) y el valor mínimo y máximo de la salida (resultado tiempo en verde), el cual se determinaron estos tiempos porque en la recolección de información se obtuvieron los datos mencionados anteriormente, para la cantidad de

automóviles se toma el valor mínimo de 0, ya que (véase en la tabla 3) es el valor en espera del semáforo 2.2 es igual a 0 en una de sus iteraciones, y el valor máximo es 30, ya que (véase la tabla 4) es el valor más alto registrado en los datos de dicha tabla y de todas en horas pico. Para el tiempo fue más sencillo, ya que las horas comienzan en 00:00 hasta las 23:59, y después de esta última hora se pasa al siguiente día. Para el resultado tiempo en verde se determinó el valor mínimo de 20 segundos y un máximo valor de 65 segundos, ya que (véase en la tabla 8) los semáforos 2.2 y semáforo 2.3 poseen la menor cantidad en segundos del tiempo en verde, y el semáforo 1.1 su tiempo es de 65 segundos en verde tomando estos valores como mínimo y máximo para la normalización y cálculo sobre las neuronas de la capa de entrada.

Teniendo en cuenta, los modelos de redes neuronales, no normalizará su valor del resultado en las capas de entrada, sino que entrará a la capa de salida y el resultado de la capa de salida se le aplica la normalización para que puede legible por los usuarios y ser tomado como valor obtenido. Las neuronas aprenderán con los valores de entrada y salida normalizados.

Una vez realizado el modelo de las redes neuronales y todas las características propias de una neurona perceptrón multicapa, se plantea el código de aprendizaje de las neuronas (véase en los anexos 1,2y3).

Teniendo en cuenta que para el cálculo del tiempo en las neuronas de entrada se utilizan las horas como guía.

Tabla 11. Máxima cantidad de automóviles en las horas del día

Horas preestablecidas	Intervalos	Máximo de automóviles
Horas pico	Desde las 10:30 hasta las 14:29 y desde 17:31 hasta 20:00	30
Horas congestionadas no pico	Desde 14:30 hasta 17:30	23
Horas de la madrugada	Desde 22:00 hasta 05:00	5
Horas valle	Desde 05:01 hasta 10:29 y desde 20:01 hasta 21:59	16

Fuente. Propia

Obteniendo estos datos (véase en la tabla 11) según los datos previamente dichos sobre la normalización acerca de la cantidad de automóviles, se destacan las horas valle donde (véase en la tabla 7) se obtuvo en el semáforo 3.2 tiene la cantidad más alta de automóviles en hora valle siendo esta 16, lo cual se toma como referencia. En horas de la tarde que no es hora pico, pero si ve una gran congestión se escogió el promedio entre la hora pico más alta y la hora valle más alta en materia de cantidad de automóviles. En horas de la madrugada se selecciona 5, ya que es un valor neutro a esas horas, es difícil expresar a cantidad de vehículos a esta hora donde es tan variable e inconstante.

Revisión

Se ha determinado como un código válido para el aprendizaje, el cual se debe utilizar de manera que se puedan obtener los valores nuevos, como los pesos, ya que eso es lo que se requiere. Determinando que este código puede ser reutilizable a la hora del desarrollo de la aplicación, para el ahorro del tiempo.

Retrospectiva

Se muestra que el software es válido, se debe realizar el proceso de aprendizaje de todas las neuronas de los modelos previamente estipulados, lo cual se debe mostrar el valor que debería ser en base a los cálculos matemáticos, el valor que calcula la neurona y el error calculado de las diferencias de los tiempos.

Sprint 6

Planeación

Se trae el aprendizaje de las neuronas lo cual se obtienen a partir de una matriz de datos previamente dados, según Torres, Aldana, Piñero & Piedra (2016) “Las redes neuronales basadas en los modelos interactivos y competencia pueden ser utilizadas tanto para el aprendizaje supervisado como el no supervisado.”(p.215) lo cual ejecuta el aprendizaje con el código susodicho, para digitar la matriz el programa de aprendizaje hizo cálculos de aprendizaje diez mil (10.000) ciclos y con todos los datos que se necesitan de entradas y salidas, lo cual trabaja de manera que al finalizar el aprendizaje la red neuronal compruebe que ha aprendido con todos los valores que ha obtenido en la matriz de aprendizaje, lo cual los datos de aprendizaje se dan de manera incremental en el tiempo y el valor de la cantidad de automóviles en el semáforo se dan de manera aleatoria tomando los datos(véase en la tabla 11), se adaptan para que no sobrepasen dichos valores. Las matrices de aprendizaje poseen 1440 columnas y 3 filas de los cuales la primera fila son las horas del día desde 00:00 expresado en decimal (0) hasta 23:59 expresado en decimal (2359). Las neuronas de los semáforos aprenderán según lo estipulado en los sprint anteriores, en la cual todos los semáforos aprenderán, pero el semáforo 1.2 posee un carril y su número máximo de autos

que tuvo fue de 7, lo cual sin importar la hora la cantidad de automóviles esperando en ese semáforo será de 7.

Para ver el aprendizaje de los semáforos

Tabla 12. Matriz modelo de aprendizaje para las neuronas de las capas de entrada

Semáforo XX			
Hora del día	Cantidad de automóviles	Tiempo con el algoritmo	Tiempo resultado de la neurona
0	0	20	X
1	2	21,8	X
2	2	21,8	X
3	0	20	X
4	3	22,7	X
.	.	.	.
.	.	.	.
.	.	.	.
2359	0	20	X

Fuente. Propia

Según el modelo de la matriz de aprendizaje para cada neurona, se obtuvo que los pesos cambian, es decir que, en cada ejecución del código de aprendizaje de las neuronas, como los datos de la matriz son diferentes, los pesos cambian.

Las neuronas de la capa de salida de cada red neuronal, se utiliza los datos nueva mente aleatorios desde 20 segundos y hasta 65 segundos expresado en la normalización (véase en la tabla 10), la matriz tendrá el valor real, pero la neurona aprenderá con sus valores normalizados, teniendo en cuenta que en su resultado deseado es el promedio de toda la columna.

Tabla 13. Matriz modelo de aprendizaje para las neuronas de la capa de salida

	Paso X							
Numero	Semáforo xx	Semáforo xy	Semáforo xz	.	.	Semáforo zz	Tiempo con el algoritmo	Tiempo resultado de la neurona
1	35	35	20	.	.	25	X	Y
2	24	41	24	.	.	46	X	Y
3	45	32	64	.	.	34	X	Y
4	20	23	42	.	.	37	X	Y
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
100	25	34	23	.	.	45	X	Y

Fuente. Propia

Teniendo en cuenta estos modelos (véase en las tablas 12 y 13), Del cual se determinaron el como resultado de todo el proceso de aprendizaje de las neuronas de la capa de entrada y la capa de salida, se obtuvieron diversos pesos debido a que la matriz de aprendizaje es aleatoria y no es una predefinida los cuales dan a conocer el aprendizaje de las redes neuronales.

Tabla 14. Pesos sinápticos de las neuronas de las capas de entrada

Pesos de las neuronas de la capa de entrada		
Semáforo	Cantidad automóviles	Hora del día
Semáforo 1.1	4,56317292526233	-3,51815383150552
Semáforo 1.2	7,34730132800794	-3,04627815406866
Semáforo 1.3	4,51001239833458	-3,52971634080554
Semáforo 1.4	4,42724215672502	-3,59091278551203
Semáforo 2.1	4,44477659910419	-3,59936808183651
Semáforo 2.2	4,43635314042539	-3,49733626708772
Semáforo 2.3	4,60442951387787	-3,57456443112968
Semáforo 3.1	4,47008873125888	-3,54261962596203
Semáforo 3.2	4,41975560473460	-3,55369792015238
Semáforo 3.3	4,33694976537321	-3,56306374670375
Semáforo 3.4	4,47414344548475	-3,56161806505700

Fuente. Propia

Tabla 15. Pesos sinápticos de las neuronas de la capa de salida del paso 1

Pesos de las neuronas de la capa de salida en la red neuronal del paso 1	
Semáforo 1.1	0.1666666666666667
Semáforo 1.3	0.1666666666666665
Semáforo 2.2	0.1666666666666668
Semáforo 2.3	0.1666666666666663
Semáforo 3.1	0.1666666666666665
Semáforo 3.2	0.1666666666666667

Fuente. Propia

Tabla 16. Pesos sinápticos de las neuronas de la capa de salida del paso 2

Pesos de las neuronas de la capa de salida en la red neuronal del paso 2	
Semáforo 1.2	0.3333333333333333
Semáforo 2.1	0.3333333333333337
Semáforo 3.1	0.3333333333333337

Fuente: Propia

Tabla 17. Pesos sinápticos de las neuronas de la capa de salida del paso 3

Pesos de las neuronas de la capa de salida en la red neuronal del paso 3	
Semáforo 1.4	0.3333333333333337
Semáforo 2.1	0.3333333333333326
Semáforo 3.4	0.3333333333333337

Fuente: Propia

Tabla 18. Pesos sinápticos de las neuronas de la capa de salida del paso 4

Pesos de las neuronas de la capa de salida en la red neuronal del paso 4	
Semáforo 1.3	0.3333333333333337
Semáforo 2.2	0.3333333333333333
Semáforo 3.4	0.3333333333333334

Fuente. Propia

Tabla 19. Pesos sinápticos de las neuronas de la capa de salida del paso 5

Pesos de las neuronas de la capa de salida en la red neuronal del paso 5	
Semáforo 1.1	0.25
Semáforo 1.2	0.25
Semáforo 2.3	0.25
Semáforo 3.3	0.25

Fuente. Propia

Una vez simulado y hecho el aprendizaje de todas las neuronas se puede tomar el cómo se va a realizar el simulador, teniendo en base todos los pesos (véase en las tablas

14,15,16,17,18 y 19) los cuales serán introducidos en el simulador, en vez de colocar los datos aleatoriamente, como se hace en el simulador (véase en el anexo 1 y 3), permitiendo así poder desarrollar el simulador.

Revisión

En este punto se acordó como pesos oficiales a los mostrados en las tablas (véase en las tablas 14,15,16,17,18 y 19), lo cual deben ser implementados en el simulador, tomando en cuenta el tiempo de ejecución sobre la prueba del aprendizaje de las neuronas, lo cual deben ser un tiempo corto, para que la ejecución de dicho simulador no se tarde en desarrollar los cálculos de las neuronas.

Retrospectiva

Se da inicio al desarrollo del simulador, lo cual contará primero con base del paradigma de programación orientado a objetos, se debe realizar el código de los objetos de las neuronas de la capa de entrada y capa de salida, lo cual cumplan con el paradigma de programación estipulado.

Sprint 7

Planeación

Teniendo en cuenta el paradigma de programación orientado a objetos según Santaolaya, Fragoso, Zamudio (2014) “Si tenemos ciertas clases y ciertas relaciones en el marco orientado a objetos, entonces estas combinaciones específicas de clases y relaciones soportarán la funcionalidad ofrecida por el marco orientado a objetos.” (p.191) lo cual en el contexto al que se quiere aplicar en el sistema semaforización del simulador, donde la programación orientada a objetos da mayor solución al diseño de las clases de las capas de

entrada y las capas de salida, por lo cual tener el control de todas las redes neuronales en el simulador.

Se plantea el código de las clases de las capas de entrada y las capas de las redes neuronales artificiales (véase en los anexos 4), lo cual muestra que en objetos de las clases se asigna primero en paso donde se encuentra actualmente el simulador, dividido solo en 5 datos prediseñados (Paso 1, Paso 2, Paso 3, Paso 4, Paso 5), después se asigna los datos que vienen de la clase principal y/o de donde se realice el envío de datos de la cantidad de autos que almacenan los semáforos, después se agrega a un vector de tamaño dinámico, según de paso donde se encuentre, los pesos de las entradas de cada espacio del vector es un semáforo, el cual se especifica con anterioridad, lo cual los pesos están asignados de manera secuencial al envío de los datos de cada semáforo, es decir que los pesos están ordenados según su semáforo, para no perder la identidad de cada semáforo.

Pasando del objeto se encuentra la clase de `calculo_semaforo` el cual, es el método que calcula y hace la operación de cada una de las neuronas de la capa de salida dependiendo de la red neuronal en la que esté, la red neuronal se diferencia en los pasos del objeto, es decir que cada paso es una red neuronal, lo cual infiere la sincronización de los semáforos sobre la simulación. La variable `z` es un vector que lo que hace es que en cada uno de sus espacios almacena el umbral de cada neurona de la capa de entrada de la red neuronal actual, de ese umbral proceden todos los datos matemáticos que se utilizan en la prueba de aprendizaje las neuronas (véase en el anexo 1), tiene la característica de que no tienen un algoritmo que calcule el tiempo para el cálculo del error, lo cual ahorrará tiempo en la ejecución del simulador.

Una vez visto la función de `calculo_semaforo` se tienen la función de `tomar_hora()` la cual se encarga de tomar la hora de la maquina en la que se está ejecutando el simulador y retorna la hora en formato decimal, para su normalización en el simulador.

`Neurona_salida()` es la función que, cuando es llamada crea el constructor de la clase de las capas de salida de la red neuronal en la que se encuentre actualmente. Enviando el paso en donde se encuentra y el valor calculado en la función de `calculo_semaforo` donde y acumula como z los valores de la función de activación, donde llama al método de `tiempo_final` de la clase de las capas de salida lo cual da un valor normalizado, por ende, se debe despejar para que pueda ser traducido para el usuario y el simulador en tiempo en segundos de 20 segundos hasta 65 segundos.

La función de `CreaVector()` recibe un numero el cual, es el tamaño que requiera el vector que se va a crear, y retorna el vector lleno de ceros para que sea utilizado.

En la clase de las capas de salida de la red neuronal (véase en el anexo 5), la función del objeto recibe dos variables, el paso en el que se encuentra actualmente, lo cual es también en que red neuronal se encuentra el simulador actualmente, y esta sufre una pequeña variación en los pesos, ya que se cargan los pesos de la capa de salida de la neurona en la que se encuentre. Se asignan los datos resultantes de la capa de entrada, lo cual dependiendo de la red neuronal en la que se encuentre cambiará el tamaño del vector.

En la función de `tiempo_final()` se calcula el tiempo que va a tener la red neuronal actual, lo cual z es una variable y no un vector, como en la clase de las capas de entrada, entonces genera después de la función de activación el valor del tiempo normalizado de la red neuronal, retornado ese valor a la función de `Neurona_salida` donde allí se normaliza

Revisión

Se analiza los códigos de los objetos, tiene una desventaja el cual no hay una clase en la que se puede utilizar los métodos lo cual se tiene incertidumbre del funcionamiento de los objetos, pero sin errores de sintaxis, serían errores de cálculo de asignación, pero partiendo de la base que no se puede comprobar

Retrospectiva

Se requiere hacer la interface gráfica del simulador, la cual debe cumplir con la maqueta estipulada y los requisitos del simulador, por lo tanto, se deben realizar las pruebas de los objetos en la interface gráfica, por ende, probar el correcto funcionamiento de la interface con el código de los objetos dando así una sinergia entre los componentes del simulador, tomaría como referencia el buen uso de la interface como pilar del simulador.

Sprint 8

Planeación

Una vez teniendo las clases de las neuronas de las capas de entrada y de salida, se empezó a diseñar la interface gráfica del simulador lo cual, debe ser lo más cercano a su maqueta (véase en la figura 24) teniendo en cuenta su funcionabilidad y si vista agradable al usuario que ejecute este simulador, Según Garay (2015)

Se denomina Interfaz de Usuario a la interacción entre la computadora y el usuario. Por su parte el Webster Dictionary define el concepto de Interfaz como: “el lugar en que sistemas independientes se reúnen y actúan sobre o se comunican entre sí.” (p. 202)

La interface de usuario debe ser la interacción del simulador, de las redes neuronales, de la cantidad de autos, con el usuario lo cual se plantea el desarrollo de la interface y su codificación (véase en el anexo 6).

Imagen 5. Interface del simulador



Fuente. Propia

Lo cual la interface del simulador es igual a la maqueta planteada con anterioridad, lo cual todas las funcionalidades que se propusieron en el sprint 3 sobre la maqueta, se pudieron aplicar al simulador, el cual en la primera parte cuenta con cuatro imágenes de los semáforos para guiar al usuario de que está pasando en el simulador.

Imagen 6. Semáforo en verde



Fuente. Propia

Imagen 7. Semáforo en rojo



Fuente. Propia

Imagen 8. Semáforo en amarillo



Fuente. Propia

Imagen 9. Semáforo en rojo y amarillo



Fuente. Propia

Lo cual se mostraron a varias personas sobre el entendimiento de la interface (incluyendo profesores de la universidad, estudiantes, etc.) y dieron su aprobación a las imágenes explicativa de los semáforos.

Lo cual se obtuvo las imágenes de simulador que a su vez también son guías para la identificación de los semáforos y su estado con base a las imágenes anteriores.

Imagen 10. Inicio del simulador



Fuente. Propia

Imagen 11. Paso uno del simulador



Fuente. Propia

Imagen 12. Entre el paso uno y el paso dos del simulador



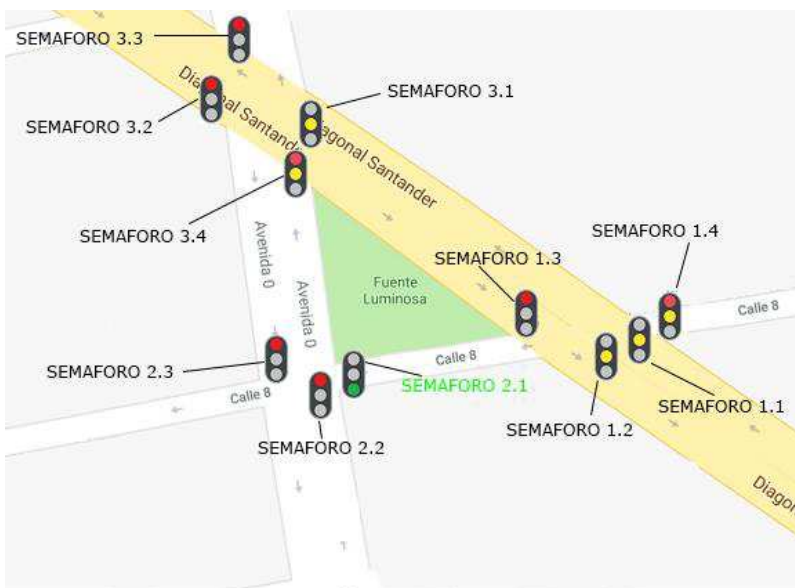
Fuente. Propia

Imagen 13. Paso dos del simulador



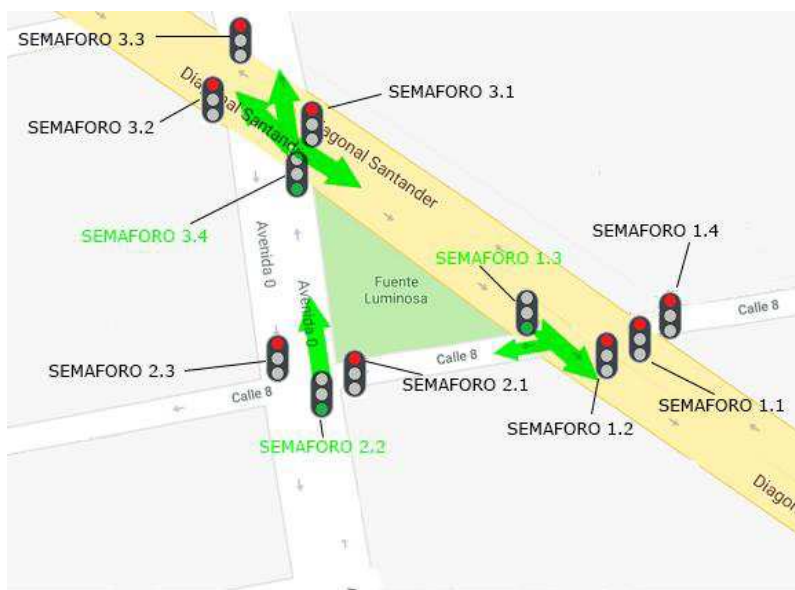
Fuente. Propia

Imagen 14. Entre el paso dos y el paso tres del simulador



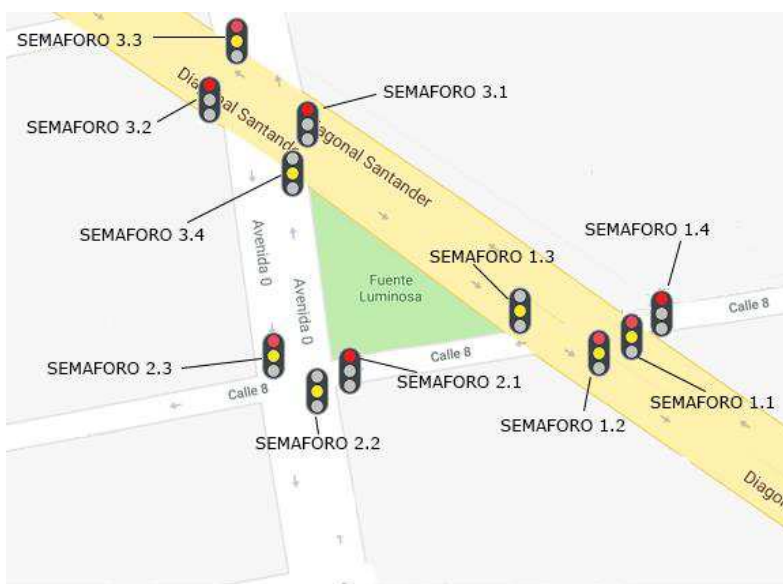
Fuente. Propia

Imagen 17. Paso cuatro del simulador



Fuente. Propia

Imagen 18. Entre el paso cuatro y el paso quinto del simulador



Fuente. Propia

Imagen 19. Paso quinto del simulador



Fuente. Propia

Imagen 20. Entre el paso quinto y el paso uno del simulador



Fuente. Propia

Lo cual se tiene en cuenta la sincronización y de manera iterativa y secuencial de las imágenes, lo cual también la imagen de inicio del simulador solo sale al comienzo del simulador y después de pulsar el botón de simular esta ya no volverá a aparecer.

Y cumpliendo con todas las estipulaciones hechas en los anteriores sprint se debe revisar, y aceptar la interface de usuario con todos lo que ha sido necesario en el simulador, se puede dar por terminada el desarrollo del simulador, y listo para hacer las pruebas.

El simulador hace un registro en la carpeta registros, el cual crea un archivo de texto plano el cual agrega cada una de las simulaciones realizadas, en la cual guarda la hora el dia, el paso en el que está simulando, los datos según el paso en el que simule, y el resultado de la red neuronal en segundos y con decimales. El archivo se guarda cuando se cierra el simulador.

Revisión

Se ha revisado satisfactoriamente la interface de usuario la cual cumple las expectativas propuestas con anterioridad, lo cual se da vía a las

Retrospectiva

Se pide que al realizar las pruebas del simulador estas queden registradas para tener control sobre lo que realiza el simulador, y hacer las pruebas del simulador en diferentes horas, lo cual se puede analizar sus resultados.

Pruebas

Para esta sección se tomaron las pruebas según Rojas, Pino & Martínez (2015) “La industria del software reconoce hoy la importancia de llevar a cabo pruebas de software, como instrumento para asegurar la calidad de los productos desarrollados” (p. 57), con los datos arrojados por el simulador, el simulador de tiempos diferentes concorde a los datos que le ingresan, lo cual sean favorables, las pruebas del simulador se utilizó en la medida del tiempo en las horas pico, en horas de la tarde, y en horas valle donde el tráfico es diferente en cada uno de los campos, lo cual se demostró el aprendizaje y la conclusión de cada red neuronal con base a la cantidad de automóviles y la hora del día.

Se realizaron 10 pruebas a diferentes horas del día lo que genera que los datos sean diferentes unos de otros, lo que dio variedad de lo que se requiere en el simulador.

Tabla 20. Primera prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
08:47:56	1	[8, 0, 2, 7, 9, 10]	38.7451311706763
08:48:35	2	[1, 13, 10]	42.59470944802378
08:49:17	3	[2, 11, 10]	41.06757744617573
08:49:59	4	[10, 7, 4]	40.06132546660484
08:50:52	5	[2, 6, 4, 0]	36.00387006557512

Fuente. Propia

Tabla 21. Segunda prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
10:00:51	1	[8, 15, 14, 11, 13, 12]	46.032474220164396
10:01:38	2	[0, 8, 8]	35.80300465248986
10:02:14	3	[15, 7, 9]	42.53816718923319
10:02:57	4	[15, 2, 3]	37.55011232407833
10:03:55	5	[11, 6, 8, 6]	40.91062658481454

Fuente. Propia

Tabla 22. Tercera prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
11:16:18	1	[17, 22, 16, 8, 4, 26]	48.18955143425629
11:17:09	2	[6, 7, 23]	45.46676315644743
11:17:58	3	[10, 26, 6]	44.77067444354786
11:18:45	4	[21, 4, 12]	43.95161005072757
11:19:34	5	[28, 4, 11, 6]	43.841935670752925

Fuente. Propia

Tabla 23. Cuarta prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
12:00:32	1	[18, 23, 23, 17, 16, 11]	51.344403225831115
12:01:24	2	[4, 8, 20]	42.033108846150824
12:02:07	3	[29, 5, 29]	51.44640336210993
12:03:05	4	[17, 27, 1]	46.10430631192835
12:03:53	5	[3, 0, 0, 10]	30.459687919309513

Fuente. Propia

Tabla 24. Quinta prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
12:31:58	1	[4, 23, 7, 1, 14, 7]	37.92592247862545
12:32:37	2	[1, 17, 21]	44.64941300344495
12:33:38	3	[13, 1, 8]	34.99122054643372
12:34:14	4	[25, 25, 22]	58.13631359038108
12:35:15	5	[27, 6, 2, 18]	45.019465631724444

Fuente. Propia

Tabla 25. Sexta prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
13:30:21	1	[2, 0, 11, 19, 29, 25]	43.67591960500266
13:31:34	2	[1, 16, 13]	38.825489450209346
13:32:20	3	[26, 1, 22]	46.57894080464018
13:33:23	4	[25, 26, 13]	53.02322603941916
13:34:28	5	[1, 1, 4, 19]	33.44343981096695

Fuente. Propia

Tabla 26. Séptima prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
14:50:04	1	[7, 5, 22, 9, 5, 9]	34.94833428361968
14:50:43	2	[6, 15, 12]	39.60390477295043
14:51:25	3	[4, 6, 17]	34.41087155565417
14:52:01	4	[19, 19, 23]	51.475600707671205
14:52:54	5	[10, 6, 19, 11]	39.908170208644094

Fuente. Propia

Tabla 27. Octava prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
18:00:41	1	[13, 23, 7, 15, 20, 0]	36.51675272437382
18:01:20	2	[4, 5, 8]	27.582742984703604
18:01:49	3	[10, 16, 17]	36.213646377919304
18:02:34	4	[7, 13, 26]	38.65568053097371
18:03:14	5	[20, 1, 14, 26]	40.185186137461216

Fuente. Propia

Tabla 28. Novena prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
19:01:01	1	[7, 16, 10, 24, 23, 12]	37.440910957819526
19:01:39	2	[3, 26, 5]	34.72158130484307
19:02:13	3	[0, 6, 29]	34.71104734837002
19:02:49	4	[16, 26, 25]	47.44322979934907
19:03:44	5	[22, 1, 24, 28]	44.488637014291626

Fuente. Propia

Tabla 29. Decima prueba del simulador

Hora	Paso	Cantidad de automóviles	Tiempo obtenido
20:31:23	1	[8, 5, 13, 8, 11, 6]	26.88784567243564
20:31:57	2	[2, 11, 13]	28.104830160321598
20:32:32	3	[10, 6, 10]	26.533634284279895
20:32:59	4	[2, 10, 4]	24.740032401756494
20:33:24	5	[9, 3, 9, 2]	25.66359097917328

Fuente. Propia

Dado estas pruebas de resultados el proceso del simulador, los resultados, tiempos de respuestas, según Jústiz, Gómez & Delgado (2014)

La calidad del software debe tenerse en cuenta durante todo el proceso de producción del mismo, por tanto, deben existir actividades que velen por su cumplimiento durante todo el ciclo de vida. Es una mala práctica esperar a que el software esté terminado para entonces aspirar a obtener la calidad en el mismo. (p. 132)

Se da el simulador como válido por el cumplimiento de la calidad del software sobre todo lo que se necesita, se da el proyecto como modelo valido para un sistema de semaforización inteligente en la zona de la fuente luminosa ya que cumple con las expectativas planteadas al inicio del proyecto, llevando un seguimiento correcto a base de lo que se quería diseñar.

CONCLUSIONES

Para el análisis de las redes neuronales artificiales como mejor técnica para el diseño de un sistema de semaforización, se debe escoger una función, en este caso una función que se acople mejor para un semáforo, demostrando ser el operador inteligente más adecuado para la fuente luminosa, lo cual realizan en mejor medida la optimización de tiempos y la optimización del flujo vehicular en esta zona, la fuente luminosa. Se tiene en como consecuencia el uso dinámico de los tiempos de los semáforos de una manera no tan sencillo, puesto que no puede ser los semáforos tener un tiempo de libre albedrío, sino que deben trabajar todos en conjuntos si es que en verdad se quiere dar solución a los problemas de atascos en la zona. Sabiendo esto, se quiere implementar una sincronización a los semáforos de ese sector, ya sea, para una mejor circulación en los ciertos casos en las calles que interceden en la fuente luminosa. Esta es una de las maneras más eficientes que puede llegar a dar los semáforos para poder hacer el flujo óptimo de vehículos y evitar la congestión vehicular en esa zona.

La metodología de desarrollo ágil Scrum en el momento de desarrollar un simulador, en este caso fue excelente, ya que un simulador necesita de un desarrollo estable, cíclico y retroalimentativo, para que haya un funcionamiento óptimo y una eficiencia bastante alta, lo que quiere decir que plantea una retroalimentación sobre la inteligencia artificial y el desarrollo de software.

Utilizando Python para el desarrollo del simulador, actúa de manera muy eficiente, ya que se utiliza el paradigma de programación orientada a objetos, lo cual ha rendido de manera significativa en el proyecto.

El procedimiento de tomar una decisión de cómo va a ser el tipo de redes neuronales la cual van a ser redes perceptrón multicapa la cual se trabajó con dos capas (entrada y salida) la cual se plantearon los modelos de redes neuronales para la implementación del sistema de semaforización inteligente para la zona de la fuente luminosa, basados en la sincronización, funcionan de manera correcta y eficiente a lo esperado.

Utilizando la función de activación sigmoidea y la lineal, lo cual se necesita normalizar los valores de entrada, teniendo en cuenta la hora del día se especifica el funcionamiento de la cantidad de automóviles.

Las pruebas dan resultados concordes a lo esperado según el estudio, entre la cantidad de automóviles aumente el resultado de la salida también aumentará, lo mismo sucederá dependiendo de la hora del día, sus horas pico, se da viable la utilización de este modelo de redes neuronales para la zona de la fuente luminosa, lo cual mejorará el flujo vehicular en esta zona.

RECOMENDACIONES

Se recomienda probar, indagar y utilizar otro operador inteligente para la simulación para un nuevo modelo de un sistema de semaforización inteligente en otra zona de la ciudad, también no solo tomando la salida (tiempo de los semáforos) sin que tomando prioridades de los semáforos habiendo que en vez de asignar un tiempo en específico se asigne mejor un orden cambiante.

Se recomienda utilizar una nueva sincronización, en la fuente luminosa, para que, en la parte de las redes neuronales, se cambien los modelos, y así allá un nuevo uso a aquellas redes neuronales ya estipuladas.

Se recomienda utilizar otro tipo de red neuronal lo cual, su aprendizaje sea diferente a ya propuesto en el documento, y que sea toda la red neuronal la que aprenda simultáneamente y no individualmente como se propone.

REFERENCIAS BIBLIOGRÁFICAS

- Angulo, E., (2011). *“Política fiscal y estrategia como factor de desarrollo de la mediana empresa comercial sinaloense. Un estudio de caso”*. (Tesis doctoral). Universidad autónoma de Sinaloa facultad de contaduría y administración, Sinaloa, México recuperado de <http://www.eumed.net/tesis-doctorales/2012/eal>
- Arana, L., & Ruiz, M., & La Serna, N. (2015). Análisis de aplicaciones empleando la computación en la nube de tipo PaaS y la metodología ágil Scrum. *Industrial Data*, 18 (1), 149-160. Recuperado de <http://www.redalyc.org/articulo.oa?id=81642256017>
- Ávila, E., & Meneses, A., & De la Cruz, V. (2013). Delfdroid: metodología ágil de desarrollo de software para dispositivos móviles. *Revista INGENIERÍA UC*, 20 (3), 59-70. Recuperado de <http://www.redalyc.org/articulo.oa?id=70732641008>
- Bances, M., y Ramos. M., (2014). Semáforos inteligentes para la regulación del tráfico vehicular. *Ingeniería: Ciencia, Tecnología e Innovación*, 1(1), 37-45, recuperado de <http://revistas.uss.edu.pe/index.php/ING/article/view/113/215>
- Benguría, S., Alarcón, B., Valdés, M., Pastellides, P. y Gómez, L., (2010). *Observación. Métodos de investigación en educación especial*, Universidad Autónoma de Madrid, Madrid, recuperado de https://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/Observacion_trabajo.pdf

- Buitrón, S., & Pino, F. (2015). Transformación del conocimiento en el proceso de elicitación de requisitos no funcionales. *El Hombre y la Máquina*, (46), 130-136. Recuperado de <http://www.redalyc.org/articulo.oa?id=47843368015>
- Calderón, F., Forero, A., Quiroga, A. y Urrego, G., (2009). Adquisición de variables de tráfico vehicular usando visión por computador. *Revista de Ingeniería*, (30) 7-15. Recuperado de <http://www.redalyc.org/articulo.oa?id=121015710006>
- Casas, J., Repullo, J., Donaldo, J., (2003). La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos (II). *casas Anguita J, et al.* 31(9) 592-600 Recuperado de <http://www.unidaddocentemfyclaspalmas.org.es/resources/10+Aten+Primaria+2003.+La+Encuesta+II.+Cuestionario+y+Estadistica.pdf>
- Ceballos, F., & Muñoz, L., & Moreno, J. (2011). Selección de perceptrones multicapa usando aprendizaje bayesiano. *Scientia Et Technica*, 16 (49), 110-115. Recuperado de <http://www.redalyc.org/articulo.oa?id=84922625019>
- Celis, J., y Escobar, C., y Sepúlveda, S., y Castro, S., & Medina, B., y Ramírez, J. (2016). Control adaptativo para optimizar una intersección semafórica basado en un sistema embebido. *Ingeniería y Ciencia*, 12 (24), 169-193. Recuperado de <http://www.redalyc.org/articulo.oa?id=83548395008>
- Challenger, I., & Díaz, Y., & Becerra, R. (2014). El lenguaje de programación Python. *Ciencias Holguín*, 20 (2), 1-13. Recuperado de <http://www.redalyc.org/articulo.oa?id=181531232001>

- Curvelo, J., & de Araújo, S., & Biazus, J., & de Souza, R. (2015). Simulación del proceso de biodegradación de aguas residuales de la industria de carne mediante una red neuronal artificial perceptrón multicapa. *Ingeniare. Revista Chilena de Ingeniería*, 23 (2), 269-275. Recuperado de <http://www.redalyc.org/articulo.oa?id=77236977011>
- Díaz, L. (2011). La observación. Universidad nacional autónoma de México. Facultad de psicología. Recuperado de http://www.psicologia.unam.mx/documentos/pdf/publicaciones/La_observacion_Lidia_Diaz_Sanjuan_Texto_Apoyo_Didactico_Metodo_Clinico_3_Sem.pdf
- Garay, M. (2015). Interfaces Inteligentes en el aprendizaje de la Modelación. *Ingeniería Industrial*, 36 (2), 200-215. Recuperado de <http://www.redalyc.org/articulo.oa?id=360441056008>
- García, Y., & Rodríguez, R., & Taboada, A. (2017). Mapeo de imágenes digitales de fondo de ojo atendiendo a rasgos de textura. *Revista Cubana de Ciencias Informáticas*, 11 (1), 106-121. Recuperado de <http://www.redalyc.org/articulo.oa?id=378349711008>
- González, L., & Vidaud, I. (2009). Factores para evaluar la viabilidad de proyectos de conservación de edificaciones esenciales, no productivas, en zonas sísmicas. *Ingeniería*, 13 (1), 25-39. Recuperado de <http://www.redalyc.org/articulo.oa?id=46713055003>
- Guzmán, R., y Álvaro, W. (2013). *Elaboración e implementación de una maqueta prototipo de semáforo inteligente para la intersección de dos avenidas* (tesis de grado). Quito, 2013. Recuperado a partir de <http://bibdigital.epn.edu.ec/handle/15000/6360>

Hardy, T; (2001). (IA: Inteligencia Artificial). POLIS, *Revista Latinoamericana*, 1(2) 1-23.

Recuperado de <http://www.redalyc.org/articulo.oa?id=30500219>

Hernández, R., y Fernández, C., y Baptista, P., (2006). Metodología de la investigación 4ta

edición, México. Recuperado de [http://files.especializacion-](http://files.especializacion-tig.webnode.com/200000775-097910b6c0/sampieri-et-al-metodologia-de-la-investigacion-4ta-edicion-sampieri-2006_ocr.pdf)

[tig.webnode.com/200000775-097910b6c0/sampieri-et-al-metodologia-de-la-](http://files.especializacion-tig.webnode.com/200000775-097910b6c0/sampieri-et-al-metodologia-de-la-investigacion-4ta-edicion-sampieri-2006_ocr.pdf)

[investigacion-4ta-edicion-sampieri-2006_ocr.pdf](http://files.especializacion-tig.webnode.com/200000775-097910b6c0/sampieri-et-al-metodologia-de-la-investigacion-4ta-edicion-sampieri-2006_ocr.pdf)

Hurtado, L., & Villarreal, E., & Villarreal, L. (2016). Detección y diagnóstico de fallas

mediante técnicas de inteligencia artificial, un estado del arte. *Dyna*, 83 (199), 19-

28. Recuperado de <http://www.redalyc.org/articulo.oa?id=49648868002>

Jústiz, D., & Gómez, D., & Delgado, M. (2014). Proceso de pruebas para productos de

software en un laboratorio de calidad. *Ingeniería Industrial*, 35(2), 131-

145. Recuperado de <http://www.redalyc.org/articulo.oa?id=360433597003>

La Opinión. (22 de abril del 2016). Regularización: a contar carros venezolanos. *La Opinión*.

Recuperado de [https://www.laopinion.com.co/cucuta/regularizacion-contar-carros-](https://www.laopinion.com.co/cucuta/regularizacion-contar-carros-venezolanos-110546#OP)

[venezolanos-110546#OP](https://www.laopinion.com.co/cucuta/regularizacion-contar-carros-venezolanos-110546#OP)

Leiva, I., & Villalobos, M. (2015). Método ágil híbrido para desarrollar software en

dispositivos móviles. *Ingeniare. Revista Chilena de Ingeniería*, 23 (3), 473-

488. Recuperado de <http://www.redalyc.org/articulo.oa?id=77241115016>

Leyton, J., & Rodríguez, M., & Correa, J. (2014). Efectos laborales vinculados al uso de

técnicas de inteligencia artificial. *Universidad & Empresa*, 16 (26), 211-249.

Recuperado de <http://www.redalyc.org/articulo.oa?id=187232713010>

- López, B., (2007). Introducción a la inteligencia artificial. Instituto Tecnológico de Nuevo Laredo, Tamps, México. Recuperado de <http://www.itnuevolaredo.edu.mx/takeyas/Articulos/Inteligencia%20Artificial/ARTICULO%20Introduccion%20a%20la%20Inteligencia%20Artificial.pdf>
- Lucena, C., (2014). Modernización de la semaforización en Colombia, un reto en los planes de movilidad. *Revista Digital Apuntes de investigación*, 8(1),1-12, recuperado de https://www.sistemamid.com/panel/uploads/biblioteca/2015-05-01_10-10-40121156.pdf
- Madariaga, C., & Rivero, Y., & Leyva, A. (2016). Propuesta metodológica para desarrollo de software educativo en la Universidad de Holguín. *Ciencias Holguín*, 22 (4), 1-17. Recuperado de <http://www.redalyc.org/articulo.oa?id=181548029003>
- Mantilla, L., Márquez, E., (2014). *Sistema de control difuso de semáforos para mejorar el tráfico vehicular en el centro histórico Trujillo*. (Tesis de grado). Universidad nacional de Trujillo. Recuperado de <http://www.inf.unitru.edu.pe/revistas/2014/8.pdf>
- Martínez, F., Hernández, C., & López, D., (2012). Sistema de comunicación TCP/IP para el control de una intersección de tráfico vehicular. *Ingeniería Investigación y Tecnología*. 14(4), 583-594 recuperado de <http://www.scielo.org.mx/pdf/iit/v14n4/v14n4a11.pdf>
- Martínez, M., (2012). *Semáforos inteligentes*. (Proyecto de investigación). Universidad Católica Nuestra Señora de la Asunción, Facultad de Ciencias y Tecnologías, Asunción, Paraguay, recuperado de http://jeuazarru.com/wp-content/uploads/2014/10/semaforos_inteligentes.pdf

- Mercado, D., Pedraza, L., y Martínez, E. (2015). Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo. *Prospectiva*, 13(2), 88-95, recuperado de <http://www.redalyc.org/articulo.oa?id=496250642011>
- Meza, L., (2010). *El paradigma positivista y la concepción dialéctica del conocimiento*. Escuela de Matemática, Instituto Tecnológico de Costa Rica, Costa Rica, recuperado de <http://revistas.tec.ac.cr/index.php/matematica/article/viewFile/2296/2087>
- Mitre, H., & Ortega, E., & Lemus, C. (2014). Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. *Ingeniería. Investigación y Tecnología*, 15(3), 403-418. Recuperado de <http://www.redalyc.org/articulo.oa?id=40431741007>
- Molina, A., Bejarano, E., Robles, H. & Dávila, H., (2014). Segmentación de los pulmones usando máquinas de soporte vectorial en imágenes de tomografía computarizada. *Sistemas, cibernética e informática*. 11(2) 29-32. Recuperado de [http://www.iiisci.org/journal/CV\\$/risi/pdfs/CA494SL14.pdf](http://www.iiisci.org/journal/CV$/risi/pdfs/CA494SL14.pdf)
- Montenegro, D., Santos, M. & Garmendia, L., (2014). Control difuso de un semáforo. Universidad Complutense de Madrid. Facultad de ingeniería. Recuperado de https://www.researchgate.net/publication/242749011_CONTROL_DIFUSO_DE_UN_SEMAFORO
- Natareno, C., (2013). *Inteligencia artificial aplicada al tránsito en la ciudad de Guatemala utilizando redes neuronales*. (Tesis de grado). Universidad de San Carlos de Guatemala. Recuperado de http://biblioteca.usac.edu.gt/tesis/08/08_0708_CS.pdf

- Navarro, A., & Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11 (2), 30-39. Recuperado de <http://www.redalyc.org/articulo.oa?id=496250736004>
- Ospino, A., & Robles, C., & Duran, A. (2014). Modelado y simulación de un panel fotovoltaico empleando técnicas de inteligencia artificial. *Ingeniería Energética*, 35 (3), 225-233. Recuperado de <http://www.redalyc.org/articulo.oa?id=329132445007>
- Pedraza, L., Hernández, C., López, D., (2012). *Control de tráfico vehicular usando ANFIS*. Revista chilena de ingeniería 20(1) p.79-88. Recuperado de <http://www.scielo.cl/pdf/ingeniare/v20n1/art08.pdf>
- Pérez, I., & Arco, L. (2016). Una revisión sobre aprendizaje no supervisado de métricas de distancia. *Revista Cubana de Ciencias Informáticas*, 10 (4), 43-67. Recuperado de <http://www.redalyc.org/articulo.oa?id=378349316004>
- Redacción Nacional. (30 de octubre del 2015). Cúcuta tiene el índice de calidad ambiental más bajo del país. *El Espectador*. Recuperado de <https://www.elespectador.com/noticias/nacional/cucuta-tiene-el-indice-de-calidad-ambiental-mas-bajo-de-articulo-596162>
- Reinoso, A., Tocaín, C., (2009). Diseño de un prototipo para controlar un semáforo inteligente usando tecnologías GSM/GPRS y wireless CPU sobre una plataforma open-soft (Linux). (Trabajo de grado). Recuperado de <http://bibdigital.epn.edu.ec/handle/15000/1588>

- Rojas, M., & Pino, F., & Martínez, J. (2015). Proceso de pruebas para pequeñas organizaciones desarrolladoras de software. *Facultad de Ingeniería*, 24 (39), 55-70. Recuperado de <http://www.redalyc.org/articulo.oa?id=413940776006>
- Rojas, R., Vásquez, C., (2016). Análisis de los factores y su relación con la informalidad del mercado N°3-huayco, de la ciudad de Tarapoto. *Accounting power for business*. 1(2) 7-15. Recuperado de http://revistascientificas.upeu.edu.pe/index.php/ri_apfb/article/view/576/593
- Salcedo, O., Pedraza, L., Hernández, C., (2006). *Modelo de Semaforización Inteligente para la Ciudad de Bogotá*. Revista científica y tecnológica de la facultad de ingeniería, universidad distrital francisco José de caldas 11(2), 61-69 recuperado de <http://revistas.udistrital.edu.co/ojs/index.php/reving/article/view/2680/3852>
- Santaolaya, R., & Frago, O., & Zamudio, S. (2014). Modelo formal para la reestructura de marcos orientados a objetos hacia arquitecturas modelo-vista-adaptador. *Ingeniería. Investigación y Tecnología*, 15 (2), 187-198. Recuperado de <http://www.redalyc.org/articulo.oa?id=40430749003>
- Sobrevilla, G., & Hernández, J., & Velasco, P., & Soriano, S. (2017). Aplicando Scrum y Prácticas de Ingeniería de Software para la Mejora Continua del Desarrollo de un Sistema Ciber-Físico. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 6(1), 1-15. Recuperado de <http://www.redalyc.org/comocitar.oa?id=512253717001>
- Torres, S., & Aldana, M., & Piñero, P., & Piedra, L. (2016). Red neuronal multicapa para la evaluación de competencias laborales. *Revista Cubana de Ciencias*

Informáticas, 10(Supl. 1), 210-223. Recuperado de
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000500016&lng=es&tlng=es

Valencia, V., (2000). Principios de los semáforos. Universidad Nacional de Colombia sede Medellín, Facultad Nacional de Minas, recuperado de
<http://www.bdigital.unal.edu.co/12057/1/10539884.2000.pdf>

Vicario, S., & Oubram, O., & Bassam, A., & Velázquez, J., & Ordóñez, E. (2017). Sistema de reconocimiento inteligente de señales mioeléctricas del movimiento de mano humana. *Ingeniería*, 21 (2), 41-53. Recuperado de
<http://www.redalyc.org/articulo.oa?id=46753192008>

ANEXOS

Anexo 1. Código de aprendizaje de las neuronas de la capa de entrada

```

from random import randrange
import math

def creaMatriz(n,m):#metodo que crea una matriz de n * m
    matriz = []
    for i in range (0,n):
        matriz.append(['0']*m)
    return matriz

def tiempo_recorrido(f_vehicular,hora):#metodo que genera el tiempo segun el flujo vehicular y la hora
    carriles=2 #varia a 1 si hablamos del semaforo 1.2 que es el unico de los 11 que posee un carril
    distancia=f_vehicular*7/carriles #calcula la distancia dando en promedio 7 metros por cada 2
    autos segun el carril
    if (hora>=1030 and hora <=1430) or (hora>=1730 and hora<=2000):#pregunta si la hora es pico
        velocidad = 5# se asigna la velocidad mas pequeña 5 Km/h
    else:
        if hora>1430 and hora <1730:#pregunta si estamos en horas de la tarde entre las horas pico
            velocidad = 7.5 #se asigna una velocidad moderada
        else:
            velocidad = 10 #se asigna una alta velocidad porque son horas valle
    velocidad=velocidad*5/18#se hace la conversion entre km/h por m/s
    time = distancia/velocidad +20 #segun la formula de la distancia se cacula el tiempo, lo cual
    se le agregan 20 segundos
    if time>65: #si el tiempo es mayor a 65, el tiempo cambiara a 65
        time=65;
    else:
        if time<20: #si el tiempo es menor a 20, el tiempo cambiara a 20
            time=20
    return time

archivo=open("resultadosxx.txt","w")
tiempo=creaMatriz(1440,2)#creacion de matriz de tiempo
autos=creaMatriz(1440,2)#creacion de matriz de cantidad de autos
semaforo=creaMatriz(1440,3)#creacion de matriz de tiempo, cantidad de autos, y tiempo calculado

```

```

pesos=[randrange(5),randrange(5)]#pesos asignados aleatoriamente para iniciar los calculos
aux=0 #auxiliar para la insercion de los tiempos en horas y minutos
for x in range(2360):#ciclo para la la insercion de datos del tiempo
    if x%100==60:# pregunta si la ya ha llegado a 60 segundos
        x+=40#aumenta 40 unidades para dar el siguiente valor
    else:
        if x%100<60:#pregunta si el valor de x es menor a 60, procedera a insertarlo
            tiempo[aux][0]=x
            tiempo[aux][1]=x/2359#se inserta x/2359 para convertir el tiempo en valores dentre 0
y 1
            aux+=1

for x in range(0,len(autos)):#insercion de la cantidad de automoviles
    if (tiempo[x][0]>=1030 and tiempo[x][0]<=1430) or (tiempo[x][0]>=1730 and
tiempo[x][0]<=2000):#pregunta si es hora pico
        autos[x][0]=randrange(30)#asigna un numero del 0 al 30 porque 30 es el valor maximo obtenido
en la recoleccion de datos
    else:
        if tiempo[x][0]>=1430 and tiempo[x][0]<=1730:#pregunta si es hora de la tarde entre las horas
pico
            autos[x][0]=randrange(23)#asigna un valor de entre 0 al 25 ya que es el promedio de
los valores maximos obtenidos de los datos
        else:
            if tiempo[x][0]>=2200 or tiempo[x][0]<=500:#pregunta si es hora de la madrugada
                autos[x][0]=randrange(5)#asigna un valor entre 0 y 5 para esas horas
            else:
                autos[x][0]=randrange(16)#sino es ninguno de los otros casos se le asigna el
valor de entre 0 y 16 porque es el maximo tiempo en horas valle
            autos[x][1]=autos[x][0]/30# divide en 30 para convertir la cantidad de autos en calores entre
0 y 1

for x in range(0,len(semaforo)):#insercion de semaforos
    semaforo[x][0]=tiempo[x][1]#asigna los valores de tiempo y cantidad de autos entre 0 y 1
    semaforo[x][1]=autos[x][1]
    semaforo[x][2]=tiempo_recorrido(autos[x][0],tiempo[x][1])#se llama al metodo para obtener el
tiempo calculado
    semaforo[x][2]=(semaforo[x][2]-20)/(65-20)#se convierte el tiempo calculado entre 0 y 1

for x in range(0,1000):#ciclo para el aprendizaje
    for t in range(0,len(semaforo)):#ciclo los casos
        z=0# iniziacion de la sumatoria por caso
        for h in range(0,2):#ciclo de los datos por caso

```

```

        z=z+semaforo[t][h]*pesos[h]#sumatoria
    y=1/(1+math.exp(-z))#funcion de activacion sigmoidal
    error=semaforo[t][2]-y#calculo del error
    for r in range(0,len(pesos)):
        pesos[r]=pesos[r]+0.5*error*semaforo[t][r]#recalculo de los pesos
archivo.write(f"pesos1: {pesos[0]}      pesos 2: {pesos[1]}\n\n")# escritura en un texto plano los
nuevos pesos que se calcularon
for t in range(0,len(semaforo)):#ciclo de prueba de aprendizaje
    z=0#sigue las mismas instrucciones del aprendizaje pero sin el calculo del error ni recalculo
    de los pesos
    for h in range(0,2):
        z=z+semaforo[t][h]*pesos[h]
    y=1/(1+math.exp(-z))
    error=semaforo[t][2]-y;
    archivo.write(f"{tiempo[t][0]}    {autos[t][0]}    {semaforo[t][2]*(65-20)+20}    {y*(65-20)+20}
\n")#escritura de los tiempos, el primero es el resultado del aprendizaje

        #y el segundo es el esperado que se calcula a través de las formulas

archivo.close()

```

Fuente: Propia.

Anexo 2. Código de la clase principal del aprendizaje de las neuronas de la capa de salida

```

from neu_tiempo import neurona_tiempo
paso=int(input("digita el paso para el aprendizaje 1-5 "))
constructor=neurona_tiempo(paso)
constructor.llenado()
constructor.imprimir_peso()
constructor.imprimir()
constructor.aprendizaje()
constructor.prueba()

```

Fuente: Propia

Anexo 3. Código de la clase objeto del aprendizaje de las neuronas de la capa de salida

```

from random import randint
from random import randrange
class neurona_tiempo:
    def __init__(self,i):

```

```

self.paso=i
self.archivo=open(f"paso{self.paso}cccc.txt","w")
if self.paso==1:

self.pesos=[randrange(5),randrange(5),randrange(5),randrange(5),randrange(5),randrange(5)]
else:
    if self.paso==2:
        self.pesos=[randrange(5),randrange(5),randrange(5)]
    else:
        if self.paso==3:
            self.pesos=[randrange(5),randrange(5),randrange(5)]
        else:
            if self.paso==4:
                self.pesos=[randrange(5),randrange(5),randrange(5)]
            else:
                if self.paso==5:

self.pesos=[randrange(5),randrange(5),randrange(5),randrange(5)]
def aprendizaje(self):
    for x in range(0,10000):
        for h in range(0,len(self.datos)):
            z=0
            for u in range(0,len(self.pesos)):
                z=z+self.datos[h][u]*self.pesos[u]
            y=z
            error=self.datos[h][len(self.pesos)]-y
            for w in range(0,len(self.pesos)):
                self.pesos[w]=self.pesos[w]+0.5*error*self.datos[h][w]
def prueba(self):
    for x in range(0,len(self.datos)):
        z=0
        for h in range(0,len(self.pesos)):
            z=z+self.datos[x][h]*self.pesos[h]
        y=z
        self.archivo.write(f"{y} {self.datos[x][len(self.pesos)]} {y*(65-20)+20}\n")
        self.archivo.write("\n")
def llenado(self):
    self.datos=self.creaMatriz(100,len(self.pesos)+1)

```

```

        for x in range(0,len(self.datos)):
            for l in range(0,len(self.pesos)+1):
                if l==len(self.pesos):
                    self.datos[x][l]=0
                    for h in range(0,l):
                        self.datos[x][l]=self.datos[x][l]+self.datos[x][h]
                    self.datos[x][l]=self.datos[x][l]/l
                else:
                    self.datos[x][l]=(randint(20, 65)-20)/(65-20)

    def imprimir(self):
        for i in range(0,len(self.datos)):
            for j in range(0,len(self.pesos)+1):
                self.archivo.write(f"{self.datos[i][j]}      ")
            self.archivo.write("\n")
        self.archivo.write("\n\n\n\n")

    def creaMatriz(self,n,m):#metodo que crea una matriz de n * m
        matriz = []
        for i in range (0,n):
            matriz.append(['0']*m)
        return matriz

    def imprimir_peso(self):
        for x in range(0,len(self.pesos)):
            self.archivo.write(f"pesos{x}: {self.pesos[x]}\n")

```

Fuente: Propia

Anexo 4. Código de la clase de las capas de entrada del simulador

```

import math
import time

from neurona_tiempos import salida# importar la clase neurona_tiempos

class neurona_de_entrada:

    def __init__(self,datos,paso):#creacion del objeto datos [] es el vector que recibe los datos
    de la cantidad de automoviles en cada paso, y paso es el paso donde se encuentra en ese momento

        self.pasos=paso#asignacion

        self.cantidad=datos

        if self.pasos=="Paso 1":#pregunta en que paso se encuentra en ese momento

            self.pesos_tiempo=[-3.5181538315055256,-3.529716340805549,-3.4973362670877277,-
            3.5745644311296867,-3.5426196259620384,-3.5536979201523873]#asigna los pesos del tiempo segun
            su paso

```

```

self.pesos_autos=[4.56317292526233,4.510012398334587,4.436353140425394,4.604429513877873,4.4
70088731258882,4.419755604734605]#asigna los pesos de la cantidad de aautomoviles

elif self.pasos=="Paso 2":
    self.pesos_tiempo=[-3.046278154068661,-3.599368081836518,-3.5426196259620384]
    self.pesos_autos=[7.3473013280079424,4.444776599104194,4.470088731258882]

elif self.pasos=="Paso 3":
    self.pesos_tiempo=[-3.590912785512034,-3.599368081836518,-3.561618065057007]
    self.pesos_autos=[4.427242156725026,4.444776599104194,4.474143445484756]

elif self.pasos=="Paso 4":
    self.pesos_tiempo=[-3.529716340805549,-3.4973362670877277,-3.561618065057007]
    self.pesos_autos=[4.510012398334587,4.436353140425394,4.474143445484756]

elif self.pasos=="Paso 5":
    self.pesos_tiempo=[-3.5181538315055256,-3.046278154068661,-3.5745644311296867,-
3.5630637467037563]

self.pesos_autos=[4.56317292526233,7.3473013280079424,4.604429513877873,4.336949765373213]

def calculo_semaforo(self):#metodo que calcula y hace la operacion de cada una de las neuronas
dependiendo del paso en el que se encuentre

    z=self.CreaVector(len(self.pesos_tiempo))#llama a CreaVector ()para crear un vector vacio con
el tamaño de la longitud del vector de pesos_tiempo,

                                                    # tambien se
puede utilizar la longitud de pesos_autos ya que es la misma en cada asignacion por paso

    self.y=self.CreaVector(len(self.pesos_tiempo))#se llama a CreaVector() para crear el vecotr
de las respuestas a cada neurona

    hora= self.tomar_hora()#se llama al metodo tomar_hora

    for x in range(0,len(self.pesos_tiempo)):#ciclo hasta la longitud de pesos_tiempo

        z1=(self.cantidad[x]/30)*self.pesos_autos[x]#se caulcula la z1 que es el calculo de
la cantidad de autos con su respectivo peso.

        #se divide en 30, ya que es el numero maximo de autos que pueden estar en un semaforo

        z2=(hora/2359)*self.pesos_tiempo[x]#se calcula z2 que es el tiempo actual de la
maquina donde se ejecuta, este es dividido en 2359, porque es su valor maximo ess 2359

        z[x]=(z1+z2)# se realiza la sumatoria de los resultados los valores

                                                    # cada z[x] es una neurona diferente

    for x in range(0,len(self.pesos_tiempo)):

        self.y[x]=1/(1+math.exp(-z[x]))#se realiza la funcion de activacion de cada neurona

def tomar_hora(self):#metodo para tomar la hora actual del ordenador

    hora=int(time.strftime("%H"))

    minutos=int(time.strftime("%M"))

    hora= hora*100+minutos# se multiplica el valor de hora para que, el valor de hora sea en
miles ej: 1200, 2300, etc.

```

```

return hora#retorna la hora

def neurona_salida(self):#metodo que llama a la clase de la neurona_tiempos

    resultado=salida(self.pasos,self.y)#crea un constructor con el paso actual, y con el vector
    de y[], con sus valores entre 0 y 1

    resultados=resultado.tiempo_final()*(65-20)+20#se asigna el valor resultante de los calculos
    de tiempo_final multiplicado por la normalizacion

    #se hace la operacion de (65-20)+20 ya que es el despeje de la normalizacion

    #65 es el valor maximo del tiempo resultante, y 20 es el minimo valor que puede dar

    return resultados#retorna el resultado

def CreaVector(self,n):#crea un vector en 0 para su llenado de 'n' tamaño

    vector=[]

    for i in range(0,n):

        vector.append(['0'])

    return vector

```

Fuente. Propia

Anexo 5. Código de la clase de las capas de salida del simulador

```

class salida:

    def __init__(self,paso,datos):#objeto que recibe el paso donde se encuentra actualmente y los
    datos obtenidos por la capa de salida

        self.paso=paso#asignacion de los datos del objeto

        self.datos=datos

        if self.paso=="Paso 1":#pregunta en que paso esta actualmente

            self.pesos=[0.1666666666666667,0.1666666666666665,0.1666666666666668,0.1666666666666663,0.
            16666666666666652,0.1666666666666667]#asigna los pesos segun el paso en el que se encuentre y
            que entrada obtenga

            elif self.paso=="Paso 2":

                self.pesos=[0.3333333333333333,0.3333333333333337,0.3333333333333337]

            elif self.paso=="Paso 3":

                self.pesos=[0.3333333333333337,0.3333333333333326,0.3333333333333337]

            elif self.paso=="Paso 4":

                self.pesos=[0.3333333333333337,0.3333333333333333,0.3333333333333334]

            elif self.paso=="Paso 5":

                self.pesos=[0.25,0.25,0.25,0.25]

    def tiempo_final(self):#metodo que realiza la capa de salida de las redes neuronales

```

```

z=0#inicializa el simulador en 0
for x in range(0,len(self.pesos)):
    z=z+self.datos[x]*self.pesos[x] #se realiza la sumatoria de todo los datos obtenidos
    por la capa de entrada y sus respectivos pesos
y=z#se asigna la funcion de activacion ocrrespondiente a la capa de salida
return y#retorna el tiempo final para cada paso

```

Fuente. Propia

Anexo 6. Código de la clase principal del simulador

```

from tkinter import *#importar la interface grafica
import time
import math
from random import randrange
from neurona_semaforos import neurona_de_entrada#importar la clase neurona_semaforo
...

    se realizan la interface grafica con los semaforos en verde dependiendo del paso en el que se
    encuentre
...

def paso1():
    global tiempos
    global x
    global imagen
    img=x
    now =f"{tiempos} Segundos en verde."#se asigna a now los segundos en verde
    tiempos-=1;
    tiempo.configure(text=now)#se agrega al Label
    if (tiempos%2)!=0:
        imagen.destroy()
        imagen=Label(miFrame,image=img2f)
    else:
        imagen.destroy()
        imagen=Label(raiz,image=img2)
    imagen.place(x=200,y=160)
    if tiempos==-1:# pregunta si el temporizador ha finalizado
        now="En amarillo"
        tiempo.configure(text=now)#en vez de mostrar 0 segundos, muestra "en amarillo"
        imagen=Label(miFrame,image=img3)

```

```

    cambio['state']=NORMAL

    paso['text']="Del paso 1 al paso 2"

    x+=2# aumenta el control en dos para continuar la secuencia

    imagen.place(x=200,y=160)# si coloca la imagen en el frame

    return#cuando el contador llega a -1 , retorna para salir del metodo

    miframe.after(1000,paso1)#ciclo para ejecutar el contador hacia atras, 1000 milisegundos, 1
    segundo

def paso2():

    global tiempos

    global x

    global imagen

    img=x

    now =f"{tiempos} Segundos en verde."#se asigna a now los segundos en verde

    tiempos-=1;

    tiempo.configure(text=now)#se agrega al Label

    if (tiempos%2)!=0:

        imagen.destroy()

        imagen=Label(miframe,image=img4f)

    else:

        imagen.destroy()

        imagen=Label(raiz,image=img4)

    imagen.place(x=200,y=160)

    if tiempos== -1:# pregunta si el temporizador ha finalizado

        now="En amarillo"

        tiempo.configure(text=now)#en vez de mostrar 0 segundos, muestra "en amarillo"

        imagen=Label(miframe,image=img5)

        cambio['state']=NORMAL

        paso['text']="Del paso 2 al paso 3"

        x+=2# aumenta el control en dos para continuar la secuencia

        imagen.place(x=200,y=160)# si coloca la imagen en el frame

        return#cuando el contador llega a -1 , retorna para salir del metodo

    miframe.after(1000,paso2)

def paso3():

    global tiempos

    global x

    global imagen

    img=x

    now =f"{tiempos} Segundos en verde."#se asigna a now los segundos en verde

```

```

tiempos-=1;
tiempo.configure(text=now)#se agrega al Label
if (tiempos%2)!=0:
    imagen.destroy()
    imagen=Label(miiframe,image=img6f)
else:
    imagen.destroy()
    imagen=Label(raiz,image=img6)
imagen.place(x=200,y=160)
if tiempos==1:# pregunta si el temporizador ha finalizado
    now="En amarillo"
    tiempo.configure(text=now)#en vez de mostrar 0 segundos, muestra "en amarillo"
    imagen=Label(miiframe,image=img7)
    cambio['state']=NORMAL
    paso['text']="Del paso 3 al paso 4"
    x+=2# aumenta el control en dos para continuar la secuencia
    imagen.place(x=200,y=160)# si coloca la imagen en el frame
    return#cuando el contador llega a -1 , retorna para salir del metodo
miiframe.after(1000,paso3)

def paso4():
    global tiempos
    global x
    global imagen
    img=x
    now =f"{tiempos} Segundos en verde."#se asigna a now los segundos en verde
    tiempos-=1;
    tiempo.configure(text=now)#se agrega al Label
    if (tiempos%2)!=0:
        imagen.destroy()
        imagen=Label(miiframe,image=img8f)
    else:
        imagen.destroy()
        imagen=Label(raiz,image=img8)
    imagen.place(x=200,y=160)
    if tiempos==1:# pregunta si el temporizador ha finalizado
        now="En amarillo"
        tiempo.configure(text=now)#en vez de mostrar 0 segundos, muestra "en amarillo"

```

```

    imagen=Label(miframe,image=img9)
    cambio['state']=NORMAL
    paso['text']="Del paso 4 al paso 5"
    x+=2# aumenta el control en dos para continuar la secuencia
    imagen.place(x=200,y=160)# si coloca la imagen en el frame
    return#cuando el contador llega a -1 , retorna para salir del metodo
    miframe.after(1000,paso4)
def paso5():
    global tiempos
    global x
    global imagen
    img=x
    now =f"{tiempos} Segundos en verde."#se asigna a now los segundos en verde
    tiempos-=1;
    tiempo.configure(text=now)#se agrega al Label
    if (tiempos%2)!=0:
        imagen.destroy()
        imagen=Label(miframe,image=img10f)
    else:
        imagen.destroy()
        imagen=Label(raiz,image=img10)
    imagen.place(x=200,y=160)
    if tiempos==1:# pregunta si el temporizador ha finalizado
        now="En amarillo"
        tiempo.configure(text=now)#en vez de mostrar 0 segundos, muestra "en amarillo"
        imagen=Label(miframe,image=img1)
        cambio['state']=NORMAL
        paso['text']="Del paso 5 al paso 1"
        x=1# aumenta el control en dos para continuar la secuencia
        imagen.place(x=200,y=160)# si coloca la imagen en el frame
        return#cuando el contador llega a -1 , retorna para salir del metodo
    miframe.after(1000,paso5)

def num_max_aleatoreo():#metodo para recolectar el numero max de automoviles que estan a cierta hora
del dia
    entrada=neurona_de_entrada(0,"0")#llama a la clase neurona_de_entrada con valores 0 para poder
    ejecutar un metodo que esta en en esta clase
    tomar_hora= entrada.tomar_hora()#toma la hora del dia para seleccionar el numero maximo

```

```

if (tomar_hora>=1030 and tomar_hora<=1430) or (tomar_hora>=1730 and tomar_hora<=2000):#pregunta
si es hora pico

    return 30#asigna un numero del 0 al 30 porque 30 es el valor maximo obtenido en la recoleccion
de datos

else:

    if tomar_hora>=1430 and tomar_hora<=1730:#pregunta si es hora de la tarde entre las horas pico

        return 25#asigna un valor de entre 0 al 25 ya que es el promedio de los valores maximos
obtenidos de los datos

    else:

        if tomar_hora>=2200 or tomar_hora<=500:#pregunta si es hora de la madrugada

            return 5#asigna un valor entre 0 y 5 para esas horas

        else:

            return 16

def simular():#metodo de simular los semaforos por cada paso

    global x#globaliza las variables de x, imagen, tiempos

    global tiempos

    x+=1#suma 1 a x que es el contador

    max_autos=num_max_aleatoreo()#llama al metodo para obtener el numero maximo para asignarlo

    if x==2:#pregunta en que paso va a realizar

        paso['text']="Paso 1"#escribe en paso, el paso donde va "paso 1", "paso 2"

        Pasos=[randrange(max_autos),randrange(max_autos),randrange(max_autos),randrange(max_autos),r
andrange(max_autos),randrange(max_autos)]#asigna aleatoriamente los pesos

        auxilio=f" Semáforo 1.1: {Pasos[0]}\n Semáforo 1.3: {Pasos[1]}\n Semáforo 2.2: {Pasos[2]}\n
Semáforo 2.3: {Pasos[3]}\n Semáforo 3.1: {Pasos[4]}\n Semáforo 3.2: {Pasos[5]}"#se agregar los
valores a una cadena

        Lautos['text']=auxilio# se asigna la cadena al Lautos

    elif x==5:

        paso['text']="Paso 2"

        Pasos=[randrange(7),randrange(max_autos),randrange(max_autos)]

        Lautos['text']=f" Semáforo 1.2: {Pasos[0]}\n Semáforo 2.1: {Pasos[1]}\n Semáforo 3.1:
{Pasos[2]}"

    elif x==8:

        paso['text']="Paso 3"

        Pasos=[randrange(max_autos),randrange(max_autos),randrange(max_autos)]

        Lautos['text']=f" Semáforo 1.4: {Pasos[0]}\n Semáforo 2.1: {Pasos[1]}\n Semáforo 3.4:
{Pasos[2]}"

    elif x==11:

        paso['text']="Paso 4"

```

```

    Pasos=[randrange(max_autos),randrange(max_autos),randrange(max_autos)]

    Lautos['text']=f"  Semáforo  1.3:  {Pasos[0]}\n  Semáforo  2.2:  {Pasos[1]}\n  Semáforo  3.4:
{Pasos[2]}"

elif x==14:

    paso['text']="Paso 5"

    Pasos=[randrange(max_autos),randrange(7),randrange(max_autos),randrange(max_autos)]

    Lautos['text']=f"  Semáforo  1.1:  {Pasos[0]}\n  Semáforo  1.2:  {Pasos[1]}\n  Semáforo  2.3:
{Pasos[2]}\n  Semáforo  3.3:{Pasos[3]}"

    entrada=neurona_de_entrada(Pasos,paso['text'])#crear el objeto con los parametros  del paso ,
    y de los datos que se obtubieron aleatoreamente

    entrada.calculo_semaforo()#calcula el tiempo para cada semaforo en la neurona de entrada

    tiempos=entrada.neurona_salida()#calcula y asigna el tiempo calculado de toda la red neuronal
    de el paso que se realiza

    Hora=time.strftime("%H:%M:%S")#asgina a hora, la hora exacta del ordenador

    archivo.write(f"Hora:{Hora},  {paso['text']},  Cantidad de autos:{Pasos},  tiempo obtenido:
{tiempos}\n")#escribe en el archivo los datos pertinentes  para el registros

    tiempos=math.floor(tiempos)# se redondea el numero para que pueda interactuar correctamente con
    el contador

    tiempo_estatico['text']=f"{tiempos} segundos."#se asigna a tiempo_estatico  el valor de tiempo
    redondeado, como cmprobante del tiempo que ha sido calculado

    if x==2:#pregunta en que paso va a realizar

        paso1()#interface de cada paso

    elif x==5:

        paso2()

    elif x==8:

        paso3()

    elif x==11:

        paso4()

    elif x==14:

        paso5()

    cambio['state']=DISABLED#desactiva el boton para que no ejecute otra simulacion mientras se
    hace el conteo

def nombrearc():#metodo que toma el nombre del archivo del registro, realiza uno dependiendo del año,
    mes, dia, hora, minuto, y segundo del dia de la maquina

    año=time.strftime("%Y")

    mes=time.strftime("%B")

    dia=time.strftime("%d")

    hora=time.strftime("%H")

    minutos=time.strftime("%M")

    segundos=time.strftime("%S")

```

```

nom=f"registros/{año}{mes}{dia}{hora}{minutos}{segundos}.txt"#asigna todos los valores en una
cadena

return nom#retorna el nombre de la cadena

x=1 #asigna el valor inicial de x que es la variable que controla las imagens, y los pasos
nombreadarchivo=nombreadarc()#llama al metodo para obtener el nombre y lo asigna a nombreadarchivo
archivo=open(nombreadarchivo,"w")#abre un nuevo archivo con el nombre, en tipo escritura


raiz=Tk()#crea la interfaz fgrafica
miframe=Frame(raiz, width=800,height=600)#crea el frame de tamaño 800 x 600
miframe.pack()#ejecuta el frame en el Tk() en modo pack para que ocupe la GUI el tamaño del frame
raiz.title("Simulador de semáforos")#asignacion de titulo

...

    diseño de la interface grafica y contruccion de todos los Widgets de el frame

...

img=PhotoImage(file="imagenes/secuencia/principio.png")#asignacion y creacion de las imagens de la
simulacion

img1=PhotoImage(file="imagenes/secuencia/1P.png")
img2=PhotoImage(file="imagenes/secuencia/2P.png")
img3=PhotoImage(file="imagenes/secuencia/3P.png")
img4=PhotoImage(file="imagenes/secuencia/4P.png")
img5=PhotoImage(file="imagenes/secuencia/5P.png")
img6=PhotoImage(file="imagenes/secuencia/6P.png")
img7=PhotoImage(file="imagenes/secuencia/7P.png")
img8=PhotoImage(file="imagenes/secuencia/8P.png")
img9=PhotoImage(file="imagenes/secuencia/9P.png")
img10=PhotoImage(file="imagenes/secuencia/10P.png")
img2f=PhotoImage(file="imagenes/secuencia/2Pf.png")
img4f=PhotoImage(file="imagenes/secuencia/4Pf.png")
img6f=PhotoImage(file="imagenes/secuencia/6Pf.png")
img8f=PhotoImage(file="imagenes/secuencia/8Pf.png")
img10f=PhotoImage(file="imagenes/secuencia/10Pf.png")
verde=PhotoImage(file="imagenes/semaforos/semaforoverde.png")
rojo=PhotoImage(file="imagenes/semaforos/semafororojo.png")

```

```

amarillo=PhotoImage(file="imagenes/semaforos/semaforoamarillo.png")
rojoamarillo=PhotoImage(file="imagenes/semaforos/semafororojoamarillo.png")

titulo=Label(miiframe,font=("Verdana",20),text="SIMULADOR DE SEMAFOROS EN LA FUENTE LUMINOSA")# Label
de bienvenido
titulo.place(x=0,y=0)

Lverde=Label(miiframe,image=verde)
Lverde.place(x=30,y=70)
LverdeT=Label(miiframe,font=("Verdana",10),text="Semáforo en verde:\n\nAvanzar")
LverdeT.place(x=80,y=70)
Lrojo=Label(miiframe,image=rojo)
Lrojo.place(x=230,y=70)
LrojoT=Label(miiframe,font=("Verdana",10),text="Semáforo en rojo:\n\nEsperar")
LrojoT.place(x=270,y=70)
Lamarillo=Label(miiframe,image=amarillo)
Lamarillo.place(x=390,y=70)
LamarilloT=Label(miiframe,font=("Verdana",10),text="Semáforo en amarillo:\n\nSemaforo apunto
de\ncambiar a rojo")
LamarilloT.place(x=430,y=70)
Lrojoama=Label(miiframe,image=rojoamarillo)
Lrojoama.place(x=600,y=70)
LrojoamaT=Label(miiframe,font=("Verdana",10),text="Semáforo en rojo\ny amarillo:\n\nSemaforo apunto
de\ncambiar a verde")
LrojoamaT.place(x=640,y=70)

imagen=Label(miiframe,image=img)
imagen.place(x=200,y=160)

paso=Label(miiframe,font=("Verdana",13),text="Inicio")
paso.place(x=0,y=160)
LcantidadT=Label(miiframe,font=("Verdana",10),text="Cantidad de automoviles\nen cada paso:")
LcantidadT.place(x=0,y=200)
Lautos=Label(miiframe,font=("Verdana",10),text="")
Lautos.place(x=0,y=240)
Ltiempo_obtenidoT=Label(miiframe,font=("Verdana",10),text="Tiempo obtenido de\nred neuronal:")
Ltiempo_obtenidoT.place(x=0,y=360)
tiempo_estatico=Label(miiframe,font=("Verdana",10),text="")

```

```
tiempo_estatico.place(x=0,y=400)
Ltiempo_obtenido=Label(miFrame,font=("Verdana",10),text="Temporizador del \ntiempo obtenido:")
Ltiempo_obtenido.place(x=0,y=460)
tiempo=Label(miFrame,font=("Verdana",10),text="")
tiempo.place(x=0,y=500)
cambio=Button(miFrame,font=("Vernada",10),width=10,text="¡Simular!",command=simular)
cambio.place(x=30,y=545)
raiz.maxsize(800,600)
raiz.mainloop()#metodo que mantiene el Tk() activo en pantalla
archivo.close()#cierra el archivo el registro de la simulacion
```