 **INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

**Sistema de reconocimiento de patrones de movilidad vial y peatonal para ciudades inteligentes**

**PROYECTO TERMINAL**

**Que para obtener el título de:**

“Ingeniero en Telemática”

Presentan

**Beltrán Alvarado Edgar**

**Flores Vargas Gerardo**

Asesores:

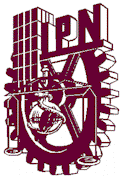
**Dra. Cyntia Eugenia Enríquez Ortiz**

**Dra. Benina Velázquez Ordoñez**

**Dr. Jesús Manuel Olivares Ceja**



Ciudad de México, junio 2020

   **INSTITUTO POLITÉCNICO NACIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS**

**Sistema de reconocimiento de patrones de movilidad vial y peatonal para ciudades inteligentes**

**PROYECTO TERMINAL**

**Presentan**

Beltrán Alvarado Edgar

Flores Vargas Gerardo

M. en C. Maricela Serrano Fragoso Presidente

Dr. Miguel Félix Mata Rivera

Secretario

Dra. Cyntia Eugenia Enríquez Ortiz Asesor 1

Dr. Jesús Manuel Olivares Ceja Asesor 2

Dra. Benina Velázquez Ordoñez Asesor 3



**Agradecimientos**

**Edgar Beltrán Alvarado**

Agradezco al Instituto Politécnico Nacional, a la Unidad Profesional Interdisciplinaria de Ingeniería y Tecnologías Avanzadas y al Centro de Investigación en Computación por prepararme para ser ingeniero y permitirme tener una carrera.

Agradezco con mucho cariño a mi mamá Sofía, por su amor, trabajo y ejemplo de perseverancia que me ha permitido alcanzar esta meta profesional y tener valores como persona.

Agradezco a mis asesores, Dra. Cyntia Eugenia Enríquez Ortiz, Dra. Benina Velázquez Ordoñez y Dr. Jesús Manuel Olivares Ceja, por sus enseñanzas diversas, tanto profesionales como personales para formarme como ingeniero y por brindarme su apoyo durante el desarrollo de este proyecto terminal.

Agradezco el apoyo incondicional y compañía de mis amigos de UPIITA y del CIC, en particular a: Jorge Antonio Juárez Leyva (Tony). Les deseo mucho éxito profesional.

**Gerardo Flores Vargas**

A mi familia, por permitirme salir adelante con todas las adversidades que se presentaron.

A la UPIITA, por sus enseñanzas y alentarme a investigar más allá de lo que nos enseñan.

A mis asesores, quienes me brindaron su valiosa orientación y guía en la elaboración de este proyecto terminal.

A mis amigos de la UPIITA, quienes me han apoyado incondicionalmente.

**Resumen**

Desde mediados del siglo XX se ha observado un crecimiento acelerado en diversas ciudades del mundo, generando diversos problemas, entre ellos los de movilidad de grupos de personas que utilizan el vehículo privado para trasladarse de un lugar a otro. Esto da como resultado que se incremente el tráfico vial. Una alternativa de solución integrando sistemas de información y telecomunicaciones ocurre en el contexto de ciudades inteligentes (*smart city*).

Los primeros trabajos de ciudades inteligentes únicamente consideraron usar las redes de comunicaciones, sin embargo, recientemente algunas propuestas tanto públicas como privadas han proporcionado alternativas de salud, medio ambiente, movilidad, gobierno, entre otras. Actualmente existen algunos trabajos que se han centrado en encontrar patrones de movilidad en forma no supervisada con base en los lugares en que los usuarios realizan llamadas telefónicas y/o datos de GPS.

Este trabajo aborda la problemática de movilidad en ciudades inteligentes, recopilando datos de ubicación de un grupo de usuarios mediante una aplicación móvil. Con la información recopilada se propone desarrollar dos algoritmos de aprendizaje automático (*machine learning*) para detectar patrones frecuentes de movilidad y correlación en las trayectorias**.**

Con este trabajo se pretende poder encontrar personas que pueden compartir sus medios de transporte en forma no supervisada. El procesamiento de información genera indicadores clave (KPI) de movilidad reflejados en un tablero de control (*dashboard*), con el propósito de presentarlo a personal de gobierno que se encargue de tomar decisiones relacionadas con la movilidad en una ciudad.

**Palabras claves**

*Smart city, dashboard*, KPI de movilidad, patrones de movilidad, compartir automóvil, *machine learning.*

**Índice General**

[**Introducción** 1](#_Toc39347369)

[**Capítulo 1** 3](#_Toc39347370)

[**1.1 Planteamiento del problema** 3](#_Toc39347371)

[**1.2 Propuesta de solución** 4](#_Toc39347372)

[**1.2.1 Módulo de Obtención de datos** 4](#_Toc39347373)

[**1.2.2 Módulo de Soluciones ciudadanas** 6](#_Toc39347374)

[**1.2.3 Módulo de Indicadores de gobierno** 6](#_Toc39347375)

[**1.3 Alcances** 6](#_Toc39347376)

[**1.4 Objetivo general** 6](#_Toc39347377)

[**1.4.1 Objetivos específicos** 7](#_Toc39347378)

[**2.1 UPIITA** 9](#_Toc39347379)

[**2.2 Nacionales** 10](#_Toc39347380)

[**2.3 Internacionales** 11](#_Toc39347381)

[**2.4 Aplicaciones comerciales** 12](#_Toc39347382)

[**2.5 Discusión del estado del arte** 13](#_Toc39347383)

[**Capítulo 3. Marco teórico** 15](#_Toc39347384)

[**3.1 Aprendizaje automatico (*Machine Learning*)** 15](#_Toc39347385)

[**3.1.1 Aprendizaje Supervisado** 15](#_Toc39347386)

[**3.1.2 Aprendizaje No supervisado** 15](#_Toc39347387)

[**3.2 Sistema de Información Geográfica (SIG)** 17](#_Toc39347388)

[**3.3 Sistema de coordenadas geográficas** 17](#_Toc39347389)

[***3.*4Fórmula de Haversine** 18](#_Toc39347390)

[**3.5 Colaboración masiva (*Crowdsourcing*)** 19](#_Toc39347391)

[**3.6 Sensado masivo de dispositivos móviles (*Mobile* *Crowdsensing*)** 19](#_Toc39347392)

[**3.7 Tablero de Control (*Dashboard*) Gobierno de datos** 20](#_Toc39347393)

[**3.8 Metodología de Cascada (*Waterfall*)** 20](#_Toc39347394)

[**3.9 Diagramas de diseño orientado a objetos** 21](#_Toc39347395)

[**3.9.1 Caso de uso** 22](#_Toc39347396)

[**3.9.2 Elementos de un diagrama de clases** 22](#_Toc39347397)

[**3.9.3 Elementos de un diagrama de secuencia** 24](#_Toc39347398)

[**3.9.4 Elementos de un diagrama de flujo** 25](#_Toc39347399)

[**Capítulo 4. Análisis** 27](#_Toc39347400)

[**4.1 Detección de Requerimientos.** 27](#_Toc39347401)

[**4.1.1 Requerimientos del módulo de obtención de datos** 27](#_Toc39347402)

[**4.1.2 Requerimientos del módulo de soluciones ciudadanas** 30](#_Toc39347403)

[**4.1.3 Requerimientos del módulo de indicadores de gobierno** 31](#_Toc39347404)

[**4.2 Análisis de requerimientos** 33](#_Toc39347405)

[**4.2.1 Análisis de Requerimientos del módulo de obtención de datos** 33](#_Toc39347406)

[**4.2.2 Análisis de Requerimientos del módulo de soluciones ciudadanas** 35](#_Toc39347407)

[**4.2.3 Análisis de Requerimientos del módulo de indicadores de gobierno** 36](#_Toc39347408)

[**4.3 Herramientas para el desarrollo de la propuesta** 37](#_Toc39347409)

[**4.3.1 Elección de herramientas para la aplicación móvil** 37](#_Toc39347410)

[**4.3.2 Elección de herramientas para el servidor web** 40](#_Toc39347411)

[**4.3.3 Elección de herramientas para el módulo de soluciones ciudadanas** 41](#_Toc39347412)

[**4.3.4 Herramientas de visualización** 43](#_Toc39347413)

[**4.3.5 Seguridad en la aplicación desarrollada** 44](#_Toc39347414)

[**Capítulo 5. Diseño** 45](#_Toc39347415)

[**5.1 Diseño de algoritmos del sistema propuesto** 45](#_Toc39347416)

[**5.1.1 Algoritmo de muestreo** 45](#_Toc39347417)

[**5.1.2 Algoritmo de detección de patrones de movilidad individual** 50](#_Toc39347418)

[**5.1.3 Algoritmo de Búsqueda de viajes compartidos** 57](#_Toc39347419)

[**5.2 Casos de uso del sistema propuesto** 72](#_Toc39347420)

[**5.2.1 Caso de uso: Registrarse** 72](#_Toc39347421)

[**5.2.2 Caso de uso: Iniciar sesión** 73](#_Toc39347422)

[**5.2.3 Caso de uso: Muestreo** 75](#_Toc39347423)

[**5.2.4 Caso de uso: Detección de movilidad individual** 76](#_Toc39347424)

[**5.2.5 Caso de uso: Búsqueda de recomendaciones de movilidad** 78](#_Toc39347425)

[**5.2.6 Caso de uso: Visualización** 79](#_Toc39347426)

[**5.3 Clases del sistema propuesto** 82](#_Toc39347427)

[**5.3.1 Diagrama de clases** 82](#_Toc39347428)

[**5.4 Base de datos del sistema propuesto** 83](#_Toc39347429)

[**5.4.1 Diagrama de base de datos** 83](#_Toc39347430)

[**5.5 Diagramas de secuencia del sistema propuesto** 84](#_Toc39347431)

[**5.5.1 Diagrama de secuencia del algoritmo de muestreo** 84](#_Toc39347432)

[**5.5.2 Diagrama de secuencia del algoritmo de detección de patrones de movilidad individual** 85](#_Toc39347433)

[**5.5.3 Diagrama de secuencia del algoritmo de búsqueda de recomendaciones para compartir transporte** 85](#_Toc39347434)

[**5.6 Cálculo de KPIs y despliegue en el tablero de control** 86](#_Toc39347435)

[**5.7 Diseño de vistas de la aplicación móvil** 88](#_Toc39347436)

[**6. Implementación** 90](#_Toc39347437)

[**6.1 Implementación de la aplicación móvil** 91](#_Toc39347438)

[**6.1.1 Implementación de las pantallas de la aplicación móvil** 91](#_Toc39347439)

[**6.1.2 Implementación del muestreo en la aplicación móvil** 96](#_Toc39347440)

[**6.1.3 Implementación del almacenamiento interno en la aplicación móvil** 97](#_Toc39347441)

[**6.1.4 Implementación de la transferencia de datos al servidor Web de la aplicación móvil** 97](#_Toc39347442)

[**6.1.5 Implementación de la sugerencia para compartir automóvil en la aplicación móvil** 98](#_Toc39347443)

[**6.1.6 Implementación del esquema de seguridad en la aplicación móvil** 99](#_Toc39347444)

[**6.2 Implementación del servidor Web** 100](#_Toc39347445)

[**6.2.1 Implementación de la obtención de los datos del muestreo en el servidor Web** 100](#_Toc39347446)

[**6.2.2 Implementación de la sugerencia para compartir automóvil en el servidor Web a la aplicación móvil** 100](#_Toc39347447)

[**6.3 Implementación del procesamiento analítico** 101](#_Toc39347448)

[**6.3.1 Implementación de la descarga de la base de datos** 101](#_Toc39347449)

[**6.3.2 Implementación de la limpieza de datos** 102](#_Toc39347450)

[**6.3.3 Implementación del algoritmo de detección de patrones de movilidad individual** 103](#_Toc39347451)

[**6.3.3 Implementación del algoritmo de viajes compartidos** 105](#_Toc39347452)

[**6.3.4 Implementación de la transferencia de datos al servidor Web** 107](#_Toc39347453)

[**6.4 Implementación del tablero de control** 108](#_Toc39347454)

[**7. Pruebas y resultados** 113](#_Toc39347455)

[**7.1 Escenario de pruebas** 113](#_Toc39347456)

[**7.2 Datos de Prueba** 113](#_Toc39347457)

[**7.3 Pruebas** 114](#_Toc39347458)

[**7.3.1 Casos para obtener sugerencia de compartir automóvil** 114](#_Toc39347459)

[**7.3.2 KPIs del tablero de control** 115](#_Toc39347460)

[**7.4 Resultados** 116](#_Toc39347461)

[**Conclusiones** 117](#_Toc39347462)

[**Trabajos a Futuro** 118](#_Toc39347463)

[**Referencias** 119](#_Toc39347464)

[**Anexos** 124](#_Toc39347465)

[**1 Análisis para obtener la muestra de una población Finita** 124](#_Toc39347466)

[**2 Encuesta realizada a la población de UPIITA** 125](#_Toc39347467)

[**Referencias Asesores** 132](#_Toc39347468)

Índice de Figuras

[**Figura 1.** Módulo principal de la propuesta 5](#_Toc39347469)

[**Figura 2.** Antes y después de utilizar K-means 16](#_Toc39347470)

[**Figura 3.** Diferencia de centroides entre K-medoids y K-means 16](#_Toc39347471)

[**Figura 4.** Representación de latitudes, longitudes, el Ecuador y el Meridiano de Greenwich 18](#_Toc39347472)

[**Figura 5.** Metodología de cascada 20](#_Toc39347473)

[**Figura 6.** Ejemplo de herencia 24](#_Toc39347474)

[**Figura 7.** Componentes del módulo de obtención de datos 28](#_Toc39347475)

[**Figura 8.** Componentes del módulo de soluciones ciudadanas 30](#_Toc39347476)

[**Figura 9.** Componentes del módulo de indicadores de gobierno 32](#_Toc39347477)

[**Figura 10.** Porcentaje de usuarios Android en el mercado, tomada de [45] 38](#_Toc39347478)

[**Figura 11.** Diagrama de flujo del algoritmo de muestreo. 47](#_Toc39347479)

[**Figura 12.** Resultado de la distancia entre dos puntos geográficos utilizando Java. 51](#_Toc39347480)

[**Figura 13.** Resultado de la distancia entre dos puntos geográficos utilizando Google MyMaps 52](#_Toc39347481)

[**Figura 14.** Obtención de la primera derivada 53](#_Toc39347482)

[**Figura 15.** Diseño del algoritmo de detección de tipo de movilidad por usuario 56](#_Toc39347483)

[***Figura 16.*** *Parte 1 del diseño del algoritmo para calcular viajes compartidos* 60](#_Toc39347484)

[***Figura 17.*** *Parte 2 del diseño del algoritmo para calcular viajes compartidos* 61](#_Toc39347485)

[***Figura 18.*** *Diseño del proceso predefinido Función Sim-Haversine para el algoritmo para calcular viajes compartidos* 62](#_Toc39347486)

[***Figura 19.*** *Diseño del proceso predefinido Función Búsqueda-Haversine para el algoritmo para calcular viajes compartidos* 63](#_Toc39347487)

[**Figura 20.** Escenario válido 1 67](#_Toc39347488)

[**Figura 21.** Escenario descartado 1 68](#_Toc39347489)

[**Figura 22.** Escenario descartado 2 69](#_Toc39347490)

[**Figura 23.** Escenario descartado 3 70](#_Toc39347491)

[**Figura 24.** Escenario descartado 4 71](#_Toc39347492)

[**Figura 25.** Diagrama de caso de uso - Registrarse 72](#_Toc39347493)

[**Figura 26.** Diagrama de caso de uso - Iniciar sesión 74](#_Toc39347494)

[**Figura 27.** Diagrama de caso de uso - Muestreo 75](#_Toc39347495)

[**Figura 28.** Diagrama de caso de uso - Detección de movilidad individual 77](#_Toc39347496)

[**Figura 29.** Diagrama de caso de uso - Búsqueda de recomendaciones para compartir transporte 78](#_Toc39347497)

[**Figura 30.** Diagrama de caso de uso - Tablero de control 80](#_Toc39347498)

[**Figura 31.** Diagrama de clases del sistema propuesto 82](#_Toc39347499)

[**Figura 32.** Diagrama de base de datos del sistema propuesto 83](#_Toc39347500)

[**Figura 33.** Diagrama de secuencia del muestreo 84](#_Toc39347501)

[**Figura 34.** Diagrama de secuencia del algoritmo de detección de patrones de movilidad individual 85](#_Toc39347502)

[**Figura 35.** Diagrama de secuencia del algoritmo de búsqueda de recomendaciones para compartir transporte 85](#_Toc39347503)

[**Figura 36.** Propuesta del tablero de control 87](#_Toc39347504)

[**Figura 37.** Vista principal 88](#_Toc39347505)

[**Figura 38.** Vista registrarse 88](#_Toc39347506)

[**Figura 39.** Vista iniciar sesión 89](#_Toc39347507)

[**Figura 40.** Vista muestreo 89](#_Toc39347508)

[**Figura 41.** Implementación del sistema para compartir automóvil 90](#_Toc39347509)

[**Figura 42.** Pantalla Portada de la aplicación móvil 91](#_Toc39347510)

[**Figura 43.** Pantalla Login de la aplicación móvil 92](#_Toc39347511)

[**Figura 44.** Pantalla Registro de la aplicación móvil 92](#_Toc39347512)

[**Figura 45.** Pantalla Pasajero de la aplicación móvil 92](#_Toc39347513)

[**Figura 46.** Pantalla Conductor de la aplicación móvil 93](#_Toc39347514)

[**Figura 47.** Pantalla Muestreo de la aplicación móvil 93](#_Toc39347515)

[**Figura 48.** Pantalla Perfil de la aplicación móvil 94](#_Toc39347516)

[**Figura 49.** Pantalla Menú Sugerencia por Día de la aplicación móvil 94](#_Toc39347517)

[**Figura 50.** Pantalla Aceptar Sugerencia de la aplicación móvil 95](#_Toc39347518)

[**Figura 51.** Pantalla Menú Sugerencia de la aplicación móvil 95](#_Toc39347519)

[**Figura 52.** Pantalla Mapa Sugerencia de la aplicación móvil 96](#_Toc39347520)

[**Figura 53.** Pantalla Notificación WhatsApp de la aplicación móvil 96](#_Toc39347521)

[***Figura 54.*** *Número total de usuarios registrados en el sistema clasificado por escuelas y carreras* 108](#_Toc39347522)

[***Figura 55.*** *Número total de usuarios registrados en el sistema clasificados por género masculino y femenino* 108](#_Toc39347523)

[**Figura 56.** Número total de usuarios registrados en el sistema clasificados por peatones y conductores 109](#_Toc39347524)

[**Figura 57.** Rango de edades de los usuarios registrado en el sistema 109](#_Toc39347525)

[***Figura 58.*** *Selección del periodo de tiempo para la visualización de los datos en el tablero de control* 110](#_Toc39347526)

[**Figura 59.** Mapa de trayectorias 110](#_Toc39347527)

[**Figura 60.** Mapa de calor 111](#_Toc39347528)

[**Figura 61.** Mapa para visualizar el patrón de movilidad individual del usuario 111](#_Toc39347529)

[**Figura 62.** Mapa para visualizar la sugerencia de compartir automóvil 112](#_Toc39347530)

[**Figura 63.** Trayectoria candidata para compartir automóvil en el tablero de control 115](#_Toc39347531)

[***Figura 64.*** *Trayectoria candidata para compartir automóvil en la aplicación móvil* 115](#_Toc39347532)

[**Figura 65.** Escuelas registradas por los usuarios de la aplicación móvil 115](#_Toc39347533)

Índice de Tablas

[**Tabla 1.** Definiciones, acrónimos y abreviaturas 0](#_Toc39347534)

[**Tabla 2.** Comparación de proyectos terminales en la UPIITA 10](#_Toc39347535)

[**Tabla 3.** Proyectos y artículos internacionales que abordan temas similares 12](#_Toc39347536)

[**Tabla 4.** Comparación de diversas aplicaciones de servicios de movilidad 13](#_Toc39347537)

[**Tabla 5.** Componentes de un diagrama de caso de uso 21](#_Toc39347538)

[**Tabla 6.** Componentes de la descripción de un caso de uso 23](#_Toc39347539)

[**Tabla 7.** Símbolos de un diagrama de secuencia 25](#_Toc39347540)

[**Tabla 8.** Símbolos para elaborar un diagrama de flujo 26](#_Toc39347541)

[**Tabla 9.** Características de Android y iOS 37](#_Toc39347542)

[**Tabla 10.** Características de Android y iOS 38](#_Toc39347543)

[**Tabla 11.** Comparación entre Java y Kotlin 39](#_Toc39347544)

[**Tabla 12.** Comparación entre 000webhost y Freehostia 40](#_Toc39347545)

[**Tabla 13.** Comparación entre los lenguajes Java, Scala, Python y R 41](#_Toc39347546)

[**Tabla 14.** Comparación entre Apache Spark y Apache Hadoop 42](#_Toc39347547)

[**Tabla 15.** Datos obtenidos de la aplicación móvil 45](#_Toc39347548)

[**Tabla 16.** Tabla de datos - ubicación 46](#_Toc39347549)

[**Tabla 17.** Tabla de datos - perfil 48](#_Toc39347550)

[**Tabla 18.** Tamaño de los tipos de datos del gestor de base de datos MySQL 8.0 48](#_Toc39347551)

[**Tabla 19.** Tamaño de las variables utilizadas en la tabla de datos ubicación 49](#_Toc39347552)

[**Tabla 20.** Estimación del almacenamiento de datos de 1 usuario 49](#_Toc39347553)

[**Tabla 21.** Estimación del almacenamiento de datos de 131 usuario 50](#_Toc39347554)

[**Tabla 22.** Tabla de datos - ubicación 50](#_Toc39347555)

[**Tabla 23.** Tabla de datos - ubicación y distancia 52](#_Toc39347556)

[**Tabla 24.** Tabla de datos - ubicación, distancia y velocidad 54](#_Toc39347557)

[**Tabla 25.** Condiciones para determinar el tipo de movilidad del usuario 54](#_Toc39347558)

[**Tabla 26.** Tipo de movilidad del usuario 54](#_Toc39347559)

[**Tabla 27.** Caso de uso - Registrarse 72](#_Toc39347560)

[**Tabla 28.** Caso de uso - Iniciar sesión 74](#_Toc39347561)

[**Tabla 29.** Caso de uso - Muestreo 75](#_Toc39347562)

[**Tabla 30.** Caso de uso - Detección de movilidad individual 77](#_Toc39347563)

[**Tabla 31.** Caso de uso - Búsqueda de recomendaciones para compartir transporte 78](#_Toc39347564)

[**Tabla 32.** Caso de uso - Tablero de control 80](#_Toc39347565)

[**Tabla 33.** Usuarios Propuestos para la preparación de datos de prueba 113](#_Toc39347566)

[**Tabla 34.** Rutas cotidianas de los usuarios **propuestos** para la preparación de datos de prueba 114](#_Toc39347567)

[**Tabla 33.** Escuelas y carreras registradas en el sistema 116](#_Toc39347568)

[**Tabla 34.** Descripción de cada variable de la ecuación para calcular la muestra de una población Finita. 124](#_Toc39347569)

[**Tabla 35.** Distribución de 124](#_Toc39347570)

[**Tabla 38.** Porcentaje de la población de UPIITA entrevistada de género hombre y mujer. 125](#_Toc39347571)

[**Tabla 39.** Porcentaje de la población entrevistada de cada una de las carreras disponibles en la UPIITA. 126](#_Toc39347572)

[**Tabla 40.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a la escuela usando transporte público. 126](#_Toc39347573)

[**Tabla 41.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a su hogar usando transporte público. 127](#_Toc39347574)

[**Tabla 42.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar escuela usando automóvil. 127](#_Toc39347575)

[**Tabla 43.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a su casa usando automóvil. 128](#_Toc39347576)

[**Tabla 44**. Porcentaje de estudiantes de UPIITA que cuentan con automóvil. 128](#_Toc39347577)

[**Tabla 45.** Porcentaje de estudiantes de UPIITA que compartirían su automóvil. 129](#_Toc39347578)

[**Tabla 46.** Porcentaje de estudiantes de UPIITA que no disponen de automóvil privado. 129](#_Toc39347579)

[**Tabla 47.** Porcentaje de estudiantes de UPIITA compartirían sus datos en forma anónima. 130](#_Toc39347580)

Índice de fragmentos de código

[**Fragmento del código 1.** Muestreo en la aplicación móvil 97](#_Toc39347581)

[**Fragmento del código 2.** Almacenamiento interno en la aplicación móvil 97](#_Toc39347582)

[**Fragmento del código 3.** Transferencia de datos de la aplicación móvil al servidor Web 98](#_Toc39347583)

[**Fragmento del código 4.** Visualización del mapa de sugerencia para compartir automóvil en la aplicación móvil 99](#_Toc39347584)

[**Fragmento del código 5.** Cifrado de la aplicación móvil 100](#_Toc39347585)

[**Fragmento del código 6.** Almacenamiento de las ubicaciones de los usuarios en el Servidor Web 100](#_Toc39347586)

[**Fragmento del código 7.** Transferencia de la sugerencia para compartir automóvil a la aplicación móvil 101](#_Toc39347587)

[**Fragmento del código 8.** Descargar la base de datos del servidor Web 101](#_Toc39347588)

[**Fragmento del código 9.** Limpieza de datos de la base de datos ubicación 103](#_Toc39347589)

[**Fragmento del código 10.** Fórmula de Haversine utilizando Scala 103](#_Toc39347590)

[**Fragmento del código 11.** Algoritmo de detección de patrones de movilidad individual utilizando Scala 105](#_Toc39347591)

[**Fragmento del código 12.** Algoritmo para sugerir viajes compartidos utilizando Scala 107](#_Toc39347592)

[**Fragmento del código 13.** Transferencia de la base de datos sugerencia al servidor Web 108](#_Toc39347593)

Definiciones, acrónimos y abreviaturas

La tabla 1, contiene las definiciones, acrónimos o abreviaturas utilizados en el documento.

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| ARF | Análisis de Requerimientos Funcionales |
| ARNF | Análisis de Requerimientos No Funcionales |
| ART | Análisis de Requerimientos Técnicos |
| BD | Base de Datos |
| Clúster | Sistema distribuido de computadoras |
| CRUD | Crear, Leer, Actualizar y Borrar (Por sus siglas en inglés *Create, Read, Update and Delete*) |
| Desplazamiento | Variación de la posición inicial y final de un objeto. Es representada por la letra griega (delta) Δ.  Ejemplo: la variación de x se escribe Δx.  Magnitud vectorial. |
| Distancia | Longitud total que recorre un objeto durante la variación de su posición inicial y final.  Unidad de medida: []  Magnitud escalar. |
| IEEE | *Institute of Electrical and Electronics Engineers* |
| KPI | Indicador Clave de Rendimiento (*Key Performance Indicator*) |
| RF | Requerimiento Funcional |
| RNF | Requerimiento No Funcional |
| RT | Requerimiento Técnico |
| Usuario | Persona que usa la aplicación móvil |
| Velocidad | Cantidad de espacio recorrido por un intervalo de tiempo de un objeto que se desplaza en una determinada dirección y sentido  Razón de cambio de la posición.  Unidad de medida: []  Magnitud vectorial. |
| Ruta frecuente | Se considera que una ruta de un usuario es una ruta frecuente, si existe al menos los datos de ubicaciones de 3 rutas de un mismo día de la semana con fechas distintas. |

**Tabla 1.** Definiciones, acrónimos y abreviaturas

# **Introducción**

Actualmente, la población mundial rebasa los siete mil quinientos millones de habitantes, concentrando en grandes ciudades a la mayor parte de estas personas. La población rural tiende a moverse hacia las áreas urbanas. En [1] se estima que esta dinámica social causará que hacia el año 2050, el 75% de la población vivirá en alguna ciudad.

Como una forma de resolver la problemática del crecimiento de la población en las grandes ciudades, durante 2014 y 2015 se propusieron ideas para crear ciudades inteligentes «Smart Cities» con los propósitos de:

* Mejorar la calidad de vida de los ciudadanos al mismo tiempo que se reduce el gasto energético y la emisión de contaminantes.
* Incrementar la seguridad.
* Mejorar la productividad de sus ciudadanos.
* Agilizar la gestión de servicios de gobierno.

El logro de estos propósitos [2], se pretende mediante el aprovechamiento de la información generada por los ciudadanos al moverse como peatones y en medios de transporte, realización de sus actividades cotidianas como compras y trámites, así como el uso de dispositivos electrónicos conectados a redes de comunicaciones.

Actualmente no existe alguna ciudad que pueda considerarse como «Smart City», sin embargo, desde el año 2015 se elaboró una lista con 100 ciudades con potencial para convertirse en ciudades inteligentes [2]. Las ciudades con mejores posibilidades para llegar a considerarse como inteligentes, son aquellas que cuentan con servicios de telecomunicaciones con cobertura para la mayoría de los ciudadanos.

En la prestación de servicios de una ciudad inteligente, los sistemas de información forman parte importante porque permiten el procesamiento de datos para elaborar los tableros de control (*dashboard*) que contienen la visualización de los indicadores clave de rendimiento (KPI, *Key Performance Indicator*) que les permiten a los encargados de la toma de decisiones gubernamentales contar con una base de información de calidad. Actualmente [3], varios de los sistemas de información que sustentan los servicios de una ciudad inteligente se encuentran en desarrollo. La mayoría se están aplicando en forma separada, esto es, sin conformar un sistema integral. Se espera en el futuro que estos funcionen en forma holística.

El interés en el desarrollo de ciudades inteligentes ha motivado la propuesta de diferentes modelos [3], entre estos:

* El modelo europeo.
* El modelo de la empresa Intel.
* El modelo de Singapur.

El modelo europeo [4] considera seis aspectos a desarrollar: economía (*economy*), gestión de gobierno (*governance*), ciudadanía (*people*), entorno (*environment*), calidad de vida (*living*) y movilidad (*mobility*). Actualmente proyectos como Horizonte 2020, 2030 y 2050 incluyen el aspecto de movilidad; debido a que es un problema importante que afecta a los ciudadanos de diferentes niveles socioeconómicos.

El modelo de la empresa Intel [5] atiende cuatro aspectos: movilidad (*mobility*), cuidado de la salud (*healthcare*), seguridad pública (*public safety*) y productividad (*productivity*).

El modelo de Singapur [6] cubre siete aspectos: transporte y movilidad (*transport and mobility*), sostenibilidad(*sustainability*), gestión de gobierno (*governance*), economía de la innovación (*innovation economy*), digitalización (*digitalization*), estándar de vida (*living standard*) y percepción experta (*expert perception*).

Es posible notar en los modelos de ciudad inteligente que el problema de la movilidad es común en ellos. En este trabajo se aborda la problemática de la movilidad en grandes ciudades, teniendo como caso particular a las ciudades inteligentes. La propuesta a desarrollar requiere que se cuente con infraestructura de comunicaciones que permita monitorear la movilidad de sus ciudadanos.

Algunas aplicaciones existentes para resolver problemas de movilidad en grandes ciudades son: Waze y Google Maps. Google utiliza datos recabados de personas moviéndose en la ciudad, sin que se tenga alguna preferencia específica. Por su parte, Waze recaba informes acerca de lo que sucede en su ciudad que proporcionan sus subscriptores; con esta información proporciona a sus usuarios rutas alternativas, con base en la densidad de tráfico detectado. Otras compañías como Moovit o Ecobici, ofrecen alternativas de movilidad que aprovechan caminos alternos a los automóviles. En las opciones de transporte privado, Blablacar, UberPool y WazeCarpool ofrecen el servicio compartido entre usuarios que no se conocen previamente; el propósito de estos viajes es reducir el tránsito vehicular en movimiento; representando una mejora en la generación de tráfico respecto a los servicios personalizados de transporte basados en dispositivos móviles como: Uber, Cabify o EasyTaxi.

En este proyecto terminal, se propone un sistema de viajes compartidos de forma no supervisada. El propósito de la propuesta es reducir la densidad de tráfico, mediante la disminución de la cantidad de personas que viajan en transporte privado. La información recabada por el sistema permitirá obtener indicadores clave (KPI) para las entidades que toman decisiones relacionadas con la planeación urbana. La información que se les presenta es relativa a la movilidad como zonas con mayor densidad de población (mapa de calor), volumen de personas (estimadas) que se movilizan en algún medio de transporte o caminando.

Este documento se estructura en siete capítulos. El primer capítulo aborda el problema de la movilidad en grandes ciudades. El segundo capítulo está conformado por el estado del arte. En el tercer capítulo se desarrolla el marco teórico que sustenta este trabajo. El cuarto capítulo contiene el análisis de requerimientos que se diseñan en el quinto capítulo. En el sexto capítulo se desarrolla la codificación del sistema. El séptimo capítulo contiene las pruebas realizadas y los resultados obtenidos. Al final se comentan las conclusiones y se incluyen las referencias del trabajo elaborado.

# **Capítulo 1**

En este capítulo se plantea el problema de movilidad en grandes ciudades y se propone una alternativa considerando viajes compartidos. Se exponen los alcances de la propuesta junto con el objetivo general y sus particulares.

## **1.1 Planteamiento del problema**

Actualmente, más de la mitad de la población del mundo vive en zonas urbanas; situación que se agrava debido a la migración del campo a las ciudades, generado un crecimiento acelerado en muchas ciudades.

Una alternativa de solución a la problemática del crecimiento de la población en las grandes ciudades que algunos grupos de especialistas dieron a conocer durante 2014 y 2015 son las ciudades inteligentes o «Smart Cities». En estas conversaciones se plantearon propuestas para mejorar la calidad de vida de los ciudadanos al mismo tiempo que se reduzca el gasto energético y la emisión de contaminantes, se incremente la seguridad, se mejorar la productividad de sus ciudadanos y se agilice la gestión de servicios de gobierno. Estas alternativas se enmarcaron en algunos modelos de ciudad inteligente, destacando el modelo europeo [4], Intel [5] y el de Singapur [6]. En estos modelos, la movilidad es una problemática de interés común.

En este trabajo se desarrolla una alternativa al problema de movilidad en las grandes ciudades en las que grupos de personas utilizan transporte privado o público para trasladarse a sus destinos: trabajos, escuelas, centros comerciales, entre otros sitios de interés. En estos viajes, frecuentemente se requiere recorrer distancias de varios kilómetros. El uso desmesurado de vehículos privados ha provocado que en estas ciudades exista aglomeración en las vías de transporte, teniendo como resultado tráfico en el flujo vehicular denso y, por lo tanto, incremento en la contaminación al medio ambiente. Actualmente, en las ciudades inteligentes propuestas [2], aún las entidades encargadas de establecer soluciones de movilidad no cuentan con información suficiente que les permita tomar decisiones adecuadas para resolver esta problemática.

Además, existen propuestas tecnológicas que se han desarrollado proponiendo su propia solución en el aspecto que desean mejorar de la movilidad, sin embargo, no se ha realizado una propuesta que utilice patrones de movilidad para disminuir la cantidad de personas que viajan en el transporte privado al sugerir la opción de compartir automóvil.

Se han encontrado diversos trabajos que han abordado el tema de patrones de movilidad utilizando diferentes maneras, como es el uso de servicios de llamadas o la ubicación de usuarios con el uso de dispositivos móviles, pero no han considerado que se utilice la movilidad de los usuarios de sus horarios y rutas similares para compartir el uso de automóvil. Por lo tanto, surge la siguiente interrogante, ¿cómo implementar un sistema telemático que proporcione sugerencias para compartir el medio de transporte en forma no supervisada a partir de patrones de movilidad de los usuarios?

## **1.2 Propuesta de solución**

En este proyecto terminal se propone el desarrollo de una alternativa al problema de movilidad en grandes ciudades con el propósito de reducir el tránsito vehicular. El sistema propuesto considera tres módulos: obtención de datos, soluciones ciudadanas e indicadores de gobierno. En la figura 1 se muestran los módulos principales del sistema telemático propuesto como parte de este proyecto terminal.

### **1.2.1 Módulo de Obtención de datos**

El propósito principal de este módulo es la obtención y almacenamiento de información de la movilidad de cada usuario mediante el registro de sus ubicaciones usando un dispositivo móvil.

#### **1.2.1.1 Ubicación del usuario**

Con el componente Ubicación del usuario se obtienen las coordenadas geográficas de los lugares en que se mueve cada usuario. Estos datos se detectan con un dispositivo móvil mediante su sensor de GPS (*Global Positioning System*). Los datos de latitud, longitud en que se ubica un usuario se muestrean y se almacenan junto con la fecha y hora de captura. Cuando se almacenen varios días de información en una base de datos en el dispositivo móvil; se enviarán a un servidor web, al confirmarse la recepción en el servidor, será borrada del dispositivo móvil.

El envío de información se determinará con base en el tipo de conexión a Internet que se tenga, de tal manera que el envío considere el ancho de banda disponible por la red en uso que tenga disponible el usuario.



**Figura 1.** Módulo principal de la propuesta

### **1.2.2 Módulo de Soluciones ciudadanas**

Este módulo del sistema es el encargado de sugerir a los usuarios conductores compartir sus medios de transporte particulares con usuarios peatones. A partir del uso del módulo de obtención de datos; se realiza el análisis de detección de patrones de movilidad individual de cada usuario para encontrar trayectorias de rutas similares en tiempo-espacio entre usuario conductores y peatones que usen la aplicación móvil para el uso compartido del vehículo privado. Se debe aclarar que no se utilizarán datos relacionados al tráfico ni tampoco en la sugerencia de ruta más eficiente. En este trabajo se propondrá desarrollar un algoritmo para sugerir viajes compartidos.

A partir del conocimiento de la metodología de cascada, se utilizará como base para realizar la propuesta de solución, partiendo de fases particulares y desplazándose en fases específicas.

### **1.2.3 Módulo de Indicadores de gobierno**

En este módulo se desarrollará un tablero de control (*dashboard*), que mostrará indicadores sobre movilidad de los usuarios que usan la aplicación visualizándose en mapas de calor, con el objetivo de que las entidades encargadas de dar soluciones, conozcan el comportamiento de dichos indicadores, entre los que se pueden mencionar: las trayectorias origen-destino de las rutas sugeridas para compartir el automóvil, el número de personas peatones y conductores, mapas de calor de las zonas más concurridas, número de sugerencias detectadas para compartir el automóvil, en base a esto, puede proponer soluciones para evitar la aglomeración de personas en áreas específicas.

## **1.3 Alcances**

El trabajo que se propone en este documento tiene los siguientes alcances:

* Se dispondrá de una aplicación móvil que podrá sugerir rutas de movilidad para los usuarios.
* Se implementará un esquema de seguridad en la aplicación móvil.
* El sistema soportará mínimo 100 usuarios.
* Se delimitará el rango de desplazamiento para ser considerado movilidad que necesite transporte.
* Se presentará la información para las entidades que toman decisiones en una ciudad en un tablero de control (*dashboard*) de tal manera que se corrobore su funcionalidad.

## **1.4 Objetivo general**

El objetivo general de este trabajo es desarrollar un sistema de solución a problemas de movilidad en ciudades inteligentes, que permita encontrar personas con destinos espacio-tiempo similares o iguales para sugerir compartir sus medios de transporte, además de proporcionar una visualización de KPIs de movilidad con un tablero de control.

### **1.4.1 Objetivos específicos**

El logro del objetivo se realiza al desarrollar los siguientes objetivos específicos:

1. Identificar requerimientos del sistema en sus diferentes módulos.
2. Desarrollar el módulo de adquisición de datos mediante el análisis de requerimientos, diseño, codificación y pruebas del módulo.
3. Desarrollar el módulo de soluciones ciudadanas mediante el análisis de requerimientos diseño, codificación y pruebas del módulo.
4. Desarrollar el módulo de indicadores de gobierno mediante el análisis de requerimientos, diseño, codificación y pruebas del módulo.
5. Integrar los módulos considerando las pruebas integrales.

**Capítulo 2. Estado del arte**

En este capítulo se presentan los trabajos relacionados a este proyecto, se organizan en tres secciones por el lugar en que se publicaron de acuerdo al interés de la institución: UPIITA, nacionales e internacionales.

## **2.1 UPIITA**

Los trabajos relacionados publicados en UPIITA se describen en esta sección.

El trabajo *Sistema de difusión de información comercial basado en geolocalización y perfil de usuario* [7], consiste en una aplicación de difusión de información relevante a clientes acerca de tiendas comerciales de su interés, a través de la consulta del perfil de cada usuario utilizando el sistema operativo Android. Mediante el uso de sistemas de información geográficos y bases de datos espaciales se pueden localizar los lugares de interés, delimitados en un rango de alcance para el usuario a partir de su localización con el uso del GPS de los dispositivos móviles.

En el trabajo *Extracción y clasificación de información del tráfico vehicular a través de Twitter y su visualización en Google Maps* [8], se desarrolla una herramienta para determinar las zonas con mayor tráfico vehicular en una ciudad mediante el análisis de información obtenida con la API de Twitter. En este caso, se analiza la información relacionada con el tráfico de la alcaldía Gustavo A. Madero, posteriormente con el uso de la API de Google Maps se accede a la base de datos de mapas para mostrar zonas con mayor conflicto vial. El resultado es información visual que pueda utilizar un usuario para que evite transitar por zonas con mayor aglomeración vehicular en un determinado momento.

En el proyecto terminal *#HELP!T7*[9], se elabora una aplicación móvil para informar al usuario acerca del tráfico vehicular en 38 colonias de la delegación Gustavo A. Madero, mediante el uso de la API de Twitter se obtuvo información relevante acerca de concentración vehicular, calles cerradas, manifestaciones, obras de vialidad, con el propósito de almacenarla en una base de datos geográfica. La información recopilada es analizada por el algoritmo de búsqueda de rutas *Shortest Path Shooting Star* para generar la ruta más corta o la ruta más rápida y para la visualización de los mapas se utilizó la herramienta *OpenStreetMap*, esta información se muestra al usuario en una aplicación móvil.

En el proyecto terminal *Aplicación de tráfico y navegación en la Ciudad de México basada en Minería de Textos*[10] se desarrolla una aplicación móvil que permite a los usuarios elegir rutas vehiculares alternas utilizando la información proporcionada por la API de Twitter acerca de información relevante de movilidad como son manifestaciones, choques automovilísticos, calles cerradas, bloqueos. Mediante el uso de algoritmos de minería de datos se consiguieron pronósticos de posibles eventos que puedan suceder en las vialidades, de tal manera se obtenga como resultado el cálculo de rutas posibles con menor tránsito para facilitar su movilidad para posteriormente ser visualizada en un mapa. Entre los algoritmos utilizados fueron: algoritmos de caminos mínimos, *Djisktra*, *Bellman* *Ford*, *Floyd* *Warshall* y para la minería de datos se utilizaron *ZeroR*, *OneR*, *Bayes*, Árbol de decisión.

En el proyecto terminal *Sistema de monitoreo, generación de itinerarios y análisis de datos de rutas de transporte público*[11] se integra el uso de la plataforma de cómputo en la nube *Amazon Web Services* y el servicio de bases de datos *DynamoDB* en la cual se buscó gestionar las rutas, estaciones y unidades de transporte de una empresa. El uso de sensores se utiliza en el rastreo de transporte, monitoreando la posición geográfica, velocidad y nivel de ocupación de los transportes. Esta información se muestra en mapas con la herramienta *OpenStreetMap* y se exporta a una aplicación móvil.

En la tabla 2 se presenta una comparación de los proyectos terminales elaborados en la UPIITA relacionados con esta propuesta indicada como “Este PT”.

**Tabla 2.** Comparación de proyectos terminales en la UPIITA

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Recurso utilizado** | **TRABAJO** | | | | | |
| **[7]** | **[8]** | **[9]** | **[10]** | **[11]** | **Este PT** |
| API de google | Si | Si | – | Si | – | Si |
| API de Twitter | – | Si | Si | Si | – | – |
| Algoritmos de aprendizaje máquina | – | Si | – | Si | – | – |
| Monitoreo de posición geográfica con GPS | – | – | – | – | Si | Si |
| Sugerencia de rutas de movilidad compartida | – | – | – | – | – | Si |

## **2.2 Nacionales**

En el trabajo *Smart cities, innovación y eficiencia urbanas: los nuevos modelos de transporte en México*[12] se discute el impacto y las futuras tendencias en el desarrollo de ciudades inteligentes en México; en este estudio el interés es la Zona Metropolitana de León, Guanajuato. En el año 2007 se establece el proyecto de los corredores confinados de transporte público, los cuales han sido adoptados en las zonas metropolitanas de México, Monterrey y León. Haciendo más eficiente la movilidad en el transporte público. A lo largo del escrito se hace mención a diversas ciudades inteligentes y los diversos proyectos que han realizado o que se continúan realizando. Además, se describen diversos modelos que puede utilizar una ciudad inteligente para llegar a ser sustentable, aclarando también que existen diversos elementos para diseñar un modelo específico para cada ciudad. En este trabajo realizó un análisis sobre la movilidad del transporte público *Optibus* en la ciudad de Guanajuato, describiendo características de las carreteras, rutas más concurridas y zonas de concentración de empleo, teniendo como objetivo incrementar la competitividad, la eficiencia, así como la captación de inversiones para convertirse en una ciudad sustentable.

## **2.3 Internacionales**

De acuerdo con un estudio realizado por las empresas *Juniper Research* e *Intel* en el año 2017 y 2018, Singapur [13] es considerada como la primera ciudad inteligente en un top de 20 ciudades, basándose en un modelo sobre ciudad inteligente (*smart city*) que se enfoca en cuatro áreas: movilidad (*mobility*), salud (*health*), seguridad (*safety*) y productividad (*productivity*). De acuerdo con el modelo realizado por el país de Singapur, se consideran como la segunda mejor ciudad inteligente y primer lugar en tener mejor movilidad de transporte público a nivel mundial [5]. En este país se han desarrollado varios tableros de visualización, uno de esos proyectos es nombrado *LIVE Singapore* [14], el cual proporciona a las personas acceso a la información en tiempo real sobre lo que sucede en su ciudad de forma visual [15], entre ellos podemos mencionar la visualización *Isochronic Singapore*, el cual proporciona de forma visual el tiempo que necesitan los ciudadanos para moverse de su origen a cualquier otro lugar dentro del país; y la visualización *raining Taxis* , la cual se encarga de dar información al usuario acerca de cómo se comporta el sistema de transporte de taxis cuando existen precipitaciones.

Copenhague, Dinamarca es una ciudad que cuenta con un tablero de control conteniendo información de sus usuarios por el uso de bicicletas [16], proveyendo información acerca de la temperatura ambiental, humedad, condiciones de las carreteras en tiempo real, niveles de contaminación y niveles de ruido. Esta información es compartida si así lo desea el usuario.

La ciudad de Londres cuenta con un tablero de control [17] compuesto por variables de densidad de tráfico, clima, nivel de contaminación, cantidad de autobuses y metros en servicio, porcentaje de personas infelices, disponibilidad de bicicletas, porcentaje de uso de estaciones del metro con y sin gente.

Nueva York, Estados Unidos de América, es una de las ciudades más turísticas, considerada la ciudad que nunca duerme, se ha desarrollado un tablero de control [18] para facilitar a los tomadores de decisiones el monitoreo de una gran variedad de indicadores (*KPI*) de la ciudad en tiempo real, como son: estadísticas de delitos, desempeño de las prestaciones de servicios, salud, obras públicas, indicadores ambientales, estadísticas de viviendas y personas sin hogar, entre otros.

En Barcelona, España se ha realizado un proyecto llamado *Sentilo* [19], este consta de una red de sensores que recopilan datos de la ciudad, permitiendo obtener la información en tiempo real de lo que sucede en el entorno de la ciudad y visualizando esa información para los ciudadanos.

El proyecto Comunicaciones 360° [20] desarrollado por el equipo científico de MOCA en España, ganador del premio GSMA GLOMO como el mejor servicio móvil en la nube en el año 2016 [21], utiliza la localización de los usuarios y el análisis *Big Data* para detectar patrones de movilidad de los usuarios y predecir el medio de transporte en el que se desplazan turistas y residentes.

El proyecto SIADE [22] es un sistema inteligente, intuitivo y visual que tiene como propósito optimizar la gestión de redes de transporte mediante el análisis masivo de datos como son los registros de datos geográficos de pasajeros y el uso de inteligencia artificial.

En el artículo *Mining Individual Mobility Patterns Based on Location History* [23] publicado en la revista IEEE Xplore, se ha desarrollado en la Universidad Nacional de Tecnología de Defensa Chang Sha de China, se llevó a cabo un proyecto de minería de datos a partir de los patrones de movilidad utilizando la información de trayectorias del GPS, censado con una frecuencia periódica, con el objetivo de saber los patrones de movilidad individual de los usuarios y el periodo en el cual frecuentan sus rutas. Entre los algoritmos utilizados se pueden mencionar: algoritmos de agrupación *CFS* para minar los puntos de referencias de los usuarios. Este algoritmo es una idea más simple, básica y novedosa en comparación a otros como son *K-means*.

En la tabla 3 se presenta una comparación entre los diversos proyectos y artículos internacionales que abordan temas similares a la propuesta de este proyecto.

**Tabla 3.** Proyectos y artículos internacionales que abordan temas similares

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Característica relevante de movilidad** | **TRABAJO** | | | |
| **[20]** | **[22]** | **[23]** | **PT** |
| Detección de patrones de movilidad | Si | Si | Si | Si |
| Detección del medio de transporte utilizado por el usuario | Si | – | – | – |
| Uso de un dispositivo móvil para detectar la ubicación | Si | Si | Si | Si |
| Ofrecer servicio de movilidad con base a patrones de movilidad | – | – | – | Si |

## **2.4 Aplicaciones comerciales**

Existen varios productos comerciales que se han desarrollado para resolver problemas de movilidad, en esta sección se exponen algunos de los más conocidos.

*Waze* [24] es una aplicación creada en el año 2008 en Israel con el propósito de informar acerca del estado del flujo vial, incidentes en la ciudad y carreteras, los cuales son proporcionados por la comunidad de usuario que utiliza la aplicación. Además, utiliza la geolocalización del usuario para conocer el estado actual del tráfico, proporcionar información básica sobre establecimientos y gasolineras diferenciadas por marca/tipo/distancia, cuenta con una versión para compartir automóvil, enviar la ubicación del usuario en tiempo real y cuenta con una barra indicadora de atasco.

*Google Maps* [25] es un servidor de aplicaciones de mapas que pertenece a Alphabet Inc. el cual proporciona servicio de mapas desplazables, fotografías satelitales, con el objetivo de permitir dar la localización de una dirección en el mapa, además de contar con una gran base de datos con información de establecimientos de las zonas, cuenta con indicador de carril, rutas para vehículo, bicicleta, caminar, transporte público, opción de ruta sin peaje, medir la densidad del tráfico.

*Uber Technologies Inc.* [26] es una empresa internacional de transporte de vehículo, a través del uso de una aplicación móvil, donde el usuario podrá contactar a los conductores que estén cerca de su punto de origen para solicitar el servicio, además esta aplicación permite solicitar el servicio de transporte compartido.

*Blablacar* [27] es un servicio de vehículo compartido que tiene como objetivo el desplazamiento de los usuarios que tengan el mismo destino en horarios iguales o casi similares.

En la tabla 4 se presenta una comparativa entre los servicios de *Waze*, *Google Maps,* *Uber*, *Blablacar* y el Sistema de reconocimiento de patrones de movilidad para ciudades inteligentes propuesto en este trabajo terminal.

**Tabla 4.** Comparación de diversas aplicaciones de servicios de movilidad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Característica relevante de movilidad** | **TRABAJO** | | | | |
| **[24]** | **[25]** | **[26]** | **[27]** | **PT** |
| Sugerencia de rutas considerando la comunidad de Twitter | Si | – | – | – | – |
| Uso de la API de Google | – | Si | – | Si | Si |
| Aplicación para compartir automóvil | – | – | Si | Si | Si |
| Sugiere compartir automóvil utilizando los patrones de movilidad de un grupo de usuarios | – | – | – | – | Si |

## **2.5 Discusión del estado del arte**

En el estudio del estado del arte, hasta ahora se pueden observar varias alternativas de solución a los problemas de movilidad. En el caso de UPIITA se utilizan diferentes herramientas como la API de Twitter para identificar el nivel actual del tráfico vehicular en la ciudad de México, algoritmos de minería de datos o algoritmos de trazado de rutas para obtener la ruta más corta o rápida posible. Se aprovechan los dispositivos móviles para obtener la ubicación de usuario y compartir el automóvil generando una ganancia monetaria.

Se observa que, ninguno de los trabajos implementa sugerencias de compartir el automóvil con base en patrones de movilidad de los ciudadanos para sugerir compartir el vehículo entre los usuarios que estén usando la misma aplicación del sistema, siendo esta, la propuesta de solución para elaborarse en este trabajo terminal. Esta propuesta de tener éxito, es posible escalarla a otras ciudades.

# **Capítulo 3. Marco teórico**

## **3.1 Aprendizaje automatico (*Machine Learning*)**

El aprendizaje automático [28] es una rama de la inteligencia Artificial que se encarga de la construcción de programas informáticos para extraer reglas o reconocimiento de patrones de comportamiento, así como adaptarse a nuevos cambios que se presenten en conjuntos de datos. Las técnicas de aprendizaje automático se dividen en dos grupos, el aprendizaje supervisado y el aprendizaje no supervisado.

### **3.1.1 Aprendizaje Supervisado**

El aprendizaje supervisado [29] es el proceso de enseñar un modelo al alimentarlo con datos de entrada y salida conocidos o correctos (datos etiquetados o datos de entrenamiento etiquetados), con el motivo de ayudar a predecir un resultado correcto [00]. Algunos algoritmos de aprendizaje supervisado son: clasificación, regresión, bosques al azar (*random forests*), entre otros.

### **3.1.2 Aprendizaje No supervisado**

El aprendizaje no supervisado [29] es el proceso de enseñar un modelo al alimentarlo con datos de entrada y salida que no están clasificados ni etiquetados con el objetivo de que el algoritmo agrupe información para encontrar patrones desconocidos en el con conjunto de datos. Algunos algoritmos de aprendizaje no supervisado son:

**Agrupación o agrupamiento o segmentación (*Clustering*).** Es una técnica de aprendizaje automático que implica la agrupación de objetos descritos que comparten una o varias características similares. Un ejemplo es agrupar clientes por comportamiento de compra.

Existen diferentes métodos de clusterización, como son: distribución, particiones o centroides, jerárquicos, densidad, categorización, entre otros.

El método de particionamiento tiene dos parámetros de entrada, k que representa el número de grupos (*clusters*) y n el número de iteraciones, donde k ≤ n.

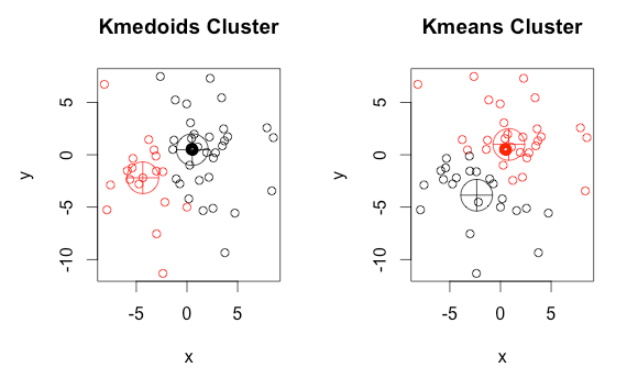
* ***K-means*.** Es un algoritmo de agrupamiento por particiones, donde son seleccionados de manera aleatoria k grupos de centroides. Se calcula en cada iteración la distancia, usualmente la distancia Euclidiana entre cada punto de datos y k centroides más cercanos. Se asigna cada punto de datos al grupo centroide más cercano utilizando la suma de las distancias al cuadrado. Se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo. Se repite este procedimiento hasta que los centroides ya no se mueven. Obsérvese como se realiza una agrupación al utilizar el algoritmo de *k-means*.

|  |  |
| --- | --- |
| **Antes de K-Means** | **Después de K-Means** |
|  | |

**Figura 2.** Antes y después de utilizar K-means

Ejemplos de casos de uso son:

* + Segmentación de clientes basado en historial de compras.
  + Detección de fraude de seguros o transacciones.
  + Detección de dispositivos IoT no autorizados basados en el tráfico de la red.
* ***K-Medoids***. Es un algoritmo de agrupamiento por particiones, donde son seleccionados de manera aleatoria k grupos de centroides llamados medoids (que son puntos que pertenecen al conjunto de datos). A diferencia de *K-Means*, que calcula la distancia al cuadrado, este algoritmo calcula la distancia absoluta entre los puntos y el centroide seleccionado, como resultado, es más robusto al ruido y valores atípicos (muy grandes o pequeños) que K-means.



**Figura 3.** Diferencia de centroides entre K-medoids y K-means

En conclusión, existe una gran variedad de algoritmos de aprendizaje no supervisado, los cuales son diferentes a los requerimientos del proyecto, por lo tanto, se desarrolla un algoritmo en específico.

## **3.2 Sistema de Información Geográfica (SIG)**

SIG o GIS (*Geographic Information System*) [30] es un conjunto de componentes específicos que permiten a los usuarios finales crear consultas, integrar, analizar, representar y ayudar a mejorar la toma de decisiones de una manera fácil, clara y eficiente de cualquier tipo de información geográfica asociada a un informe, gráfico, territorio o mapa.

Los SIG, requieren elementos que les permitan mostrar de manera más clara la representación de la información de sus mapas, siendo estos los siguientes [31]: posición, forma, tamaño, color, textura y orientación.

Existen diversas aplicaciones para utilizar en un proyecto un SIG, como son: planificación urbana, gestión de recursos naturales, geografía histórica, gestión de servicios, rutas de transporte, cartografía, planificación comercial, evaluación de riesgos y emergencias, impacto ambiental, estudios demográficos, entre otros.

## **3.3 Sistema de coordenadas geográficas**

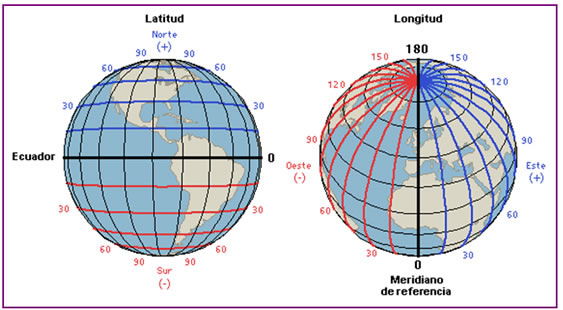
Un sistema de coordenadas geográficas o terrestres es un conjunto de valores que permiten referenciar cualquier punto de la superficie terrestre. Se denomina sistema de referencia [32] al conjunto de ejes, puntos o planos que coinciden en un punto origen y a partir del cual se calculan las coordenadas de cualquier otro punto. En el planeta Tierra, estos ejes están basados en líneas imaginarias llamadas paralelos y meridianos.

Existen diferentes tipos de sistemas para representar un espacio geométrico, como son: sistema de coordenadas cartesianas, sistema de coordenadas polares, sistema de coordenadas cilíndricas, sistema de coordenadas esféricas, entre otros. Sin embargo, para representar la superficie terrestre, se utiliza el sistema de coordenadas angulares utilizando latitud y longitud para cada punto; las unidades utilizadas en estas medidas son los grados.

La latitud proporciona la posición de un lugar, en dirección Norte o Sur entre 0° y 90°, teniendo al Ecuador como origen para ambas latitudes (norte o sur), este lugar tiene la latitud de 0°. Las coordenadas que se encuentren al Sur del Ecuador, reciben la denominación Sur (S), con signo negativo; y aquellas que se encuentran hacia al Norte del Ecuador, reciben la denominación Norte (N), con signo positivo. La latitud es siempre menor o igual a 90º. Los polos tienen latitud 90º y -90º respectivamente. La latitud se indica en líneas paralelas que se trazan desde el Ecuador hacia el Norte y hacia el Sur.

La longitud proporciona la localización de un lugar en dirección Este u Oeste a partir de un meridiano de referencia conocido como meridiano de Greenwich que tiene el valor de 0º (referido como la longitud 0°). Las coordenadas que se encuentran en un mismo meridiano tienen la misma longitud desde el norte hasta el sur. Aquellos puntos geográficos que se encuentren al Oriente del Meridiano de Greenwich, reciben la denominación Este (E) variando entre 0° y 180°. Los puntos geográficos ubicados al Occidente del Meridiano de Greenwich reciben la denominación Oeste (W) variando entre 0° y 180°.

En la figura 4 [32] se muestra una representación en dos superficies terrestres de la latitud, longitud, el Ecuador y el Meridiano de Greenwich.



**Figura 4.** Representación de latitudes, longitudes, el Ecuador y el Meridiano de Greenwich

## ***3.*4Fórmula de Haversine**

En este trabajo se requiere calcular la distancia entre dos puntos geográficos, para esto se utiliza la fórmula de Haversine [33], esta ecuación se utilizada en la navegación astronómica para calcular la distancia lineal entre dos puntos geográficos sobre la esfera terrestre. Está fórmula requiere tener el valor del radio de la Tierra, pero debido a la forma variable de nuestro planeta, actualmente se consideran tres valores diferentes para establecer el radio terrestre, esto se debe a que la Tierra no es perfectamente redonda sino geoide. Los radios utilizados actualmente son:

* Radio ecuatorial: 6378 km
* Radio polar: 6357 km
* Radio equivolumen: 6371 km

La ecuación utilizada en el cálculo de la distancia de Haversine entre dos puntos geográficos cada uno representado por su latitud y longitud, está dada por la expresión (1).

donde:

(1)

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Descripción** | **Unidades** |
| *r* | radio equivolumen de la Tierra | kilómetros |
|  | latitud de un punto 2 | radianes |
|  | latitud de un punto 1 | radianes |
|  | longitud de un punto 2 | radianes |
|  | longitud de un punto 1 | radianes |
| *d* | Distancia entre los puntos 1 y 2 | kilómetros |

## **3.5 Colaboración masiva (*Crowdsourcing*)**

Existen diversas maneras de obtener información de múltiples usuarios, entre ellas, la que se obtiene de la Web, como el caso de grupos de personas alrededor del mundo quienes participan de manera voluntaria y colaborativa en redes sociales o en foros, en servicios, para mejorar o desarrollar una tecnología. La colaboración de personas de diferentes lugares del mundo ha permitido dar solución de algún problema o aumentar el conocimiento de algún tema. Ejemplos de estos logros son: Wikipedia, GNU/Linux, Stack Overflow, hackatones, entre otros [34].

## **3.6 Sensado masivo de dispositivos móviles (*Mobile* *Crowdsensing*)**

Actualmente, con el desarrollo de los dispositivos móviles inteligentes, redes inalámbricas y la computación en la nube en conjunto con la participación humana, ha sido posible el desarrollo de una técnica que permite recolectar la información de los dispositivos móviles de un grupo de usuarios conocida como sensado masivo (*mobile crowdsensing*) [35].

La información se obtiene de diversos sensores que están incorporados en los dispositivos móviles. Estos con sus capacidades de procesar, almacenar y enviar información pueden realizar cálculos y presentarle al usuario información que guie sus actividades. Algunos ejemplos de estos dispositivos son: teléfonos inteligentes (*smartphones*), tabletas (*tablets*), relojes inteligentes (*smartwatches*) y dispositivos vestibles (*wearables*).

Algunos ejemplos de la información que puede recolectar un dispositivo móvil son: localización geográfica del usuario a través del sensor de GPS, nivel de ruido utilizando el micrófono, temperatura ambiental, presión atmosférica, velocidad de desplazamiento o movimiento del dispositivo a través del sensor de acelerómetro, entre otros.

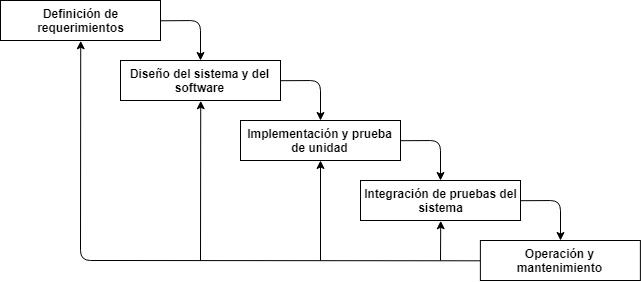
## **3.7 Tablero de Control (*Dashboard*) Gobierno de datos**

Las personas encargadas de tomar decisiones que afectan a un grupo social, se apoyan en información estratégica que actualmente se coloca en un tablero de control, el cual permite visualizar indicadores claves de rendimiento (KPIs), con el objetivo de organizar la información que consideren relevante acerca de tareas, proyectos, gestión de costos y otros aspectos críticos que afecten a la población o sea necesaria para lograr sus objetivos [36].

## **3.8 Metodología de Cascada (*Waterfall*)**

También llamada como ciclo de vida de un programa o secuencial, es una metodología que avanza por etapa, no se puede continuar a la siguiente etapa hasta que finalice la etapa anterior. Al final de cada etapa, se debe realizar una revisión final para determinar si el proyecto debe avanzar a la siguiente fase [37].

A continuación, en la figura 5, se muestra el modelo de la metodología de cascada.



**Figura 5.** Metodología de cascada

Las principales etapas que deben reflejar las actividades del desarrollo de un proyecto son las siguientes:

1. **Análisis y definición de requerimientos.** Se describen las restricciones y metas que debe cumplir el sistema solicitado por el usuario. Posteriormente, se detallan los requerimientos del sistema.
2. **Diseño del sistema y del software.** Se asignan los requerimientos para el hardware o software del sistema, así como establecer la arquitectura del sistema.
3. **Implementación y prueba de unidad.** En esta etapa, se realizarán los programas que servirán para verificar que cada unidad cumpla con su especificación de requerimientos.
4. **Integración y prueba del sistema**. Se integran los programas individuales para obtener un sistema completo que cumple con los requerimientos del software. Después de probarlo, se libera el sistema de software al usuario o cliente.
5. **Operación y mantenimiento.** Corrección de los errores que no se detectaron en etapas anteriores del ciclo de vida del software, así como mejorar la implementación del sistema conforme se descubren nuevos requerimientos.

Esta metodología es utilizada para estructurar el escrito de este proyecto terminal y para realizar la base de datos de este sistema.

## **3.9 Diagramas de diseño orientado a objetos**

En esta sección se presentan los principales diagramas y componentes para diseñar diagramas orientado a objetos, basados en la notación UML v2.5, los cuales sirven para entender la notación utilizada en este trabajo. Obsérvese en la tabla 5.

**Tabla 5.** Componentes de un diagrama de caso de uso

|  |  |  |
| --- | --- | --- |
| **Componente** | **Representación** | **Descripción** |
| Actor |  | Es una entidad que cuenta con un rol específico para interactuar en el sistema. |
| Funcionalidad |  | Acción que debe ser realizada por uno o varios actores |
| Relación |  | Símbolo utilizado para relacionar uno o varios actores con una o varias funcionalidades del sistema |
| Sistema |  | Módulo donde interactúan actores, relaciones y funcionalidades |

### **3.9.1 Caso de uso**

En este trabajo se entiende un caso de uso como una representación de los actores que interactúan con el sistema y las principales funciones que transforman los datos. Se representa mediante el uso de un diagrama y su descripción. Un diagrama está compuesto por actores, funcionalidades del sistema y relaciones entre los actores y las funcionalidades del sistema. En la tabla 6, se muestran los componentes de un diagrama de casos de uso. En la tabla 7, se definen los elementos de la descripción de un diagrama de casos de uso [39].

### **3.9.2 Elementos de un diagrama de clases**

En este trabajo se mencionan las definiciones necesarias para entender los **elementos de una clase**, así como los diferentes **tipos de clases** que pueden existir al modelar un sistema de información.

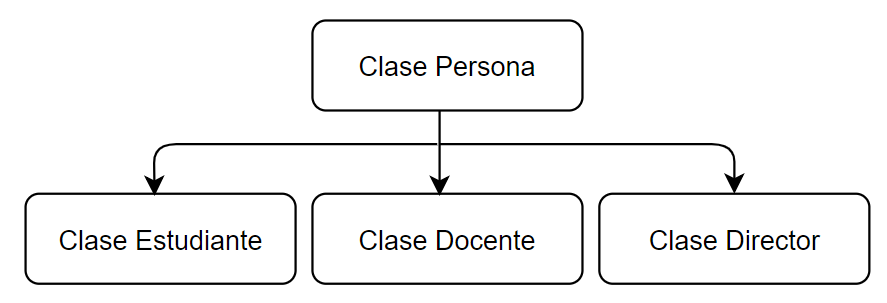
* **Clase:** se entiende que, una clase es una descripción abstracta de una entidad de la realidad en términos de propiedades (datos) y métodos (procesos) encapsulados en una unidad [40].
  + **Propiedad (atributos)**: Es un dato que representa un valor o un rango de valores primitivos (int, short, byte, long, char, double, float) puros en una clase.
  + **Método:** Es un proceso que contiene un conjunto finito de instrucciones que puede ser aplicado por o en los objetos de una clase. Un método termina con paréntesis, donde puede o no incluirse parámetros [40].
* **Objeto:** Una clase puede tener uno o varios objetos. Se entiende que un objeto es una variable que puede ser utilizada dentro de una clase para emplear sus atributos o métodos.
* **Constructor:** Es un método especial de una clase que es llamado automáticamente siempre que es creado un objeto en esa clase. Tiene la función de inicializar el objeto creado [41].
* **Sobrecarga:** Declarar en una clase múltiples constructores o métodos con el mismo nombre y diferentes parámetros con el propósito de inicializar un objeto de una clase de diferentes maneras.

**Tabla 6.** Componentes de la descripción de un caso de uso

|  |  |
| --- | --- |
| **Nombre** | Identificación del caso de uso |
| **Actores** | Tipo de usuarios que interactúan en el sistema |
| **Versión** | Asignación de un código de identificación para el nivel de desarrollo de un proyecto. Ejemplo: 1.0 (abril 2019) |
| **Descripción** | Se detalla la interacción entre los usuarios y el sistema del caso de uso |
| **Disparador** | Evento que provoca que el actor deba inicializar el caso de uso |
| **Precondiciones** | Situaciones que suceden antes de comenzar el caso de uso |
| **Postcondiciones** | Situación que sucede después de que fue realizado de forma normal y exitosa el caso de uso |
| **Flujo Normal** | Explicación de los pasos a seguir del caso de uso |
| **Flujos alternativos** | Explicación de los pasos del caso de uso que suceden en dado caso que no se cumpla un paso del flujo normal |
| **Excepciones** | Descripción de los pasos del caso de uso que suceden en dado caso que se produzca un error o una situación no descrita en el flujo normal |
| **Prioridad** | Importancia que tiene el caso de uso para ser usado |
| **Frecuencia de uso** | Cuántas veces es usado el caso de uso por el usuario o sistema |
| **Reglas de negocio** | Requerimientos del cliente que deben seguirse en el caso de uso |
| **Requerimientos especiales** | Requerimientos que debe tener el caso de uso |
| **Suposiciones** | Asumir que una o varias acciones son cumplidas por el actor o el sistema |

En un diagrama de clases existen diferentes **relaciones entre clases** que permiten la reutilización de código, siendo estas las siguientes:

* **Herencia:** Es la propiedad de una subclase de obtener los mismos atributos y métodos de la superclase. A continuación, en la figura 6, se muestra un ejemplo de una herencia.



**Figura 6.** Ejemplo de herencia

* 1. **Generalización:** Se realiza a partir de un conjunto de clases y se busca una clase que las agrupe [40].
  2. **Especialización:** Se genera a partir de que una clase genérica obtiene clases que son casos particulares de la misma [40].
  3. **Polimorfismo:** Facultad de asumir muchas formas, es decir, los métodos deben ser definidos en la superclase y redefinidos en las subclases, los objetos deben ser manipulados mediante referencias a la superclase [41].
* **Composición:** Se refiere al termino de incluir una clase dentro de otra(s) con el propósito de crear clases más complejas [40].
* **Agregación:** Es una relación que define cuando una clase es parte de otra clase más compleja o mediante la agrupación de una colección de la clase existente [40].
* **Asociación:** Es un vínculo entre clases, que indica la existencia de una relación entre dichas clases [37]. Para indicar el tipo de relación en una asociación, se debe definir la cardinalidad de las relaciones. Una cardinalidad indica el número de entidades que puede estar relacionada con una entidad. Por ejemplo:
  1. 1:n - uno a varios
  2. 1:1 - uno a uno
  3. n:n - varios a varios

### **3.9.3 Elementos de un diagrama de secuencia**

En este trabajo se entiende que, un diagrama de secuencia como la modelación de las interacciones entre los actores y los objetos del sistema, así como las interacciones entre los objetos en sí [37].

A continuación, en la tabla 7 se describe cada uno de los componentes que tiene un diagrama de secuencia:

**Tabla 7.** Símbolos de un diagrama de secuencia

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Símbolo** | **Descripción** |
| Actor |  | Entidades que interactúan con el sistema o que son externas al sistema |
| Clase |  | Describe a la entidad que interactúa en el sistema |
| Objeto |  | Variable de una clase |
| Línea de vida de un objeto u clase |  | Representación del paso del tiempo en forma de línea vertical, la cual crece hacia abajo. |
| Mensaje |  | Flecha que representa la comunicación entre los objetos del sistema. |
| Casilla de activación |  | El rectángulo vertical representa el tiempo en que un objeto necesita para completar su tarea. |

### **3.9.4 Elementos de un diagrama de flujo**

Una forma de entender cómo funciona un algoritmo, es mediante la representación de un diagrama de flujos.

Un diagrama de flujo es la representación gráfica de una secuencia de actividades de un proceso a desarrollarse, facilitando su entendimiento y su funcionamiento correcto. En la tabla 8, se muestran los símbolos que son utilizados para representar un diagrama de flujos [42].

**Tabla 8.** Símbolos para elaborar un diagrama de flujo

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Nombre** | **Descripción** |
|  | Inicio/Final | Indica el Inicio o comienzo del flujo de un diagrama |
|  | Acción o Proceso | Indica las funciones y actividades que debe realizar un determinado proceso |
|  | Decisión o alternativa | Indica la toma de una decisión entre dos o más opciones |
|  | Entrada/Salida | Representa la información que entra o sale del sistema |

# **Capítulo 4. Análisis**

En este capítulo se describe el análisis que requiere identificar los requerimientos funcionales, no funcionales y técnicos; estos son analizados para determinar formas apropiadas de diseño e implementación de esta propuesta.

* Los requerimientos funcionales son las exigencias que debe tener un software, sistema o aplicación para funcionar de la manera correcta y deseada.
* Los requerimientos no funcionales son características y limitaciones que debe tener el software, como el rendimiento, la fiabilidad, la escalabilidad, facilidad de uso, seguridad, entre otros.
* Los requerimientos técnicos permiten identificar las herramientas que se utilizarán para realizar dicho sistema, como es la elección del sistema operativo, lenguaje de programación, entorno de desarrollo, entre otros.

En la segunda parte del capítulo se especifica el análisis que se debe llevar a cabo para satisfacer los requerimientos detectados. En la última parte se exponen las herramientas propuestas para ser utilizadas en este sistema.

## **4.1 Detección de Requerimientos.**

Esta sección se ha estructurado basándose en las especificaciones del estándar IEEE Práctica Recomendada para Especificaciones de Requerimientos de Software ANSI/IEEE 830, 1998.

El propósito de obtener los requerimientos es contar con los elementos esperados del sistema propuesto. Los requerimientos se analizan para determinar la mejor alternativa de su atención.

El sistema tendrá como producto una aplicación en un dispositivo móvil para el usuario y un tablero de control para los tomadores de decisiones. En la segunda parte se describen los requerimientos, restricciones y alcances de los módulos que componen al sistema siendo los siguientes: obtención de datos, soluciones ciudadanas e indicadores de gobierno. Además, se realiza un esquema para describir cómo es la interacción de los componentes de cada módulo.

### **4.1.1 Requerimientos del módulo de obtención de datos**

En esta sección se describen los requerimientos del módulo de obtención de datos, el cual está conformado por una aplicación móvil para leer las ubicaciones de los usuarios que se almacenarán en un servidor Web. En la figura 7, se muestran el contexto del módulo de obtención de datos.

|  |
| --- |
|  |

**Figura 7.** Componentes del módulo de obtención de datos

#### **4.1.1.1 Requerimientos Funcionales de la aplicación del usuario**

La elaboración de la aplicación móvil utiliza los requerimientos funcionales siguientes:

**RF1.1:** Al ingresar al sistema, un usuario debe registrarse, de tal manera que se genere su perfil de usuario para que se almacene en la base de datos ubicada en el servidor Web. Es requisito que el usuario haya recibido un código único como invitación. Como resultado el usuario obtiene una clave de usuario y su contraseña.

**RF1.2:** Un usuario registrado debe iniciar sesión para utilizar la aplicación.

**RF1.3:** Cuando el usuario ingresa por primera vez en la aplicación**,** se requiere su autorización para que la aplicación móvil tenga acceso a los sensores de ubicación y el acceso a Internet (este puede ser mediante *Wi-Fi* o GPRS (por sus siglas en inglés *General Packet Radio Service*)).

**RF1.4:** La aplicación tendrá una opción para que el usuario actualice la información de su perfil de usuario que se encuentra en el servidor web.

**RF1.5:** Se requiere que la aplicación utilice un gestor de base de datos para almacenar internamente las ubicaciones por donde se mueve el usuario. Estos datos se enviarán al servidor únicamente cuando se tenga conexión *Wi-Fi*, mismos que serán eliminados del dispositivo móvil.

**RF1.6:** El usuario propietario del dispositivo móvil podrá activar el sensor de ubicación en el momento que lo solicite la aplicación. Si el usuario desea evitar el rastreo de sus rutas privadas, debe desactivar el proceso de muestreo.

**RF1.7:** La aplicación debe recopilar las ubicaciones de su usuario mediante un servicio ejecutándose en segundo plano.

**RF1.8:** El sistema estimará si se está usando algún tipo de transporte, el usuario camina o se encuentra detenido. Esta información se usará cuando se procesen las sugerencias de transportación.

**RF1.9:** El intercambio de datos entre la aplicación y el servidor web será con un cifrado de datos. El servidor tendrá activado un Firewall.

#### **4.1.1.2 Requerimientos No Funcionales de la aplicación del usuario**

En esta subsección se especifican los requerimientos no funcionales de la aplicación del usuario.

**RNF1.1:** El usuario deberá instalar la aplicación en un dispositivo móvil.

**RNF1.2:** Los usuarios de UPIITA que tengan derecho al uso de la aplicación se les proporcionarán una clave de uso único para que la instalen. Al momento de instalar se usará la dirección MAC del dispositivo para que en caso de extravío se deshabilite el acceso al sistema bloqueando la dirección MAC.

**RNF1.3:** Se realizará un estudio de factibilidad para conocer la proporción de población de la UPIITA que acepta compartir su automóvil para trasladarse juntos a la escuela o sus destinos al salir de clase.

#### **4.1.1.3 Requerimientos Técnicos de la aplicación del usuario**

En esta subsección se especifican los requerimientos técnicos de la aplicación del usuario.

**RT1.1:** El dispositivo móvil debe tener un procesador de 800 MHz o superior, 1 Gb de memoria RAM mínimo, al menos 8 Gb disponibles de almacenamiento persistente, disponer de un sensor de ubicación GPS, así como acceso a Internet por medio de *Wi-Fi* o GPRS. El dispositivo debe tener el sistema operativo Android versión 6 o superior.

**RT1.2:** Se deben realizar pruebas de funcionamiento de las características de Android que se utilizarán en el sistema.

**RT1.3:** Se requiere un servidorWeb que soporte bases de datos y un lenguaje de programación.

#### **4.1.1.4 Requerimientos Funcionales de la base de datos**

En esta subsección se especifican los requerimientos funcionales de la base de datos de la aplicación móvil.

**RF2.1:** El sistema de base de datos debe almacenar los datos que se envían desde la aplicación móvil.

**RF2.2:** Se requiere que los gestores de las bases de datos cuenten con las funcionalidades de: crear tablas, leer, actualizar y borrar datos almacenados.

#### **4.1.1.5 Requerimientos No Funcionales de la base de datos**

En esta subsección se especifican los requerimientos no funcionales de la base de datos de la aplicación móvil.

**RN2.1:** Se requiere estimar el volumen de datos que se almacenará en el sistema de gestión de base de datos en cada una de sus tablas.

#### **4.1.1.6 Requerimientos Técnicos de la base de datos**

En esta subsección se especifican los requerimientos técnicos de la base de datos de la aplicación móvil.

**RT2.1:** Se requiere contar con un sistema gestor de base de datos relacional que almacenen los datos de la ubicación de los usuarios.

### **4.1.2 Requerimientos del módulo de soluciones ciudadanas**

En esta sección se describen los requerimientos del módulo de soluciones ciudadanas, el cual estará conformado por la base de datos de ubicación de usuario generada en el módulo de ubicación de usuario para ser procesado en el clúster, de manera que se generen soluciones relacionadas a la movilidad de los usuarios de la aplicación móvil. En la figura 8, se presentan los componentes del módulo de soluciones ciudadanas.

|  |
| --- |
|  |

**Figura 8.** Componentes del módulo de soluciones ciudadanas

#### **4.1.2.1 Requerimientos Funcionales del clúster**

En esta subsección se especifican los requerimientos funcionales del clúster.

**RF3.1:** Únicamente el usuario administrador, podrá tener acceso a la plataforma del clúster.

**RF3.2:** El usuario administrador debe iniciar sesión para utilizar la plataforma.

**RF3.3:** El sistema analizará los patrones de movilidad individual de los usuarios para estimar el tipo de movilidad de los usuarios.

**RF3.4:** El sistema calculará quienes son los usuarios candidatos para compartir medio de transporte.

**RF3.5:** El sistema almacenará en el servidor Web quienes son los usuarios candidatos a compartir sus medios de transporte.

#### **4.1.2.2 Requerimientos No Funcionales del clúster**

En esta subsección se especifican los requerimientos no funcionales del clúster.

**RNF3.1:** El acceso de la información del clúster estará disponible para el administrador cuando el equipo de computación este encendido.

**RNF3.2:** El sistema de información continuará brindando su servicio sin ser afectado por posible crecimiento de la base de datos.

**RNF3.3:** El sistema procesará la información en el menor tiempo posible.

#### **4.1.2.3 Requerimientos Técnicos del Clúster**

En esta subsección se especifican los requerimientos no funcionales del clúster.

**RT3.1:** La plataforma de procesamiento distribuido, debe contar con computadoras ordinarias que cumplan las siguientes características mínimas siguientes:

- Procesador i7.

- 8 GB de memoria RAM (recomendable 16GB).

- 1 Tb en disco duro mínimo.

- Tarjeta de red de 1 Gbps.

- 1 mouse.

- 1 teclado.

- 1 monitor.

**RT3.2:** Se dará preferencia al uso de software del sistema de licencia libre (*open source)*.

### **4.1.3 Requerimientos del módulo de indicadores de gobierno**

En esta sección se describen los requerimientos del módulo de indicadores de gobierno, el cual estará conformado por la base de datos de ubicación de los usuarios y soluciones ciudadanas para ser procesadas en el clúster, de tal manera que se generen los indicadores relacionados a la movilidad de los usuarios de la aplicación móvil en un tablero de control. En la figura 9, se presenta los componentes del módulo de soluciones ciudadanas.

|  |
| --- |
| **Tablero de control** |

**Figura 9.** Componentes del módulo de indicadores de gobierno

#### **4.1.3.1 Requerimientos Funcionales del tablero de control**

En esta subsección se especifican los requerimientos funcionales del tablero de control.

**RF4.1:** En el tablero de control se visualizará un mapa de calor de las zonas con mayor concurrencia de la población que autorizó su recopilación de datos.

**RF4.2:** En el tablero de control se visualizarán las trayectorias origen-destino de los usuarios establecidas en un intervalo de tiempo indicadas por el usuario administrador.

**RF4.3:** En el tablero de control se visualizará el tipo de movilidad individual del usuario.

**RF4.4:** En el tablero de control se visualizarán las rutas de usuarios que son candidatos a compartir transporte.

#### **4.1.3.2 Requerimientos No Funcionales del tablero de control**

En esta subsección se especifican los requerimientos no funcionales del tablero de control.

**RNF4.1:** La interfaz gráfica tendrá letreros de tamaño adecuado para su visualización y colores contrastados.

#### **4.1.3.3 Requerimientos Técnicos del tablero de control**

En esta subsección se especifican los requerimientos técnicos del tablero de control.

**RT4.1:** El tablero de control, debe contar con una computadora ordinaria que cumplan las siguientes características mínimas siguientes:

- Procesador Pentium III.

- 4 GB de memoria RAM.

- 40 GB en disco duro mínimo.

- Tarjeta de red.

- 1 mouse

- 1 teclado

- 1 monitor

**RT4.2:** Se dará preferencia al uso de software del sistema de licencia libre (*open source)*.

## **4.2 Análisis de requerimientos**

En esta sección, se presenta el análisis de requerimientos de la sección 4.1:

### **4.2.1 Análisis de Requerimientos del módulo de obtención de datos**

#### **4.2.1.1 Análisis de Requerimientos Funcionales de la aplicación del usuario**

**ARF1.1:** El usuario requiere registrarse para generar su perfil de usuario en el sistema, la información requerida es la siguiente: nombre completo, correo electrónico, teléfono celular, sexo, disponibilidad de automóvil, en caso de contar con automóvil indicar si acepta compartirlo, si aplica preferencia de pasajeros con quien comparte (hombre o mujer), fecha de nacimiento, contraseña, código de acceso único.

Además de los datos mencionados, se consideran otras variables que deben ser almacenadas en la tabla del perfil de usuario para mapear las trayectorias de los usuarios, las cuales serán obtenidas de la información del dispositivo móvil, siendo las siguientes: ID de usuario, día/mes/año, hora/minuto/segundo, celular robado.

**ARF1.2:** Después de que el usuario sea registrado en el sistema, se requiere que ingrese su correo electrónico y contraseña para acceder a la aplicación.

**ARF1.3:** El usuario debe aceptar los permisos solicitados por la aplicación al ingresar por primera vez sesión.

**ARF1.4:** Se desarrollará una sección en la aplicación para actualizar la información del perfil de usuario, en dado caso que este desee modificarlo, únicamente será posible esta función cuando el dispositivo tenga conexión a Internet.

**ARF1.5:** Se utilizará un gestor de bases de datos para almacenar en el dispositivo móvil las ubicaciones del usuario. Además, el dispositivo móvil intentará enviar la información recopilada cada 4 horas con intentos en un intervalo de tiempo una hora, siempre y cuando esté conectado a una red *Wi-Fi*. La información enviada se eliminará del dispositivo móvil para evitar saturar su almacenamiento.

**ARF1.6:** La aplicación tendrá una opción que le permitirá al usuario activar o desactivar el sensor de ubicación (GPS o GPS-A).

**ARF1.7:** Se implementa un servicio de segundo plano para realizar el almacenamiento interno de la ubicación del usuario. Este servicio empezará a funcionar después de que el usuario haya iniciado sesión.

**ARF1.8:** Se realizará un algoritmo para estimar el tipo de movilidad del usuario, ya sea que utilice un medio de transporte, esté caminando o detenido. Además, este algoritmo podrá detectar los patrones de movilidad para saber que usuarios comparten rutas y horarios similares para llegar sus destinos. En el módulo de soluciones ciudadanas será implementado el algoritmo para procesar la información de ubicaciones de los usuarios utilizando la infraestructura de un clúster

**ARF1.9:** Se hará la selección de un esquema de seguridad: MD5, AES, 3DES, SHA o RSA.

#### **4.2.1.2 Análisis de Requerimientos No Funcionales de la aplicación del usuario**

En esta subsección se expone el análisis requerimientos no funcionales de la aplicación del usuario.

**ARNF1.1:** La aplicación se podrá descargar de una tienda de aplicaciones, la cual se instalará de forma similar a otras aplicaciones por el usuario.

**ARNF1.2:** Se desarrollará un programa para detectar la dirección MAC del dispositivo móvil, un programa para deshabilitar el acceso del dispositivo de una dirección MAC establecida.

**ARNF1.3:** Se realizarán los cálculos necesarios para establecer el tamaño de la población de UPIITA a ser entrevistada con el propósito de determinar la factibilidad de implementación del sistema propuesto en un grupo de personas con al menos un mismo destino en común.

#### **4.2.1.3 Análisis de Requerimientos Técnicos de la aplicación del usuario**

En esta subsección se expone el análisis de requerimientos técnicos de la aplicación del usuario.

**ART1.1:** El usuario verificará que su dispositivo cumple con las características de hardware requerido.

**ART1.2:** Se verificará que es posible utilizar el sensor de ubicación GPS, y el uso del Internet en los dispositivos móviles Android en la versión 6.0 o superior.

**ART1.3:** Se realizará una elección de cuál será el sistema gestor de bases de datos y el lenguaje de programación a utilizar en el servidor Web.

#### **4.2.1.4 Análisis de Requerimientos Funcionales de la base de datos**

En esta subsección se expone el análisis de requerimientos funcionales de la base de datos de la aplicación móvil.

**ARF2.1:** Se realizarán los diagramas de clases de ambas bases de datos para conocer los datos a ser almacenados.

**ARF2.2:** Se implementarán las operaciones CRUD (crear, leer, actualizar y borrar) en los gestores de las bases de datos.

#### **4.2.1.5 Análisis de Requerimientos No Funcionales de la base de datos**

En esta subsección se expone el análisis de requerimientos no funcionales de la base de datos de la aplicación móvil.

**ANRF2.1:** Se verificarán los datos almacenados cada hora, para determinar en cuánto tiempo el almacenamiento interno del teléfono podría llegar a un tamaño de 1 MB.

#### **4.2.1.6 Análisis de Requerimientos Técnicos de la base de datos**

En esta subsección se expone el análisis de requerimientos técnicos de la base de datos de la aplicación móvil.

**ART2.1:** Se seleccionará un gestor de base de datos para el almacenamiento interno de la ubicación de los usuarios en los dispositivos móviles. Además, se realizará una elección de cuál será el sistema gestor de bases de datos a utilizar en el servidor Web.

### **4.2.2 Análisis de Requerimientos del módulo de soluciones ciudadanas**

En esta sección se describe el análisis de requerimientos del módulo de soluciones ciudadanas, los cuales serán abordados a continuación:

#### **4.2.2.1 Análisis de Requerimientos Funcionales del clúster**

En esta subsección se expone el análisis de requerimientos funcionales del clúster.

**ARF3.1:** Se registrará y proporcionará un usuario y contraseña al administrador después de instalar un sistema operativo en el clúster.

**ARF3.2:** El administrador registrado en el sistema, requiere que ingrese su nombre de usuario y contraseña para acceder a la plataforma.

**ARF3.3:** Se desarrollará un algoritmo para analizar los patrones de movilidad individual del usuario para estimar si el usuario camina, está detenido o usa un medio de transporte.

**ARF3.4:** Se realizará un algoritmo para detectar los horarios y rutas similares de un grupo de usuarios para sugerir compartir el medio de transporte. El usuario decidirá si acepta la recomendación. Si el usuario acepta la sugerencia del sistema, este podrá ponerse en contacto con la persona(s) con quien(es) podría compartir automóvil.

**ARF3.5:** Se enviará la base de datos de sugerencia para compartir automóvil al servidor Web. Finalmente, los usuarios seleccionados recibirán notificaciones en la aplicación móvil de quienes son los estudiantes con los que podrían compartir sus medios de transporte. La base de datos de sugerencia para compartir automóvil será utilizada en el módulo de indicadores de gobierno.

#### **4.2.2.2 Análisis de Requerimientos No Funcionales del clúster**

En esta subsección se expone el análisis de requerimientos no funcionales del clúster.

**ANR3.1:** El administrador podrá compilar los algoritmos de patrón de movilidad individual y sugerencia de compartir automóvil. sin embargo, el procesamiento de datos no será realizado en tiempo real.

**ANR3.2:** Se añadirán más número de nodos al clúster, si se requiere disminuir el tiempo de procesamiento de los datos.

**ANR3.3:** El clúster no procesará la información en tiempo real debido a las limitaciones del equipo disponible.

#### **4.2.2.3 Análisis de Requerimientos Técnicos del clúster**

En esta subsección se expone el análisis de requerimientos técnicos del clúster.

**ART3.1:** Se revisará que el equipo a ser utilizado tenga las características solicitadas, de no ser así, se usará el equipo que se tenga disponible.

**ART3.2:** Se utilizará un gestor de base de datos para realizar el almacenamiento y procesamiento distribuido de los datos de ubicaciones de los usuarios.

### **4.2.3 Análisis de Requerimientos del módulo de indicadores de gobierno**

En esta sección se describe el análisis de requerimientos del módulo de indicadores de gobierno, los cuales serán abordados a continuación:

#### **4.2.3.1 Análisis de Requerimientos Funcionales del tablero de control**

**ARF4.1:** Se desarrollará una sección en el tablero de control para visualizar un mapa de calor de las zonas con mayor aglomeración.

**ARF4.2:** Se desarrollará una sección en el tablero de control para visualizar las trayectorias origen-destino de los usuarios delimitados por un tiempo seleccionado por el usuario administrador.

**ARF4.3:** Se desarrollará una sección en el tablero de control para el tipo de movilidad individual del usuario.

**ARF4.4:** Utilizando la base de datos de sugerencias de usuarios, se desarrollará una sección en el tablero de control que contenga la visualización de las rutas de los usuarios candidatos a compartir transporte.

#### **4.2.3.2 Análisis de Requerimientos No Funcionales del tablero de control**

En esta subsección se expone el análisis de requerimientos no funcionales del tablero de control.

**ARNF4.1:** El tablero de control dispondrá de diferentes secciones para mostrar los KPIs.

#### **4.2.3.3 Análisis de Requerimientos Técnicos del tablero de control**

En esta subsección se expone el análisis de requerimientos técnicos del tablero de control.

**ART3.1:** Se revisará que el equipo a ser utilizado tenga las características mínimas solicitadas, de no ser así, se usará el equipo que se tenga disponible.

**ART3.2:** Se utilizará un sistema de información geográfica que pueda visualizar los indicadores clave de rendimiento en diferentes navegadores web utilizando un servidor local.

## **4.3 Herramientas para el desarrollo de la propuesta**

En esta sección se presentan las herramientas que serán utilizadas en este proyecto terminal.

### **4.3.1 Elección de herramientas para la aplicación móvil**

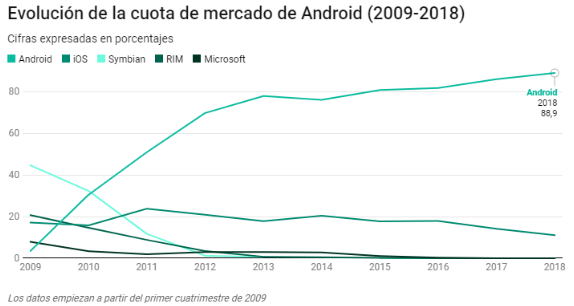
**4.3.1.1 Android**

Actualmente existen diversos sistemas operativos que se utilizan en los teléfonos inteligentes como lo son Android, iOS, Windows Phone, Symbian, BlackBerry, entre otros. A continuación, se muestran algunas de las características principales que son consideradas al momento de elegir el sistema operativo más adecuado para ser desarrollo en el dispositivo móvil. En la tabla 9, se muestran las características de los dos sistemas operativos que tienen una mayor demanda comercial [43] [44].

**Tabla 9.** Características de Android y iOS

|  |  |  |
| --- | --- | --- |
| **Característica** | **Android** | **iOS** |
| Costo de licencia | No | Sí |
| Lenguaje de programación | Java, Kotlin, c# | Swift, C++, Objective-C |
| Entorno de programación | Android Studio, Eclipse, Xamarin | Xcode, Appcode, Xamarin |

A continuación, se muestra en la figura 10, una estadística del porcentaje de usuarios que utilizan los diferentes sistemas operativos que existen en los dispositivos móviles [45].



**Figura 10.** Porcentaje de usuarios Android en el mercado, tomada de [45]

Debido a la existencia de una mayor cantidad de usuarios que utilizan el sistema operativo Android, teniendo un mercado del 88.9% en el 2018 y por ser un sistema operativo de licencia gratuita. Se define que el desarrollo de la aplicación móvil será utilizando el sistema operativo Android.

**4.3.1.2 Android Studio**

Es un entorno de desarrollo integrado (IDE por sus siglas en inglés: *Integrated Development Environment*) para desarrollar aplicaciones en el sistema operativo Android. Este entorno ofrece funciones para mejorar la compilación y la verificación de errores de sintaxis, esto se debe a su compatibilidad con IntelliJ IDEA. A continuación, en la tabla 10, se realiza una reseña de las características de los IDE de desarrollo para Android que existen [46] [47].

**Tabla 10.** Características de Android y iOS

|  |  |  |
| --- | --- | --- |
| **Característica** | **Android Studio** | **Eclipse** |
| Costo de licencia | No | No |
| Lenguaje de programación | Java, Kotlin | Java, c++ |
| Auto-completación de código | Sí | Sí |
| Calidad de software (JUnit) | Sí | Sí |
| Compilación de proyectos | Gradle | Archivos ‘jar’ |
| Crear emuladores | Sí | No |
| Requisitos mínimos de memoria RAM | 2 GB de RAM (8 GB de RAM recomendable) | 4 GB de RAM (8 GB de RAM recomendable) |

En este trabajo, se utilizará Android Studio, debido a los requisitos mínimos que requiere de memoria RAM y por la ventaja de crear emuladores.

**4.3.1.3 JAVA**

Es un lenguaje de programación de propósito general, alto nivel y orientado a objetos; fue diseñado específicamente para permitir a los desarrolladores que sus aplicaciones se escriben una vez y las ejecuten en cualquier plataforma (mejor conocida por sus siglas en inglés como *WORA* que significan “*write once, run anywhere*”) [41]. En tabla 11, se realiza una comparación entre los lenguajes de programación JAVA y Kotlin [48].

**Tabla 11.** Comparación entre Java y Kotlin

|  |  |  |
| --- | --- | --- |
| **Característica** | **Java** | **Kotlin** |
| Métodos extra para su ejecución | No | 7,000 métodos |
| Tamaño extra para la depuración de APK | No | Aproximadamente 1MB |
| Funciones “async/await” | Sí | No |

Se ha seleccionado el lenguaje de programación de JAVA debido a que cuenta con funciones async, lo que permite desarrollar aplicaciones en segundo plano.

**4.3.1.4 SQLite**

El sistema operativo Android incluye el soporte de aplicaciones relacionadas a bases de datos utilizando el gestor de base de datos SQLite, esto se debe a los pocos recursos que utilizan, siendo recomendable para integrarlas en los dispositivos móviles, haciendo posible el almacenamiento interno de la información de manera persistente [49].

**4.3.1.5 Plataforma de Mapas de Google (*Google Maps Platform*)**

Es una API (por sus siglas en inglés “*Application Programming Interface*” o Interfaz de Programación de Aplicaciones) desarrollada por Google que permite proporcionar en dispositivos móviles o páginas web servicios de mapeo web (conocido en inglés como *web mapping*), como son la visualización de la información precisa de mapas (mundo real como son continentes, países, ciudades, calles, etc) estáticos y dinámicos, condiciones de tráfico en tiempo real (Google Traffic), planificación de rutas de un punto A a un punto B para viajar en transporte público, automóvil, a pie o bicicleta, así como permitir traza polígonos, polilíneas y señales que indiquen la posición de múltiples localizaciones [50].

Además, este servicio de Google cuenta con 25 millones de actualizaciones de ubicaciones todos los días, permitiendo al usuario tener información precisa para encontrar destinos específicos que desee localizar. Esta herramienta es únicamente usada para la visualización de mapas en la aplicación móvil.

### **4.3.2 Elección de herramientas para el servidor web**

**4.3.2.1 000Webhost**

Es un servicio de alojamiento de páginas web (*hosting*) de manera gratuita, así como la información a ser almacenada. A continuación, en la tabla 12, se realizó una comparación entre los servidores 000Webhost y Freehostia en su versión gratuita [51] [52]:

**Tabla 12.** Comparación entre 000webhost y Freehostia

|  |  |  |
| --- | --- | --- |
| **Características** | **000webhost** | **Freehostia** |
| Almacenamiento | 1 GB | 250 MB |
| Ancho de banda | 10 GB | 6 GB |
| Número de sitios web permitidos | 2 | 5 |
| Alojamiento gratuito | Sí | Sí, 30 días |
| Soporte para php | Sí | Sí |
| Gestor de base de datos | Mysql | Mysql |
| PhpMyAdmin | Sí | Sí |
| Copias de seguridad instantáneas | Sí | Sí |
| Soporte FTP | Sí | Sí |
| Administración de archivos en web | Sí | Sí |
| Firewall | Sí | Sí |

Se ha seleccionado el servidor Web 000webhost por tener un mayor espacio de almacenamiento y ancho de banda. En este proyecto se utilizará la versión gratuita del servidor Web, teniendo las siguientes limitaciones:

1. 30 conexiones de concurrencia (por base de datos).

2. 15,000 consultas máximas por hora (lectura de la base de datos).

3. 5000 actualizaciones máximas por hora (escritura de la base de datos).

4. Una capacidad de 200 tablas por base de datos.

### **4.3.3 Elección de herramientas para el módulo de soluciones ciudadanas**

**4.3.3.1 Scala**

Es un lenguaje de programación multiparadigma que combina propiedades de lenguajes orientado a objetos con lenguajes funcionales. A continuación, en la tabla 13, se realizó una comparación entre los lenguajes de programación Scala, Java, Python y R [53]:

**Tabla 13.** Comparación entre los lenguajes Java, Scala, Python y R

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Característica** | **Java** | **Scala** | **Python** | **R** |
| Tipo de Lenguaje | Orientado a objetos | Orientado a objetos y funcional | Orientado a objetos y funcional | Específico para Científico de datos |
| Velocidad de procesamiento | Rápido | Rápido (10 veces más rápido que Python) | Lento | Lento |
| Curva de aprendizaje | Fácil | Fácil | Fácil | Moderado |
| Perfil de usuario | Programador Web/Hadoop | Programador *Big Data* | Ingeniería de datos | Científico de datos / Estadísticos |
| Uso | Desarrollo Web y Hadoop | Spark | *Machine Learning* y visualización de datos | Visualización/ Análisis de datos |
| Disponibilidad de bibliotecas de *Machine Learning* | Limitado | Limitado | Extenso | Extenso |
| Utiliza JVM | Sí | Sí | No | No |
| Concurrencia | Soportada | Soportada | No soportada | NA |
| Inmutabilidad | Sí | Sí | No | No |
| Lenguaje nativo de *Apache Spark* | No | Sí | No | No |

Se ha elegido el lenguaje de programación Scala debido a su rápida curva de aprendizaje, su capacidad de inmutabilidad, su similitud con Java y su compatibilidad con el marco de trabajo de *Spark*.

**4.3.3.2 Apache Spark**

Es un marco de trabajo desarrollado para el procesamiento de datos a gran escala en computo paralelo y distribuido, en su página Web oficial hay una prueba de competencia que muestra su rendimiento con respecto a Apache Hadoop de 10 a 100 veces más rápido [54]. A continuación, en la tabla 14, se realizó una comparación entre los marcos de trabajo *Spark* y *Hadoop*:

**Tabla 14.** Comparación entre Apache Spark y Apache Hadoop

|  |  |  |
| --- | --- | --- |
| **Característica** | **Spark** | **Hadoop** |
| Rendimiento | 10 a 100 veces más rápido que Hadoop | HDFS escribe una vez y lee varias veces, generando mayor almacenamiento de ficheros en cada procesamiento de datos |
| Procesamiento | Memoria RAM | Disco duro |
| Almacenamiento | HDFS u otro gestor de base de datos (Cassandra) | HDFS u otro gestor de base de datos (Cassandra) |
| Tipo de Procesamiento | Local o por lotes (distribuido) | Local o por lotes (distribuido) |
| Tolerante a fallos | Sí | Sí |
| Escalable | Sí | Sí |
| Hardware básico (*commodity hardware*) | Sí | Sí |
| Problemas de latencia | No | Sí, debido al procesamiento de Map y Reduce |
| Modelo de programación | Conjunto de datos distribuidos resilientes (RDD) | MapReduce |
| Modelo de trabajo | Maestro - Esclavo | Maestro - Esclavo |
| Lenguajes soportados | Java, Scala, Python y R | Java |

**4.3.3.3 Apache Cassandra**

Es una base de datos columnar de código libre diseñada para aplicaciones analíticas, tiene como propósito leer y escribir datos de manera eficiente desde y hacia el almacenamiento en disco duro. Cassandra es escalable de forma horizontal añadiendo nodos basados en equipo básico (*commodity hardware*), permite la replicación en todos los nodos del clúster para evitar pérdidas de información si un nodo llegará a fallar, además el lenguaje de consulta utilizado por este gestor de bases de datos es CQL, el cual se asemeja a la sintaxis SQL. Los lenguajes soportados son: Java, JavaScript, PHP, Python y Scala [55].

### **4.3.4 Herramientas de visualización**

**4.3.4.1 JavaServer Pages (JSP)**

La tecnología JSP está basada en el lenguaje de programación Java la cual facilitar el desarrollo de páginas web dinámicas. La tecnología JSP permiten incrustar etiquetas HTML y código Java. Además, se puede utilizar en combinación con Servets que manejan la lógica de negocio. JSP es una parte de Java EE (*Enterprise Edition*), una plataforma completa para aplicaciones de clase empresarial [56].

**4.3.4.2 CSS**

Hoja de estilo en cascada o CSS (por sus siglas en inglés *cascading style sheets*), es un lenguaje que contiene la información topográfica de los elementos visuales de la página, el color y la posición del código HTML, además de evitar la duplicidad de estilos en una página web [57].

**4.3.4.3 JavaScript**

Es un lenguaje de programación orientado a objetos utilizado para crear efectos atractivos y dinámicos en las páginas web [58].

**4.3.4.4 D3.JS**

Es una biblioteca de JavaScript para manipular documentos basados en datos. D3 permite desarrollar comportamientos dinámicos para la interacción y la animación en páginas Web [59].

**4.3.4.5 Leaflet**

Leaflet [60] es una biblioteca de JavaScript de código abierto para el desarrollo de mapas interactivos. Esta herramienta permite desarrollar mapas de calor, polilíneas, círculos, marcadores, entre otros. A diferencia de la API de Google, esta biblioteca no requiere de una licencia después de haber transcurrido un año.

**4.3.4.6 MySQL**

Es un sistema de gestión de bases de datos relacionales (RDBMS), multiplataformas basado en un modelo cliente-servidor. Tiene una rápida curva de aprendizaje, escalabilidad vertical, diseñado para realizar operaciones transaccionales. Además, incluye las características ACID (Aislamiento, Concurrencia, Atomicidad y Consistencia) [61].

### **4.3.5 Seguridad en la aplicación desarrollada**

Actualmente, los sistemas de información son utilizados en muchas de las actividades que realizan los ciudadanos de manera cotidiana o servicios que proporcionan las entidades gubernamentales, instituciones públicas o privadas. Todas las actividades que utilizan un sistema de información requieren un esquema de seguridad para evitar ser atacado por algún tipo de virus informático, código malicioso o robo de información para ser distribuida a través del Internet.

Se puede definir al término Seguridad Informática como [62] “una medida que impida la ejecución de operaciones no autorizadas sobre un sistema, cuyos efectos puedan conllevar daños sobre la información, comprometer su confidencialidad, autenticidad o integridad, disminuir el rendimiento de los equipos o bloquear el acceso de usuarios autorizados al sistema”.

A continuación, se describen algunas herramientas que serán utilizadas para realizar las pruebas de seguridad en la aplicación móvil y en el servidor web. Además, se mencionan los esquemas de seguridad que ocupa el sistema distribuido para comunicarse de manera segura entre las diferentes computadoras lo componen.

**4.3.5.1 Autentificación**

El usuario se registrará en el sistema, para iniciar sesión el usuario debe identificarse utilizando su correo y contraseña.

**4.3.5.2 Estándar de cifrado avanzado (AES, Advanced Encryption Standard)**

AES-256 es un estándar de cifrado simétrico [63], es decir, dos o más usuarios tienen una única clave secreta, esta clave será la que cifrará y descifrará la información transmitida a través del canal. Fue desarrollado en 1997 y definido dentro de un marco internacional uniforme y publicado por *National Institute of Standards and Technology* (NIST) [64]. Actualmente es considerado como uno de los más seguros.

La estructura del algoritmo AES puede ser utilizada por claves criptográficas de 128, 192 y 256 bits para encriptar y descifrar datos en bloques de bits. Además, es posible cifrar un mensaje utilizando matrices de estado (matriz de 4x4), la cual realiza rondas de cálculo utilizando operaciones XOR.

Se ha elegido este algoritmo debido a que, diversos estudios [64] han realizado pruebas para descifrar la llave cifrada por este algoritmo, utilizando Tianhe-2, una de las supercomputadoras más rápidas del mundo en el 2016, tardaría millones de años descifrando el cifrado AES-256 bits.

# **Capítulo 5. Diseño**

En este capítulo se desarrolla el diseño del sistema de información propuesto en este proyecto terminal. En la primera sección se exponen los tres principales algoritmos que se utilizan para obtener la movilidad de cada usuario, encontrar los patrones de movilidad individuales y buscar las recomendaciones de viajes compartidos. En la segunda sección se presentan los casos de uso elaborados con base en los requerimientos, para cada uno se muestra su diagrama y su descripción. En la tercera sección se explica el diseño de la base de datos. En la cuarta sección se muestra el diseño de las clases del sistema propuesto con sus propiedades y métodos. En la quinta sección se especifican los diagramas de secuencia. En la última sección se bosqueja el diseño del tablero de control con sus indicadores.

Los diagramas que se presentan siguen los lineamientos establecidos en la notación UML v2.5.

## **5.1 Diseño de algoritmos del sistema propuesto**

En esta sección se explican los tres algoritmos y elementos que servirán para realizar este sistema.

### **5.1.1 Algoritmo de muestreo**

En esta subsección se aborda el algoritmo de muestreo, teniendo como propósito la obtención de las ubicaciones de los usuarios, para posteriormente ser utilizado en el algoritmo de patrones de movilidad individual y búsqueda de viajes compartidos.

En esta subsección se han utilizado las herramientas de MySQL, SQLite, PHP, Java, y el sistema operativo Android.

#### **5.1.1.1 Aplicación móvil y servidor Web**

Se desarrolla una aplicación móvil para el sistema operativo Android que registra las coordenadas geográficas de los usuarios que estén dados de alta en el sistema. En esta aplicación se requiere la autorización de cada usuario para recabar la ubicación de los lugares donde se desplazan cotidianamente.

Los datos de cada muestra obtenidos por el algoritmo de muestreo que esta implementado en el dispositivo móvil son: **IdUsuario, fecha, hora, latitud y longitud**. La descripción de cada una de las variables se encuentra en la tabla 15.

**Tabla 15.** Datos obtenidos de la aplicación móvil

|  |  |
| --- | --- |
| **Dato** | **Descripción** |
| IdUsuario | Identificador único de cada usuario registrado en el sistema |
| Fecha | Identificador de tiempo con el formato Día/Mes/Año |
| Hora | Identificador del tiempo con el formato Hora:Minuto:Segundo |
| Latitud | Variable correspondiente a la coordenada latitud del GPS del dispositivo móvil muestreado |
| Longitud | Variable correspondiente a la coordenada longitud del GPS del dispositivo móvil muestreado |

Los datos obtenidos de la aplicación móvil se almacenan en una tabla de datos nombrada **ubicación**, utilizando SQLite. Obsérvese la tabla 16.

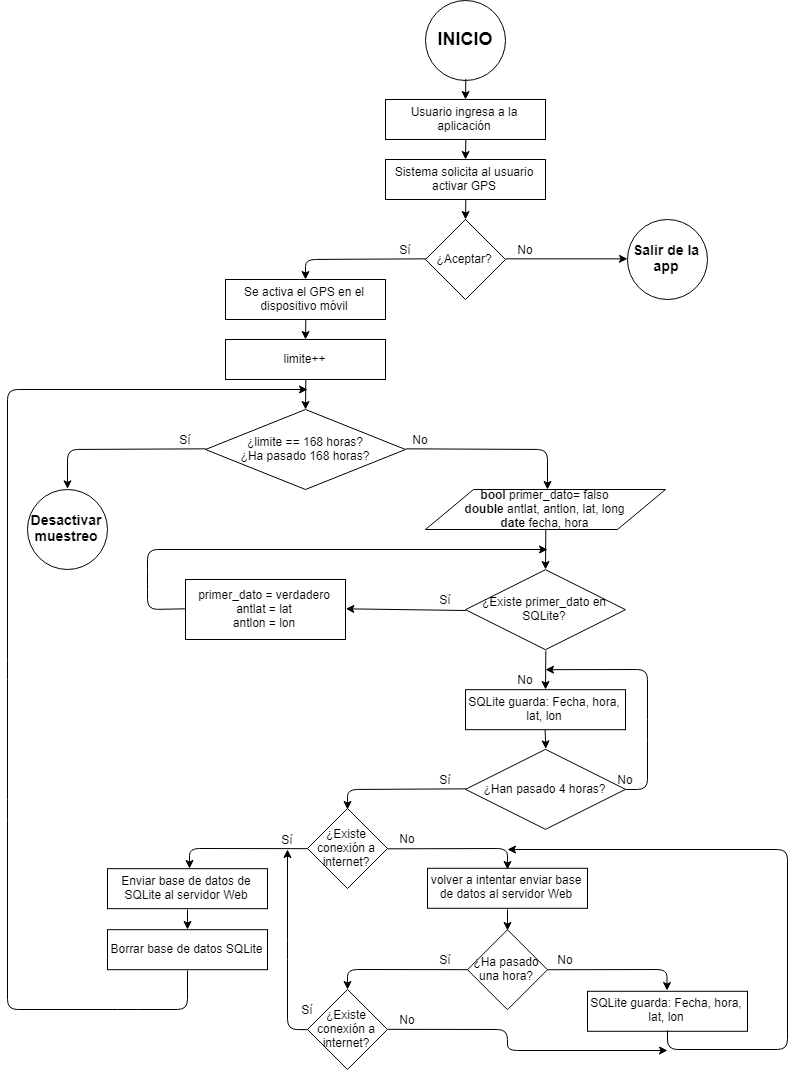
**Tabla 16.** Tabla de datos - ubicación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IdUsuario** | **Fecha** | **Hora** | **Latitud (grados)** | **Longitud (grados)** |
| 1 | 27/07/2019 | 22:00:58 | 0 | 0 |
| 1 | 27/07/2019 | 22:01:58 | 0 | 0 |
| 1 | 27/07/2019 | 22:02:58 | 19.48728 | -99.09528 |
| 1 | 27/07/2019 | 22:03:58 | 19.48762 | -99.09617 |
| 1 | 27/07/2019 | 22:04:58 | 19.486766 | -99.0962 |
| 1 | 27/07/2019 | 22:05:58 | 19.48856 | -99.09862 |
| 1 | 27/07/2019 | 22:15:58 | 19.48156 | -99.1036 |

A continuación, se enlista la serie de pasos para realizar el algoritmo de muestreo:

1. El algoritmo estará implementado dentro de la aplicación móvil.
2. El algoritmo comenzará a funcionar una vez que el usuario inicie sesión.
   1. Si es la primera vez que el usuario inicializa la aplicación, el sistema le solicitará al usuario activar el sensor GPS de su dispositivo móvil.
      1. Si el usuario acepta el permiso, el sensor de GPS será activado por la aplicación móvil y comenzará a almacenar en una base de datos interna la información de la ubicación del usuario cada minuto.
3. El algoritmo verificará si existe la primera muestra en la base de datos de ubicación del usuario.
   1. Si no existe ninguna muestra, se inicializa un contador para detener el funcionamiento de la aplicación después de un periodo de 7 días consecutivos.
4. El algoritmo transferirá la información de la base de datos de la ubicación del usuario del dispositivo móvil a un servidor web cada 4 horas, siempre y cuando exista conexión a Internet *Wi-Fi*.
   1. Si no existe conexión a internet, el sistema volverá a intentarlo durante periodos de una hora.
5. El algoritmo procederá a borrar la base de datos interna del dispositivo móvil después de haber enviado la información al servidor Web, con el propósito de evitar la saturación del almacenamiento del dispositivo móvil.

A continuación, se muestra en la figura 11 el diagrama de flujo del algoritmo de muestreo.

****

**Figura 11.** Diagrama de flujo del algoritmo de muestreo.

Además, el perfil del usuario se almacenará en el servidor Web, esta información es obtenida cuando el usuario se registra en la aplicación móvil, la tabla es nombrada **perfil** y contiene los atributos que se muestra en la tabla 17.

**Tabla 17.** Tabla de datos - perfil

|  |  |  |  |
| --- | --- | --- | --- |
| **Descripción** | **Nombre variable** | **Tipo** | **Comentario** |
| Identificador usuario | idU | INT | NA |
| Identificador nombre | nombre | VARCHAR | NA |
| Identificador primer apellido | apellido1 | VARCHAR | NA |
| Identificador segundo apellido | apellido2 | VARCHAR | NA |
| Fecha de nacimiento | fechaNacimiento | DATE | NA |
| Género | genero | INT | 1-Mujer/2-Hombre/3-Otros |
| Correo electrónico | correo | VARCHAR | NA |
| Tipo de usuario | rol | INT | Peatón / Conductor |
| ¿Puede aportar alguna remuneración por viaje? | aportarDinero | INT | 1 - Sí / 2 - No |
| ¿Desea obtener alguna remuneración económica? | costoViaje | INT | 1 - Sí / 2 - No |
| Marca del auto | marcaAuto | VARCHAR | NA |
| Modelo del auto | modelo | VARCHAR | NA |
| Color del auto | color | VARCHAR | NA |
| Número de placas del auto | numPlacas | VARCHAR | NA |
| Preferencia de pasajeros | preferenciaP | INT | 1 - Mujer / 2 - Hombre /  3 -Otros/ 4 - Indiferente |
| Número de asientos Disponibles | numA | INT | NA |
| Licencia de conducir | licenciaC | VARCHAR | NA |
| Escuela | escuela | VARCHAR | NA |
| Carrera | carrera | VARCHAR | NA |
| Código único de acceso | codigoUnico | VARCHAR | NA |
| Número de asientos disponibles | numAsiento | INT | NA |
| Contraseña | contraseña | VARCHAR | NA |

#### **5.1.1.2 Estimación del almacenamiento del servidor Web**

Para calcular el volumen de una base de datos se debe realizar el siguiente procedimiento:

1. Obtener el tamaño de cada tipo de dato del gestor de base de datos [65][66] donde se almacenarán los datos en el servidor Web. Obsérvese en la tabla 18.

**Tabla 18.** Tamaño de los tipos de datos del gestor de base de datos MySQL 8.0

|  |  |
| --- | --- |
| **Tipo de dato** | **Tamaño Requerido** |
| TINYINT | 1 byte |
| SMALLINT | 2 bytes |
| MEDIUMINT | 3 bytes |
| INT, INTEGER | 4 bytes |
| BIGINT | 8 bytes |
| FLOAT (p) | 4 bytes si 0<= p <= 24  8 bytes si 25 <= p <= 53 |
| FLOAT | 4 bytes |
| DOUBLE [PRECISIÓN], REAL | 8 bytes |
| YEAR | 1 byte |
| DATE | 3 bytes |
| TIME | 3 bytes |
| DATETIME | 8 bytes |
| TIMESTAMP | 4 bytes |
| CHAR (***M***) | ***M*** × ***w*** bytes, <= ***M*** <= 255, donde ***w*** es el número de bytes requeridos para el tamaño máximo de caracteres en el conjunto de caracteres. |
| VARCHAR (***M***) | ***L*** + 1 bytes si los valores de columna requieren 0 − 255 bytes,  ***L*** + 2 bytes si los valores pueden requerir más de 255 bytes |
| TINYBLOB | ***L*** + 1 bytes, donde ***L*** < 28 |
| BLOB | ***L*** + 2 bytes, donde ***L*** < 216 |
| MEDIUMBLOB | ***L*** + 3 bytes, donde ***L*** < 224 |
| LONGBLOB | ***L*** + 4 bytes, donde ***L*** < 232 |

Donde ***M*** representa la longitud de la columna declarada en caracteres para los tipos de cadenas no binarias y bytes para los tipos de cadenas binarias. ***L*** representa la longitud real en bytes de un valor de cadena dado.

1. Sumar la cantidad de bytes correspondientes de cada columna por cada tabla. Obsérvese en la tabla 19.

**Tabla 19.** Tamaño de las variables utilizadas en la tabla de datos ubicación

|  |  |  |
| --- | --- | --- |
| **Variable** | **Tipo de dato** | **Tamaño Requerido** |
| IdUsuario | INT | 4 bytes |
| Fecha | DATE | 3 bytes |
| Hora | TIME | 3 bytes |
| Latitud | DOUBLE | 8 bytes |
| Longitud | DOUBLE | 8 bytes |

Se obtiene un total de **26 bytes** por cada registro en la tabla **ubicación**.

1. Multiplicar el tamaño de bytes de las tablas por la cantidad de filas que puede llegar a tener cada tabla.

Se observa en la tabla 20, la estimación de la base de datos con un muestreo de un minuto y un usuario utilizando el sistema.

**Tabla 20.** Estimación del almacenamiento de datos de 1 usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tiempo** | **Bytes** | **KB** | **MB** | **GB** |
| 1 minuto | 26 |  |  |  |
| 1 hora | 1560 | 1.52 |  |  |
| 1 día (24 horas) | 37440 | 36.5625 |  |  |
| 1 semana (7 días) | 262080 | 255.94 | 0.24 |  |
| 1 mes (30 días) | 1123200 | 1096.88 | 1.07 |  |
| 6 meses | 6739200 | 6581.25 | 6.42 |  |
| 1 año (12 meses) | 13478400 | 13162.5 | 12.85 |  |

Se observa en la tabla 21, la estimación de la base de datos con un muestreo de un minuto y 131 usuarios utilizando el sistema.

**Tabla 21.** Estimación del almacenamiento de datos de 131 usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tiempo** | **Bytes** | **KB** | **MB** | **GB** |
| 1 minuto | 3406 | 3.32 |  |  |
| 1 hora | 204360 | 199.57 | 0.19 |  |
| 1 día (24 horas) | 4904640 | 4789.69 | 4.67 |  |
| 1 semana (7 días) | 34332480 | 33527.81 | 32.74 |  |
| 1 mes (30 días) | 147139200 | 143690.63 | 140.32 | 0.13 |
| 6 meses | 882835200 | 862143.75 | 841.94 | 0.82 |
| 1 año (12 meses) | 1765670400 | 1724287.5 | 1683.87 | 1.64 |

### **5.1.2 Algoritmo de detección de patrones de movilidad individual**

En esta subsección se presenta el algoritmo de detección de patrones de movilidad individual del usuario. A partir de la base de datos de ubicación de los usuarios generada por el algoritmo de muestreo que se encuentra almacenada en el servidor Web, se desarrolla el análisis correspondiente a la detección de los patrones de movilidad para identificar el tipo de movilidad del usuario en bloques de un minuto con los valores: detenido, caminando, utilizando un transporte vehicular o movilidad errónea.

En esta subsección se han utilizado las herramientas de Java, Scala y Cassandra.

A continuación, se enlistan y se describen los pasos necesarios para realizar el algoritmo que calcula los patrones de movilidad individual.

* **Descargar la base de datos ubicación y perfil del servidor Web**

Se descargará la base de datos ubicación y perfil del servidor Web a la computadora local para realizar el proceso de limpieza de datos, detección del patrón individual de movilidad y sugerencia de compartir automóvil.

* **Limpieza de datos**

En esta parte, se realiza la limpieza de datos de la base de datos ubicación. En este proceso se eliminan los datos atípicos (también denominados *outliers*, representa un valor muy grande de los otros valores que se encuentran en un conjunto de datos). En este trabajo, se eliminan los registros con latitud y longitud igual a 0, debido a que el GPS del dispositivo móvil no obtuvo una coordenada geográfica. Obsérvese en la tabla 22.

**Tabla 22.** Tabla de datos - ubicación

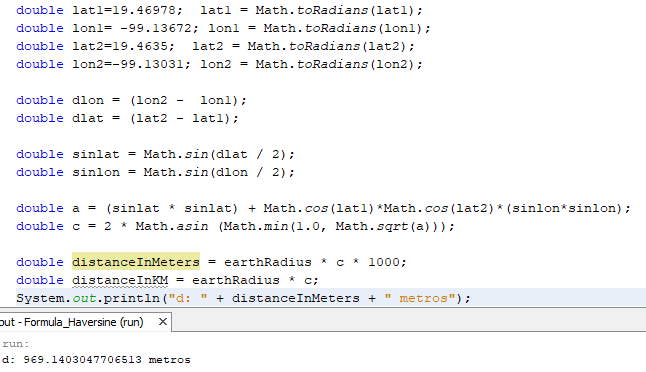
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IdUsuario** | **Fecha** | **Hora** | **Latitud (grados)** | **Longitud (grados)** |
| 1 | 27/07/2019 | 22:02:58 | 19.48728 | -99.09528 |
| 1 | 27/07/2019 | 22:03:58 | 19.48762 | -99.09617 |
| 1 | 27/07/2019 | 22:04:58 | 19.486766 | -99.0962 |
| 1 | 27/07/2019 | 22:05:58 | 19.48856 | -99.09862 |
| 1 | 27/07/2019 | 22:15:58 | 19.48156 | -99.1036 |

Una vez finalizado este proceso de limpieza de datos, se genera una base de datos con el nombre **ubicación**, siendo esta la versión final de la base de datos que fue descargada del servidor Web.

* **Conversión de coordenadas geográficas a unidades métricas de longitud en kilómetros**

Inicialmente, el algoritmo utiliza como datos de entrada la latitud y longitud de la base de datos de la ubicación del usuario. Posteriormente, se requiere convertir las coordenadas geográficas de cada muestra de ubicación del usuario a unidades métricas de distancia en kilómetros (km). Para realizar esta conversión, se utilizará la expresión matemática de Haversine, obsérvese en la ecuación 1, y el radio equivolumen de la Tierra.

Para verificar el funcionamiento de la fórmula de Haversine, descrita en la ecuación 1, se ha realizado un programa en Java y se ha corroborado utilizando utilizado Google MyMaps. A continuación, se muestran en la figura 12 y 13, las pruebas anteriormente mencionadas



**Figura 12.** Resultado de la distancia entre dos puntos geográficos utilizando Java.



**Figura 13.** Resultado de la distancia entre dos puntos geográficos utilizando Google MyMaps

Se ha llegado a la conclusión de que Google MyMaps utiliza el radio equivolumen de la Tierra para obtener la distancia, verificando así, que se ha obtenido el mismo resultado para obtener la distancia entre dos puntos geográficos.

Utilizando los datos de la tabla 21, se calcula la distancia entre los puntos consecutivos de cada muestra cómo se observa en la tabla 23.

**Tabla 23.** Tabla de datos - ubicación y distancia

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **IdUsuario** | **Fecha** | **Hora** | **Latitud (grados)** | **Longitud (grados)** | **Distancia(m)** |
| 1 | 27/07/2019 | 22:02:58 | 19.48728 | -99.09528 | 0 |
| 1 | 27/07/2019 | 22:03:58 | 19.48762 | -99.09617 | 100.66 |
| 1 | 27/07/2019 | 22:04:58 | 19.486766 | -99.0962 | 5.44 |
| 1 | 27/07/2019 | 22:05:58 | 19.48856 | -99.09862 | 272.7 |
| 1 | 27/07/2019 | 22:15:58 | 19.48156 | -99.1036 | 937.21 |

* **Obtención de la primera derivada (velocidad)**

En esta segunda etapa, se aborda la obtención de la primera derivada.

Para calcular la velocidad del usuario en un intervalo de tiempo, se debe realizar la primera derivada. Para ello, se requiere utiliza la ecuación 1, mencionada anteriormente en la etapa 1 y el intervalo de tiempo que existe en la distancia calculada.

Posteriormente se realiza la operación de diferencia entre la distancia final e inicial del desplazamiento y la diferencia entre el intervalo de tiempo final e inicial.

La definición de la velocidad es equivalente a la primera derivada, expresada mediante la fórmula (2).

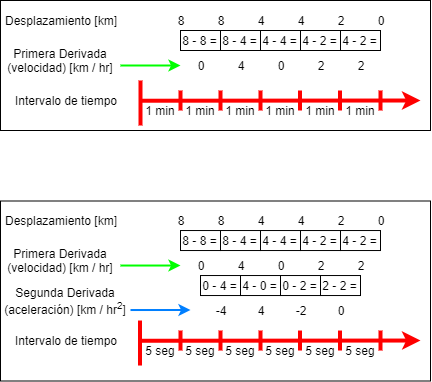
(2)

|  |
| --- |
|  |

donde:

|  |  |  |
| --- | --- | --- |
| **Símbolo** | **Descripción** | **Unidades** |
|  | Posición inicial en el eje X | kilómetros |
|  | Posición final en el eje X | kilómetros |
|  | Tiempo en la posición inicial | horas |
|  | Tiempo en la posición final | horas |
|  | Velocidad | kilómetros/horas |

En la figura 14, se muestra un ejemplo de cómo se realiza la primera derivada en este proyecto terminal utilizando una ventana de tiempo de un minuto por muestra. Se ha propuesto este intervalo debido al rango de precisión del GPS (*RAW GPS*) o mediciones GNSS (sistema global de navegación por satélite o por sus siglas en inglés *Global Navigation Satellite System*), este rango mide aproximadamente 50 metros [68]. Si un usuario promedio se desplaza a una velocidad de 5 km/h [67] [68], se tendría un tiempo aproximado a 36 segundos. Para este trabajo será utilizado una muestra por minuto, con el motivo de evitar posibles errores del rango de precisión del GPS que se encontraron al utilizar una ventana de tiempo de 30 segundos.



**Figura 14.** Obtención de la primera derivada

Al realizar la operación de la primera derivada en este proyecto terminal, existe la posibilidad de inhibir en el tipo de movilidad que esté utilizando el usuario, ya sea que esté caminando, utilizando un medio de transporte, o detenido.

Utilizando la primera derivada, se obtiene la velocidad en que se desplaza un usuario, como se observa en la tabla 24.

**Tabla 24.** Tabla de datos - ubicación, distancia y velocidad

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **IdUsuario** | **Fecha** | **Hora** | **Latitud (grados)** | **Longitud**  **(grados)** | **Distancia**  **(m)** | **Tiempo**  **(s)** | **Velocidad**  **(km/h)** |
| 1 | 27/07/2019 | 22:02:58 | 19.48728 | -99.09528 | 0 | 0 | 0 |
| 1 | 27/07/2019 | 22:03:58 | 19.48762 | -99.09617 | 100.66 | 60 | 6.03 |
| 1 | 27/07/2019 | 22:04:58 | 19.486766 | -99.0962 | 5.44 | 60 | 0.32 |
| 1 | 27/07/2019 | 22:05:58 | 19.48856 | -99.09862 | 272.7 | 60 | 16.36 |
| 1 | 27/07/2019 | 22:15:58 | 19.48156 | -99.1036 | 937.21 | 600 | 0.56 |

Se han propuesto cuatro tipos de movilidades que podría tener un usuario si se conoce la velocidad en la que se desplaza durante cada intervalo de muestra obtenida. Se ha empleado el rango de velocidades propuesto en cada caso para disminuir los posibles errores de precisión del GPS del celular de cada usuario que podrían afectar en este algoritmo [69]. A continuación, se describen los siguientes casos:

* **Detenido:** Una persona en promedio camina por lo menos a una velocidad de 1 km/h, por este motivo, se ha propuesto un rango de 0 a 0.49 km/h.
* **Caminando:** Una persona en promedio camina de 1 a 5 km/h, por este motivo, se ha propuesto un rango de 0.5 a 9.9 km/h
* **Transporte:** Una persona que utiliza un transporte público o auto se desplaza aproximadamente a más de 10 km/h, por este motivo, se ha propuesto un rango de 10 a 129 km/h
* ***Outlier*:** Un usuario promedio que utiliza un medio de transporte no puede viajar a una velocidad mayor de 130 km/h en un intervalo de tiempo de un minuto.

A continuación, se describen los parámetros para determinar el tipo de movilidad que está teniendo cada usuario a partir de los rangos propuestos en la tabla 25. En la tabla 26, se muestra el tipo de movilidad de un usuario considerando los rangos de velocidades anteriormente mencionados.

**Tabla 25.** Condiciones para determinar el tipo de movilidad del usuario

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de movilidad** | **Rango de Velocidad** | **Código de movilidad** | **Descripción** |
| Detenido | 0km/ h ≤ Velocidad ≤ 0.5km/h | 0 | Usuario se desplaza menos de 10 metros en un minuto |
| Caminando | 0.5km/h ≤ Velocidad < 10km/h | 1 | Un usuario se desplaza caminando |
| Transporte | 10km/h ≤ Velocidad < 130km/h | 2 | Un usuario se desplaza en un medio de transporte |
| *Outlier* | (𝑑𝑖𝑠𝑡𝑎𝑛𝑐𝑖𝑎>2166𝑚 ⋀ 𝑡>300) ⋁ 𝑡𝑖𝑒𝑚𝑝𝑜>300 𝑠 ⋁ 𝑑𝑖𝑠𝑡𝑎𝑛𝑐𝑖𝑎>2166𝑚 | 10 | Un usuario supera el límite de tiempo y distancia que puede desplazarse en automóvil |

En la tabla 26, se describen el tipo de movilidad que tiene el usuario de la tabla 24.

**Tabla 26.** Tipo de movilidad del usuario

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IdUsuario** | **Fecha** | **Hora** | **Latitud (grados)** | **Longitud**  **(grados)** | **Distancia**  **(m)** | **Tiempo (s)** | **Velocidad**  **(km/h)** | **Código de movilidad** |
| 1 | 27/07/2019 | 22:02:58 | 19.48728 | -99.09528 | 0 | 0 | 0 | 0 |
| 1 | 27/07/2019 | 22:03:58 | 19.48762 | -99.09617 | 100.66 | 60 | 6.03 | 1 |
| 1 | 27/07/2019 | 22:04:58 | 19.486766 | -99.0962 | 5.44 | 60 | 0.32 | 0 |
| 1 | 27/07/2019 | 22:05:58 | 19.48856 | -99.09862 | 272.7 | 60 | 16.36 | 2 |
| 1 | 27/07/2019 | 22:15:58 | 19.48156 | -99.1036 | 937.21 | 600 | 0.56 | 10 |

#### **5.1.2.1 Análisis para calcular los patrones de movilidad individual**

A continuación, se describen la serie de pasos para calcular el patrón de movilidad del usuario anteriormente mencionada en esta subsección, obsérvese en la figura 15.

**Fórmula de Haversine**

Se codifica la fórmula de Haversine dentro de un método.

1. **For NumUsuarios.**

Se almacena en una variable el número total de fechas de cada usuario en cada iteración.

Se almacena en un *ArrayList* todas las fechas correspondientes de cada usuario en cada iteración, sin duplicidad.

|  |  |
| --- | --- |
| **ArrayListFecha** | **{2019-01-01, 2019-01-02, 2019-01-03}** |

1. **For NumFechaPorUsuario**

Se almacena en variables de tipo arrayList los atributos hora, lat y lon en cada fecha de cada usuario en cada iteración.

1. **For AuxFechaLength**

Se realiza la ejecución del punto 4. Antes de terminar cada iteración, se almacenan los atributos distancia, velocidad, hora, fecha y IdUsuario en una tabla.

1. **If PrimerDato ==false**
   1. Se realiza la inicialización de los atributos latitud, longitud, hora, tipo de movilidad y se inicializa el primer registro de cada fecha.

**Else**

* 1. Se calcula la distancia utilizando el método de la fórmula de Haversine. Se Acumula la distancia.
     1. **If PrimerV == false**

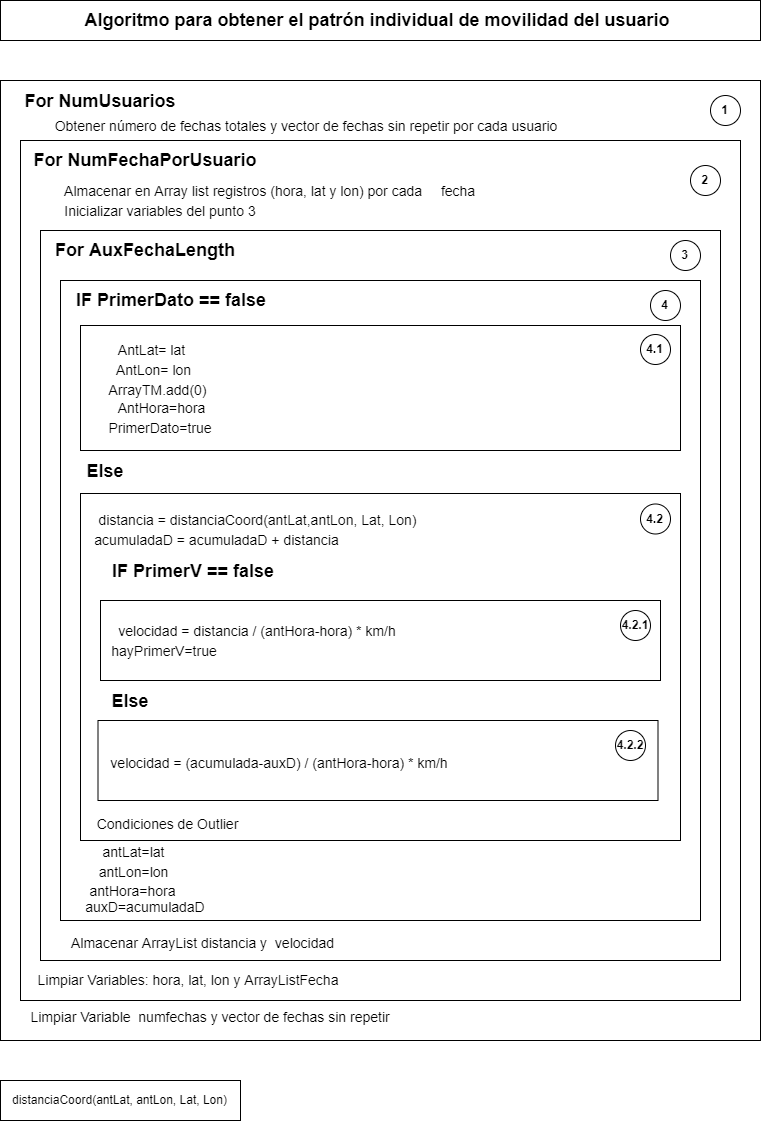
Se obtiene la velocidad en Km/h y se inicializa el primer registro de velocidad de cada fecha.

* + 1. **Else**

Se obtiene la velocidad en km/h

velocidad = (acumulada-auxD) / (antHora-hora) \* km/h

Se identifica y almacena el tipo de movilidad mencionada en la tabla 28 en cada iteración.



**Figura 15.** Diseño del algoritmo de detección de tipo de movilidad por usuario

### **5.1.3 Algoritmo de Búsqueda de viajes compartidos**

En esta subsección se presenta el algoritmo de búsqueda de viajes compartidos. A partir de la base de datos de ubicación de los usuarios generada por el algoritmo de muestreo. A continuación, se desarrolla el análisis correspondiente para realizar dicho algoritmo.

A partir de un estudio realizado por la empresa Moovit, se ha recopilado la información relacionada a los hábitos de movilidad de los usuarios de la Ciudad de México [70].

* La distancia promedio que camina un usuario por viaje es de un kilómetro para llegar a un destino específico, donde 4 de cada 10 personas caminan más de un kilómetro.
* El tiempo de espera promedio de un usuario en una parada o estación es de 11 minutos, sin embargo, 32% de las personas esperan más de 20 minutos.

Utilizando la información anteriormente mencionada, se propone delimitar una distancia y tiempo máximo de **un kilómetro** y **30 minutos** de espera respectivamente, para aumentar la posibilidad de obtener sugerencias de compartir automóvil. Esto se debe al número de usuarios que utilizan la aplicación móvil, teniendo un total de **131** estudiantes, como se observa en el análisis realizado en el anexo de este trabajo.

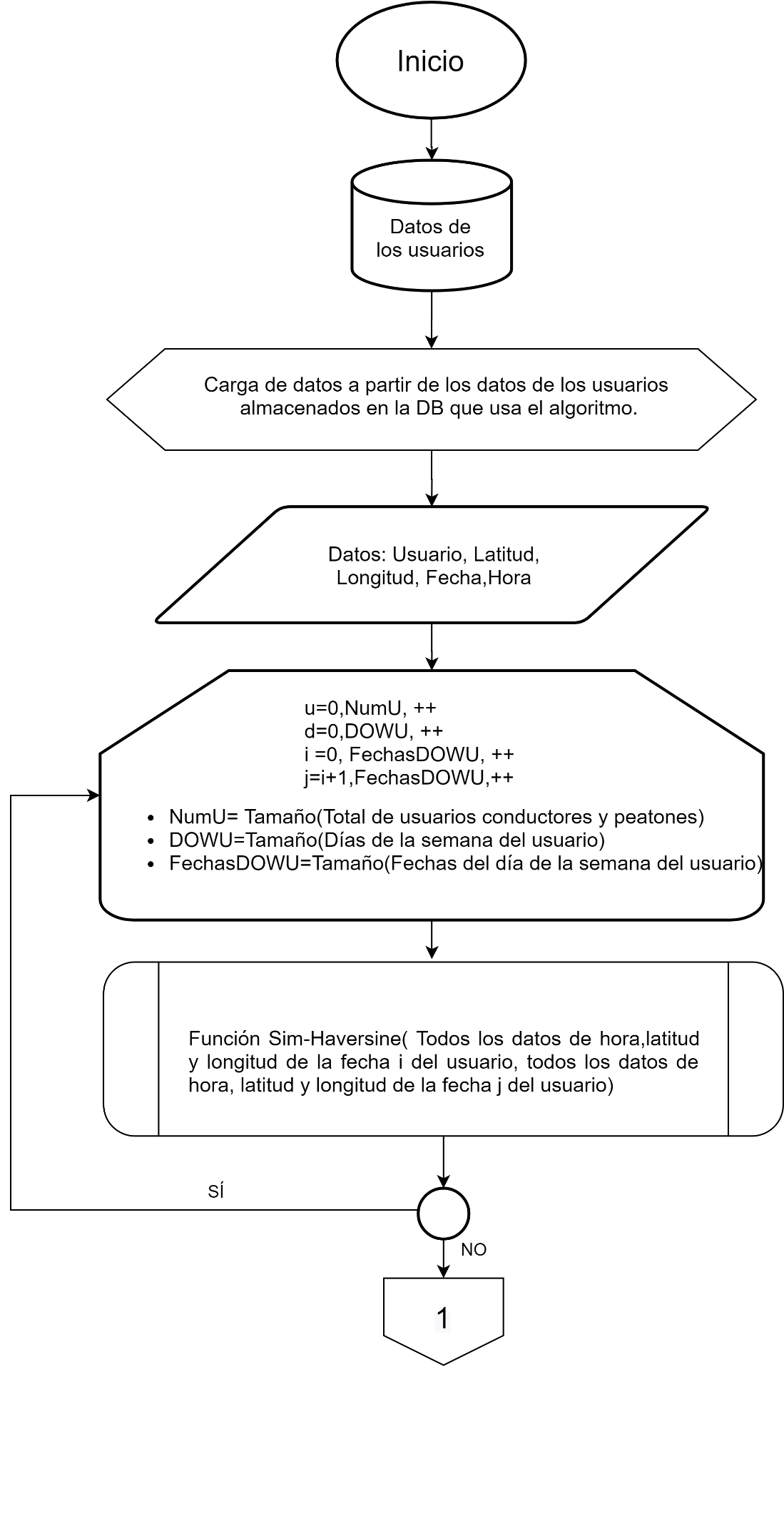
Una sugerencia de viajes compartidos se obtiene cuando dos usuarios (un peatón y un conductor o un conductor y otro conductor) tienen una ruta frecuente semejante establecido por la ventana de espacio-tiempo anteriormente establecida. Este algoritmo utiliza la base de datos de perfil y ubicación, obtenidos del servidor Web.

A continuación, se describen la serie de pasos para definir el algoritmo que calcula sugerencias de viajes compartidos, así como los posibles casos de sugerencias, obsérvese en las figuras 16, 17, 18 y 19.

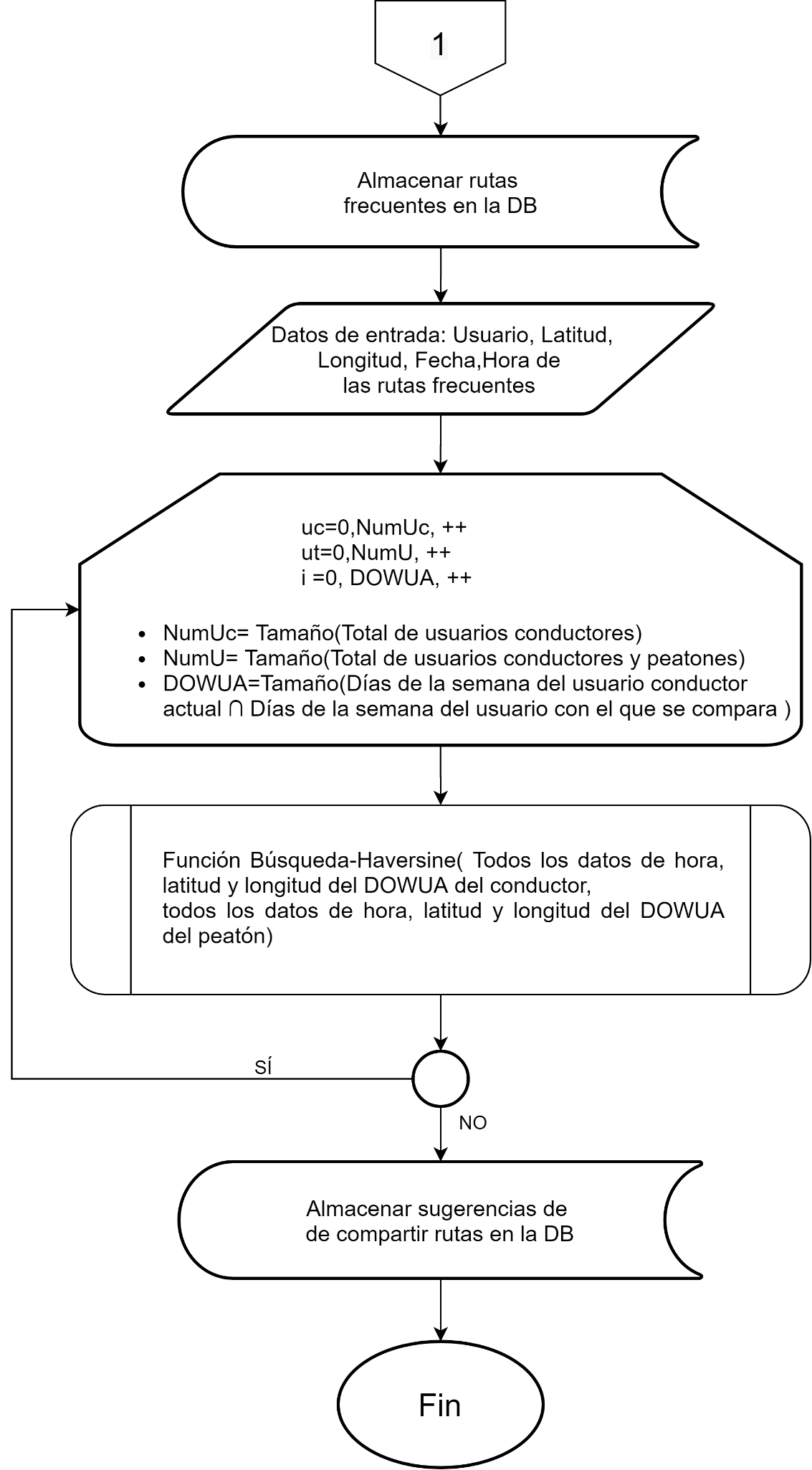
#### **5.1.3.1 Análisis para calcular sugerencias de viajes compartidos**

1. Se deben determinar si existe una ruta frecuente para cada día de la semana por cada uno de los usuarios. Se considera que una ruta de un usuario es una **ruta frecuente**, si existe al menos los datos de ubicaciones de 3 rutas de un mismo día de la semana con fechas distintas, las cuales si el algoritmo al evaluar todas las rutas de un usuario de un mismo día de la semana encuentra patrones de movilidad en espacio-tiempo similares en la mayoría de las rutas (75% de las rutas), se califica como ruta frecuente de ese día de la semana.
2. Para determinar una ruta frecuente, se deben comparar los datos de latitud y longitud de un día de la semana con respecto a los datos de latitud y longitud del mismo día, pero con fecha distinta. Este proceso se repetirá para las demás fechas de ese mismo día y posteriormente con cada uno de los días de la semana restantes.
3. Se utiliza la fórmula de Haversine para obtener la distancia entre cada una de las coordenadas geográficas de cada uno de los registros correspondientes al punto 2.
4. **Si la distancia entre dos puntos geográficos es menor o igual a 300 metros** (aproximadamente la distancia de 2 cuadras), esos registros **se consideran que están dentro de la ventana de espacio**.
5. Después de la evaluación del punto cuatro, **si los registros cumplen con el punto anterior en al menos en dos kilómetros continuos de la ruta del usuario** (y esa distancia es al menos el 80% de la ruta total evaluada) **se encuentra dentro de la ventana de espacio, entonces la ruta cumple con la primera condición de coincidencia en espacio**.
6. **Si al menos un punto geográfico de la ruta evaluada que cumple con el punto cinco está dentro de la ventana de tiempo**, esa ruta **cumple** con las condiciones de la **ventana de espacio-tiempo** y, por lo tanto, se califica como **rutas con patrones de movilidad similares**. Pueden existir múltiples rutas similares, pero distintas con otras rutas similares *p. ej.* laruta con fecha *‘a’* es similar a las rutas con fecha *‘b’, ‘c’, ‘f’, ‘g’ y ‘h’* mientras que la ruta con fecha *‘d’* es similar a la ruta *‘e’* pero distinta a las anteriores, por lo que se clasifican las rutas similares obteniendo un valor de frecuencia.
7. Si la ruta similar con el valor mayor de frecuencia (moda) representa el 75% de las rutas evaluadas de ese día de la semana, entonces esa ruta se califica como una **ruta frecuente**. Los días de la semana que tengan una ruta frecuente serán evaluadas después con las rutas frecuentes de los demás usuarios.
8. Se deben comparar cada uno de los registros de latitud y longitud de cada uno de los días de la semana de las rutas frecuentes de cada usuario conductor con respecto a cada usuario peatón o conductor.
9. Si la distancia entre dos puntos geográficos obtenida a partir de la fórmula de Haversine es menor o igual a un kilómetro, esos registros se consideran que están dentro de la ventana de espacio anteriormente propuesto.
10. Después de la evaluación del punto nueve, **si los registros cumplen con el punto anterior en al menos en dos kilómetros continuos de la ruta del conductor comparados con una ruta del peatón están dentro de la ventana de espacio, entonces, la ruta cumple la primera condición de coincidencia en espacio.** Adicionalmente para la evaluación de un caso conductor-conductor, esa ruta debe representar el 95% de la distancia total de la ruta.
11. **Si al menos un punto geográfico de una ruta que cumple con el punto cuatro está dentro de la ventana de tiempo, esa ruta cumple con las condiciones de la ventana de espacio-tiempo y, por lo tanto, cumplen con una sugerencia de viaje compartido.**
12. Si la ruta cumple con el punto cinco, los datos almacenados son los siguientes: Id sugerencia, id pasajero, id conductor, fecha, día, hora inicio, hora de bajada, punto de subida, punto de bajada, posición del peatón, distancia, tiempo caminando, correo conductor, correo peatón.

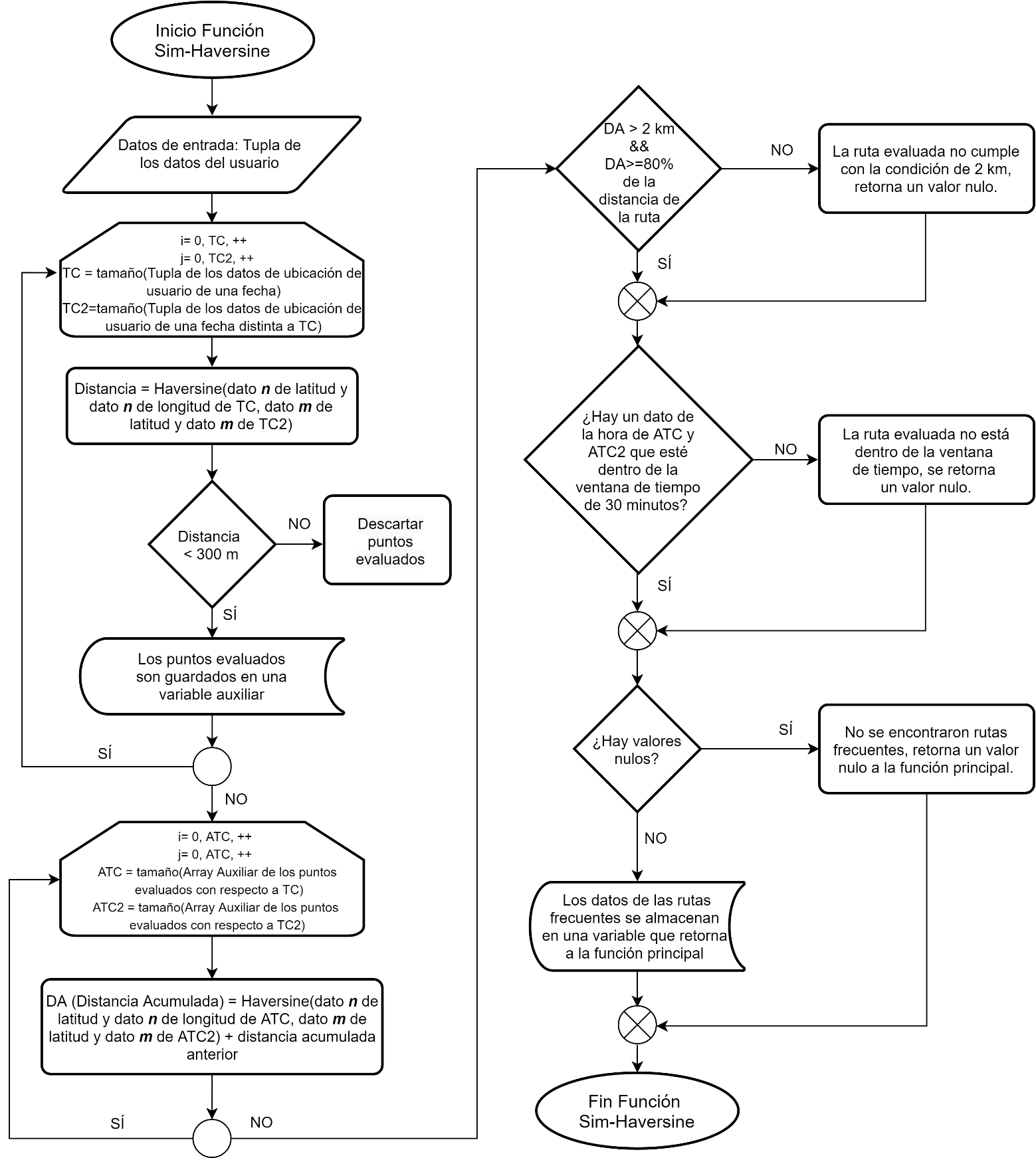
#### **5.1.3.2 Diagramas de flujo del algoritmo de viajes compartido.**



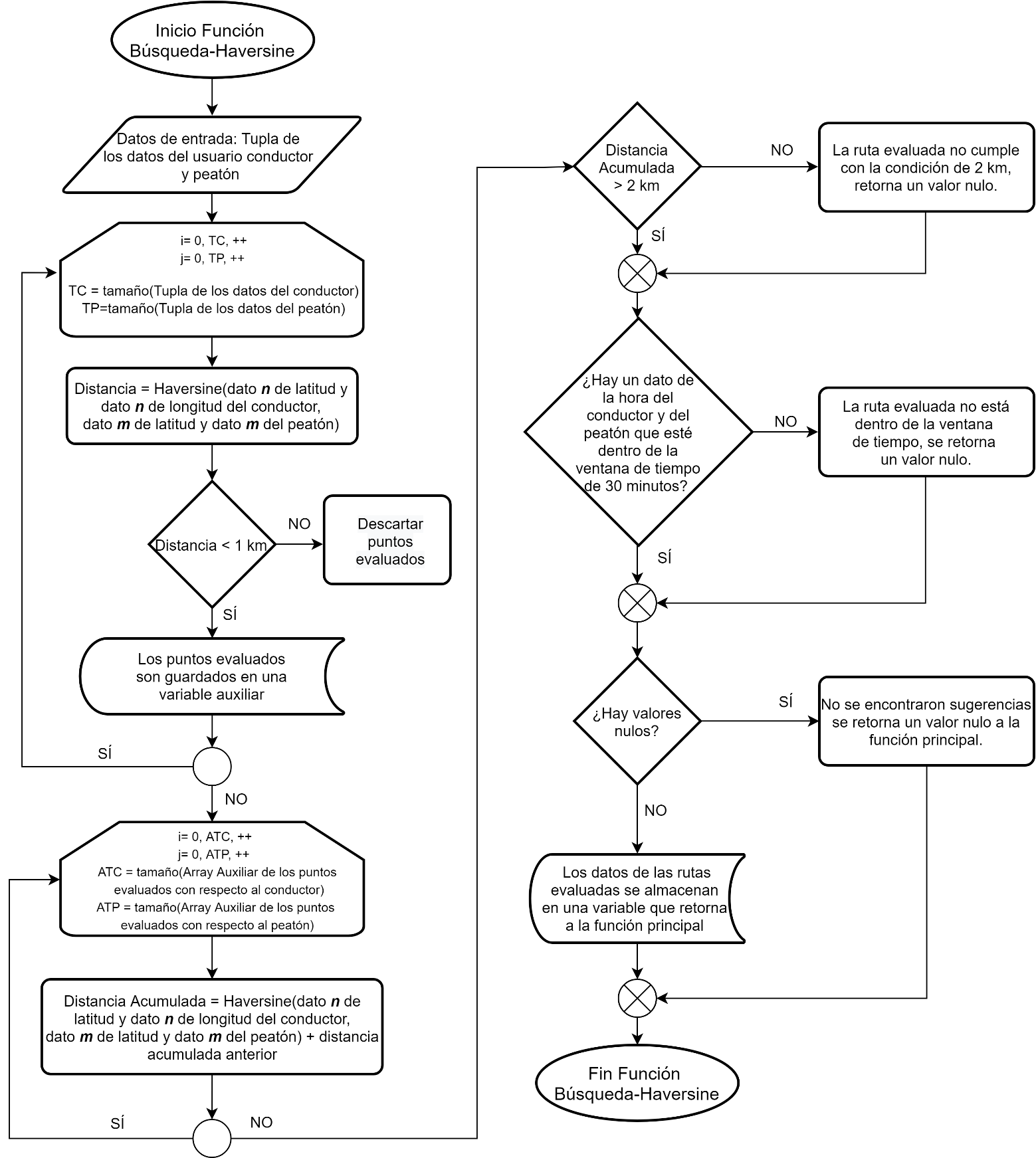
***Figura 16.*** *Parte 1 del diseño del algoritmo para calcular viajes compartidos*



***Figura 17.*** *Parte 2 del diseño del algoritmo para calcular viajes compartidos*



***Figura 18.*** *Diseño del proceso predefinido Función Sim-Haversine para el algoritmo para calcular viajes compartidos*



***Figura 19.*** *Diseño del proceso predefinido Función Búsqueda-Haversine para el algoritmo para calcular viajes compartidos*

#### **5.1.3.3 Modelo matemático del algoritmo de viajes compartido.**

Sea una familia finita numerable de N usuarios conductores donde , es la familia de M tuplas de un usuario conductor.

Sea una familia finita numerable de N usuarios peatones donde , es la familia de M tuplas de un usuario peatón.

Para saber si un usuario conductor o un usuario peatón tiene rutas frecuentes, es necesario evaluar sus datos contenidas en sus tuplas, donde el día de la semana o sea el mismo contenido en las tuplas con fechas o distintas.

Los elementos de las tuplas de un usuario conductor **,** para cada día , con distintas fechas formarán parte de las tuplas de , donde es el número de días de la semana que contienen fechas distintas. A las tuplas de se les calcula la distancia de Haversine usando la fórmula (1) y las tuplas evaluadas que cumplen con la condición (4), formarán parte las tuplas de , incluyendo que son las tuplas de las horas de . Esto se repite para un usuario peatón **.**

(4)

A las tuplas de se les calcula la suma de sus distancias de Haversine:

(5)

(6)

Donde:

es la distancia total de la ruta de un conductor que cumple con la condición (4)

es la distancia total de otra ruta de ese mismo conductor que cumple con la condición (4)

Para que se considere una ruta que cumple con la ventana de distancia mínima de 2 kilómetros, se debe observar la condición (7):

(7)

Adicionalmente a la condición (7), las distancias de ydeben representar un mínimo del 80% de la ruta original evaluada.

Para conocer si una ruta que cumple con la condición (7) está dentro de la ventana de tiempo de 30 minutos, debe existir al menos una evaluación con las tuplas de las horas de que esté dentro de ese lapso, por lo tanto, se debe cumplir con la condición (8):

(8)

Si se cumplen con las condiciones anteriores, entonces existe una ruta frecuente cómo se indica en la expresión (9):

(9)

El conjunto de rutas frecuentes de un usuario conductor encontradas se agrupan en cómo se indica en la expresión (10):

(10)

Para el caso del usuario peatón son los mismos pasos, por lo que su expresión (11) queda de la siguiente manera:

(11)

Las rutas frecuentes de un usuario conductor y de un usuario peatón serán evaluadas con base en los días de la semana que tienen en común.

(12)

Donde:

es una tupla que contiene las tuplas del usuario conductor y el usuario peatón que intersecan con los días de la semana de ambos.Cuando significa que no existen días en las que se puedan evaluar las rutas del peatón y del conductor.

A las tuplas de y de contenidas en se les calcula la distancia de Haversine usando la fórmula (1) y las tuplas evaluadas que cumplen con la condición (13), formarán parte las tuplas de , incluyendo y que son las tuplas de las horas de y de **.**

(13)

A las tuplas de se les calcula la suma de sus distancias de Haversine:

(14)

(15)

Donde:

es la distancia total de la ruta de un conductor que cumple con la condición (13)

es la distancia total de la ruta de un peatón que cumple con la condición (13)

Para que se considere una ruta que cumple con la ventana de distancia mínima de 2 kilómetros, se debe observar la condición (7):

(16)

Para conocer si una ruta cumple con la condición (7) está dentro de la ventana de tiempo de 30 minutos, debe existir al menos una evaluación con las tuplas de las horas de que esté dentro de ese lapso, por lo tanto, se debe cumplir con la condición (8):

(17)

Si se cumplen con las condiciones anteriores, entonces existe una sugerencia para compartir un viaje como se indica en la expresión (9):

(18)

El conjunto de sugerencias encontradas se agrupan en cómo se indica en la expresión (10):

(19)

#### **5.1.3.3 Escenario válido para sugerir viajes compartidos**

La ruta del conductor comparada con la ruta de peatones cumple con la ventana de espacio-tiempo, sentido y dirección hacia un mismo destino. En la figura 20, se muestran dos rutas que cumplen la condición de este escenario, donde se tiene como destino en común la UPIITA.

|  |  |
| --- | --- |
| Sentido de la ruta del peatón  **Simbología**  UPIITA  Metro Lindavista  Metro Potrero  Ruta del conductor: Potrero-UPIITA  PH 7:10  PH 7:15  PH 7:30  PH 7:45  VH 7:20  VH 7:25  VH 7:30  VH 7:35  Ruta del peatón: Lindavista-UPIITA  **PH:** Hora de la ubicación del peatón  **VH:** Hora de la ubicación del conductor  Sentido de la ruta del conductor |  |

**Figura 20.** Escenario válido 1

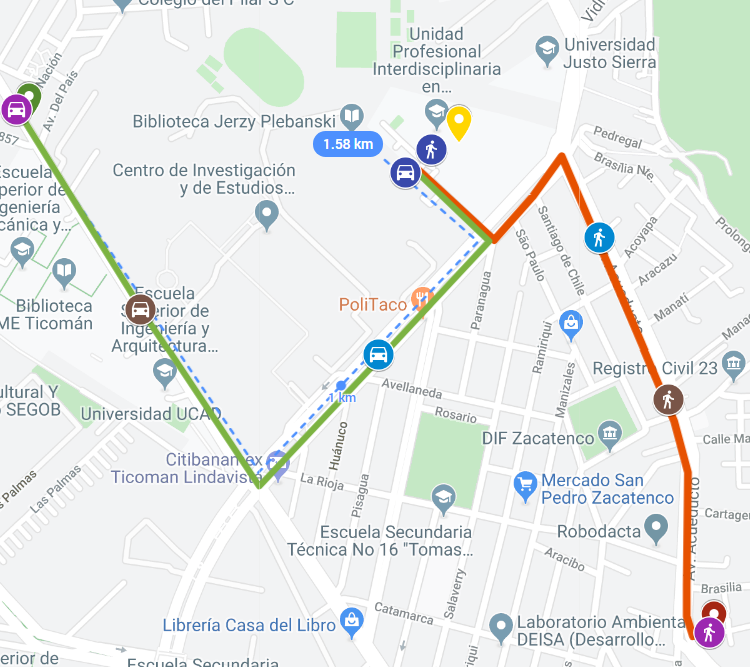
#### **5.1.3.4 Escenarios descartados para sugerir viajes compartidos**

Los escenarios que no cumplen con alguna condición son:

* + 1. Las rutas comparadas entre un conductor y un peatón no cumplen con la ventana de espacio de dos kilómetros, obsérvese en la figura 21.

|  |  |
| --- | --- |
|  |  |

**Figura 21.** Escenario descartado 1

* + 1. Las rutas comparadas entre un conductor y un peatón cumplen con el punto uno de la subsección 5.1.3.1, sin embargo, la ruta continua es menor a dos kilómetros. En la figura 22, se muestra un escenario con estos parámetros.

|  |  |
| --- | --- |
|  |  |

**Simbología**



UPIITA



Santa María Ticomán

Av. Acueducto, San Pedro



Ruta del peatón



Ruta del conductor

**PH:** Hora de la ubicación del peatón

**VH:** Hora de la ubicación del conductor



PH 10:55



PH 11:03



PH 11:11



PH 11:20



VH 11:21



VH 11:22



VH 11:23



VH 11:24

**Figura 22.** Escenario descartado 2

* + 1. La ruta del conductor y del peatón cumplen con el punto dos de la subsección 5.1.3.1, sin embargo, el sentido de las rutas es opuesta. En la figura 23, se muestra un escenario con estos parámetros.

|  |  |
| --- | --- |
| Sentido de la ruta del peatón  **Simbología**  UPIITA  Metro Lindavista  Metro Potrero  Ruta del conductor: Potrero-UPIITA  PH 12:20  PH 12:30  PH 12:45  PH 13:02  VH 7:20  VH 7:25  VH 7:30  VH 7:35  Ruta del peatón: Lindavista-UPIITA  **PH:** Hora de la ubicación del peatón  **VH:** Hora de la ubicación del conductor  Sentido de la ruta del conductor |  |

**Figura 23.** Escenario descartado 3

* + 1. La ruta del conductor y del peatón tienen el mismo sentido, pero horarios fuera del lapso propuesto. En la figura 24, se muestra un escenario con estos parámetros.

|  |  |
| --- | --- |
| Sentido de la ruta del peatón  **Simbología**  UPIITA  Metro Lindavista  Metro Potrero  Ruta del conductor: Potrero-UPIITA  PH 12:20  PH 12:30  PH 12:45  PH 13:02  VH 7:20  VH 7:25  VH 7:30  VH 7:35  Ruta del peatón: Lindavista-UPIITA  **PH:** Hora de la ubicación del peatón  **VH:** Hora de la ubicación del conductor  Sentido de la ruta del conductor |  |

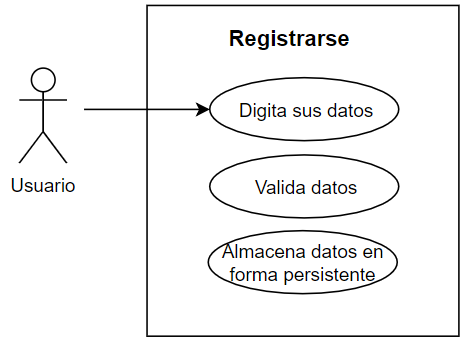
**Figura 24.** Escenario descartado 4

## **5.2 Casos de uso del sistema propuesto**

En esta sección se presentan los casos de uso de la funcionalidad esperada para encontrar las recomendaciones de viajes compartidos y de esta forma reducir la cantidad de vehículos en una ciudad.

### **5.2.1 Caso de uso: Registrarse**

En esta sección se presenta el caso de uso Registrarse. En la figura 25, se muestra el diagrama del caso de uso Registrarse. En la tabla 27, se detalla la descripción del caso de uso Registrarse.



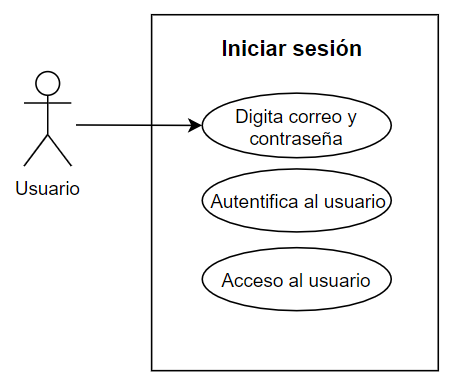
**Figura 25.** Diagrama de caso de uso - Registrarse

**Tabla 27.** Caso de uso - Registrarse

|  |  |
| --- | --- |
| **Nombre** | **Registrarse** |
| Actores | Usuario |
| Versión | 0.1 (Abril 2019) |
| Descripción | Cuando un usuario activa el ícono de la aplicación móvil, el sistema despliega campos vacíos y un botón para que el usuario escriba su correo electrónico, su contraseña y pulse el botón registrarse |
| Disparador | El usuario activa el ícono de la aplicación móvil |
| Precondiciones | El usuario debe tener instalada la aplicación y tener un código de activación |
| Postcondiciones | El sistema valida la información ingresada por el usuario y despliega la vista “iniciar sesión” |
| Flujo Normal | 1. El usuario debe pulsar la opción “registrarse” de la vista “principal” 2. El usuario debe ingresar los datos solicitados por el sistema: 3. Código de acceso único 4. Nombre completo 5. Correo electrónico 6. Teléfono celular 7. Sexo 8. Contraseña 9. Disponibilidad de automóvil 10. Permitir compartir su automóvil 11. Preferencia de pasajeros (hombre o mujer) 12. Fecha de nacimiento 13. El usuario debe pulsar el botón “aceptar” 14. El sistema almacena datos de manera persistente en el dispositivo del usuario 15. El sistema despliega la vista “iniciar sesión” |
| Flujos alternativos | 3.1 Si el usuario no tiene la información solicitada, deberá registrase en el sistema  3.2 El usuario podrá regresar a la vista “Índice”. |
| Excepciones | 3.1 Si la conexión a internet está fallando en el sistema cuando el usuario desee ingresar los datos solicitados, el sistema desplegará el mensaje ‘Problemas de conexión a Internet’  3.2 Si los datos ingresados por el usuario no se encuentran en la base de datos del servidor web, se desplegará el mensaje ‘El usuario o contraseña no existen en el sistema’. |
| Prioridad | Alta |
| Frecuencia de uso | Baja |
| Reglas de negocio | El usuario debe ser un estudiante de UPIITA vigente |
| Requerimientos especiales | El usuario debe tener un código único para poder registrarse en la aplicación móvil.  La aplicación móvil solo estará disponible para dispositivos con el sistema operativo Android y la versión 6 o superior. |
| Suposiciones | El usuario debe ser un estudiante de UPIITA vigente y debe tener conexión a internet |

### **5.2.2 Caso de uso: Iniciar sesión**

En esta sección se presenta el caso de uso Iniciar sesión. En la figura 26, se muestra el diagrama del caso de uso Iniciar sesión. En la tabla 28, se detalla la descripción del caso de uso Iniciar sesión.



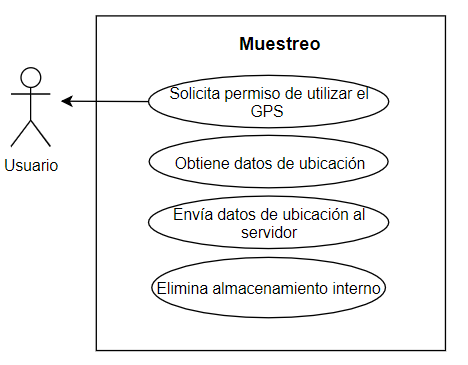
**Figura 26.** Diagrama de caso de uso - Iniciar sesión

**Tabla 28.** Caso de uso - Iniciar sesión

|  |  |
| --- | --- |
| **Nombre** | **Iniciar sesión** |
| Actores | Usuario |
| Versión | 0.1 (Abril 2019) |
| Descripción | Cuando un usuario activa el ícono de la aplicación móvil, el sistema despliega campos vacíos y un botón para que el usuario escriba su correo electrónico, su contraseña y pulse el botón Iniciar sesión |
| Disparador | El usuario activa el ícono de la aplicación móvil |
| Precondiciones | El usuario debe tener instalada la aplicación y estar registrado en la aplicación móvil |
| Postcondiciones | El sistema valida la información ingresada por el usuario y despliega la vista “muestreo” |
| Flujo Normal | 1. El usuario debe pulsar en la opción “iniciar sesión” de la vista “principal” 2. El usuario debe ingresar los datos solicitados por el sistema:    1. correo electrónico    2. contraseña 3. El usuario debe pulsar en el botón “aceptar” 4. El sistema autentifica los datos ingresados por el usuario 5. El sistema despliega la vista “muestreo” |
| Flujos alternativos | 3.1. Si el usuario no tiene la información solicitada, deberá registrarse en el sistema.  3.2 El usuario podrá regresar a la vista “principal”. |
| Excepciones | 3.1 Si la conexión a Internet está fallando en el sistema cuando el usuario desee ingresar los datos solicitados, el sistema desplegará el mensaje ‘Problemas de conexión a Internet’  3.2 Si los datos ingresados por el usuario no se encuentran en la base de datos del servidor web, se desplegará el mensaje ‘El usuario o contraseña no existen en el sistema’. |
| Prioridad | Alta |
| Frecuencia de uso | Baja |
| Reglas de negocio | El usuario debe ser un estudiante de UPIITA vigente |
| Requerimientos especiales | La aplicación móvil solo estará disponible para dispositivos con el sistema operativo Android y la versión 6 o superior. |
| Suposiciones | El usuario debe ser un estudiante de UPIITA vigente y debe tener conexión a internet |

### **5.2.3 Caso de uso: Muestreo**

En esta sección se presenta el caso de uso Muestreo. En la figura 27, se muestra el diagrama del caso de uso Muestreo. En la tabla 29, se detalla la descripción del caso de uso Muestreo.



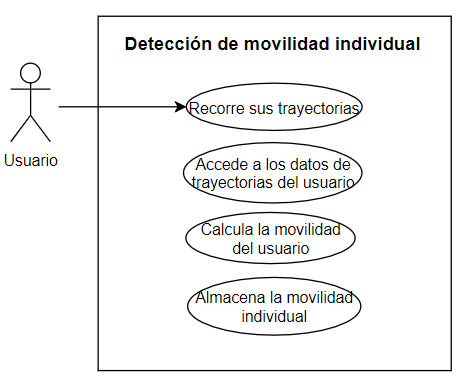
**Figura 27.** Diagrama de caso de uso - Muestreo

**Tabla 29.** Caso de uso - Muestreo

|  |  |
| --- | --- |
| **Nombre** | **Muestreo** |
| Actores | Usuario |
| Versión | 0.1 (Abril 2019) |
| Descripción | El sistema solicita al usuario activar el sensor de GPS de su dispositivo móvil para obtener la localización, la fecha y la hora de donde se encuentra el usuario cada 10 segundos utilizando un servicio en segundo plano |
| Disparador | El usuario activa el sensor GPS de su dispositivo móvil |
| Precondiciones | El usuario debe iniciar sesión en la aplicación móvil |
| Postcondiciones | El sistema almacena los datos de ubicación del usuario en un servidor web |
| Flujo Normal | 1. El sistema solicita al usuario activar el GPS de su dispositivo 2. El usuario activa el sensor GPS de su dispositivo móvil 3. El sistema almacena en una base de datos dentro del dispositivo del usuario la ubicación geográfica, el día y la hora del usuario cada 10 segundos 4. El sistema envía los datos almacenados de ubicación del usuario a un servidor web cada 4 horas 5. El sistema elimina la base de datos que se encuentra dentro del dispositivo móvil |
| Flujos alternativos | 4.1 Si el sistema no tiene conexión a Internet después de las 4 horas establecidas para enviar la base de datos de ubicación de usuario al servidor Web, el sistema volverá a enviar la información en periodos de una hora hasta que el dispositivo se conecte a una red *Wi-Fi*. |
| Excepciones | 3.1 Si existe un problema en el sistema cuando se esté almacenando la información del usuario, el sistema desplegará el mensaje ‘Problemas con el almacenamiento interno’  4.1 Si la conexión a Internet está fallando en el sistema al momento de enviar los datos de ubicación del usuario al servidor web, el sistema desplegará el mensaje ‘Problemas con la conexión a Internet’ |
| Prioridad | Alta |
| Frecuencia de uso | Alta |
| Reglas de negocio | El usuario debe ser un estudiante de UPIITA vigente |
| Requerimientos especiales | La aplicación móvil solo estará disponible para dispositivos con el sistema operativo Android y la versión 6 o superior |
| Suposiciones | El usuario tendrá activado el GPS y el dispositivo encendido en todo momento  El dispositivo del usuario tendrá conexión a Internet *Wi-Fi* cuando el sistema almacene su información de ubicación en el servidor web |

### **5.2.4 Caso de uso: Detección de movilidad individual**

En esta sección se presenta el caso de uso Detección de movilidad individual. En la figura 28, se muestra el diagrama del caso de uso Detección de movilidad individual. En la tabla 30, se detalla la descripción del caso de uso Detección de movilidad individual.



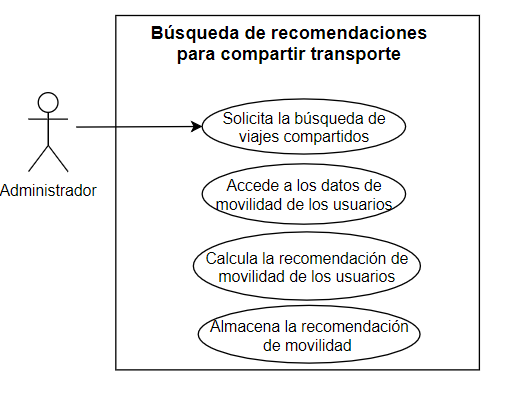
**Figura 28.** Diagrama de caso de uso - Detección de movilidad individual

**Tabla 30.** Caso de uso - Detección de movilidad individual

|  |  |
| --- | --- |
| **Nombre** | **Detección de movilidad individual** |
| Actores | Usuario |
| Versión | 0.1 (Abril 2019) |
| Descripción | El sistema calcula el tipo de movilidad del usuario utilizando la base de datos de ubicación del usuario. |
| Disparador | El usuario solicita al sistema calcular lar la movilidad individual del usuario. |
| Precondiciones | Tener la base de datos de ubicación del usuario en el servidor web |
| Postcondiciones | Se obtendrá el tipo de movilidad que está teniendo cada usuario |
| Flujo Normal | 1. Las trayectorias de movilidad de cada usuario se almacenan en el servidor web 2. El sistema accede a la base de datos de ubicación del usuario 3. El sistema calcula la velocidad del usuario 4. El sistema calcula la aceleración del usuario 5. El sistema calcula el tipo de movilidad del usuario 6. El sistema almacena la movilidad individual en el servidor web |
| Flujos alternativos | 6.1 El sistema almacena la movilidad individual en el clúster |
| Excepciones | 2.1 Si la conexión a Internet está fallando en el sistema al momento de descargar la base de datos de ubicación del usuario, el sistema desplegará el mensaje ‘Problemas con la conexión a Internet’  6.1 Si la conexión a Internet está fallando en el sistema al momento de almacenar la base de datos de movilidad individual, el sistema desplegará el mensaje ‘Problemas con la conexión a Internet’ |
| Prioridad | Alta |
| Frecuencia de uso | Media |
| Reglas de negocio | El usuario debe ser un estudiante de UPIITA vigente |
| Requerimientos especiales | La aplicación móvil solo estará disponible para dispositivos con el sistema operativo Android y la versión 6 o superior |
| Suposiciones | El sistema estará conectado a Internet en todo momento |

### **5.2.5 Caso de uso: Búsqueda de recomendaciones de movilidad**

En esta sección se presenta el caso de uso Búsqueda de recomendaciones de movilidad. En la figura 29, se muestra el diagrama del caso de uso Búsqueda de recomendaciones para compartir transporte. En la tabla 31, se detalla la descripción del caso de uso Búsqueda de recomendaciones para compartir transporte.



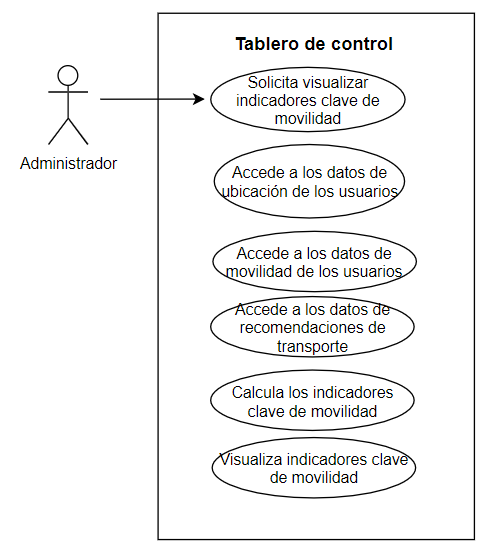
**Figura 29.** Diagrama de caso de uso - Búsqueda de recomendaciones para compartir transporte

**Tabla 31.** Caso de uso - Búsqueda de recomendaciones para compartir transporte

|  |  |
| --- | --- |
| **Nombre** | **Búsqueda de recomendaciones para compartir transporte** |
| Actores | Administrador |
| Versión | 0.1 (Abril 2019) |
| Descripción | El administrador solicita al sistema la búsqueda de recomendaciones de viajes compartidos. El sistema calcula la co-movilidad de los usuarios y almacena la recomendación de movilidad. |
| Disparador | El administrador solicita la búsqueda de recomendaciones de viajes compartidos al sistema |
| Precondiciones | El sistema debe tener acceso a la base de datos de movilidad de usuarios del servidor web |
| Postcondiciones | El administrador obtiene quienes son los usuarios que pueden compartir transporte |
| Flujo Normal | 1. El administrador solicita la búsqueda de recomendaciones de viajes compartidos al sistema 2. El sistema accede a los datos de movilidad de los usuarios del servidor web 3. El sistema calcula la movilidad de los usuarios 4. El sistema almacena la recomendación de movilidad 5. El sistema muestra al administrador quiénes son los usuarios que pueden tener viajes compartidos |
| Flujos alternativos | 2.1 El sistema descarga los datos de movilidad de los usuarios del servidor web |
| Excepciones | 2.1 Si la conexión a internet está fallando en el sistema al momento de descargar la base de datos de movilidad del usuario, el sistema desplegará el mensaje ‘Problemas con la conexión a internet’ |
| Prioridad | Alta |
| Frecuencia de uso | Media |
| Reglas de negocio | Los usuarios candidatos a compartir transporte deberán tener coincidencias de rutas en mínimo 2 kilómetros, en un rango de 250 metros y una ventana de tiempo de 10 minutos. |
| Requerimientos especiales |  |
| Suposiciones | El muestreo de la ubicación de los usuarios habrá sido de 7 días sin interrupciones. |

### **5.2.6 Caso de uso: Visualización**

En esta sección se presenta el caso de uso Tablero de control. En la figura 30, se muestra el diagrama del caso de uso Tablero de control. En la tabla 32, se detalla la descripción del caso de uso Tablero de control.



**Figura 30.** Diagrama de caso de uso - Tablero de control

**Tabla 32.** Caso de uso - Tablero de control

|  |  |
| --- | --- |
| **Nombre** | **Búsqueda de recomendaciones para compartir transporte** |
| Actores | Administrador |
| Versión | 0.1 (Abril 2019) |
| Descripción | El administrador solicita al sistema la visualización de indicadores clave de movilidad de los usuarios. El sistema accede a las bases de datos de ubicación de usuarios, movilidad de usuarios y recomendación de transporte para mostrar en un navegador web los indicadores clave de movilidad como pueden ser: trayectorias origen destino, movilidad caminando, corriendo, transporte público, automóvil privado, número de usuarios que comparten automóvil, número de rutas que comparten los usuarios |
| Disparador | El administrador solicita la visualización de indicadores clave de movilidad de los usuarios al sistema |
| Precondiciones | El sistema debe tener acceso a la base de datos de ubicación de los usuarios, movilidad de los usuarios y soluciones ciudadanas |
| Postcondiciones | El administrador observa los indicadores clave de movilidad solicitados al sistema |
| Flujo Normal | 1. El administrador solicita la visualización de indicadores de movilidad al sistema 2. El sistema accede a las bases de datos del servidor web 3. El sistema muestra varias opciones de indicadores clave de rendimiento |
| Flujos alternativos | 2.1 El sistema descarga las bases de los datos de movilidad de los usuarios del servidor web  3.1 El sistema muestra el número de usuarios que están utilizando la aplicación  3.2 El sistema muestra el número de usuarios de género masculino y femenino  3.3 El sistema muestra los mapas de calor con zonas de aglomeración: Mucha, mediana y poca aglomeración  3.4 El sistema muestra las trayectorias de origen destino de los usuarios  3.5 El sistema muestra el tipo de movilidad individual de un usuario: Caminando, corriendo, metro, transporte público, privado, compartiendo medio de transporte.  3.6 El sistema visualiza las rutas que comparten los usuarios |
| Excepciones | 2.1 Si la conexión a Internet está fallando en el sistema al momento de descargar la base de datos de movilidad del usuario o soluciones ciudadanas, el sistema desplegará el mensaje ‘Problemas con la conexión a Internet’ |
| Prioridad | Alta |
| Frecuencia de uso | Media |
| Reglas de negocio | Únicamente el usuario administrador podrá tener acceso al tablero de control |
| Requerimientos especiales |  |
| Suposiciones | El sistema estará conectado a Internet en todo momento |

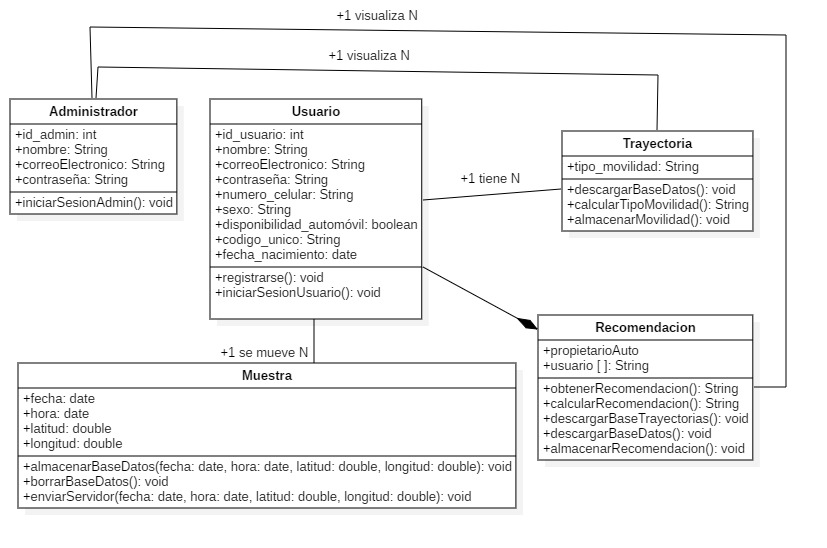
## **5.3 Clases del sistema propuesto**

En esta sección se presentan las clases que conforman el sistema de recomendación de viajes compartidos propuesto en este proyecto terminal.

### **5.3.1 Diagrama de clases**

En esta sección se presenta el diagrama de clases del sistema de información propuesto en este proyecto terminal.

En la figura 31, se muestra el diagrama de clases del sistema propuesto



**Figura 31.** Diagrama de clases del sistema propuesto

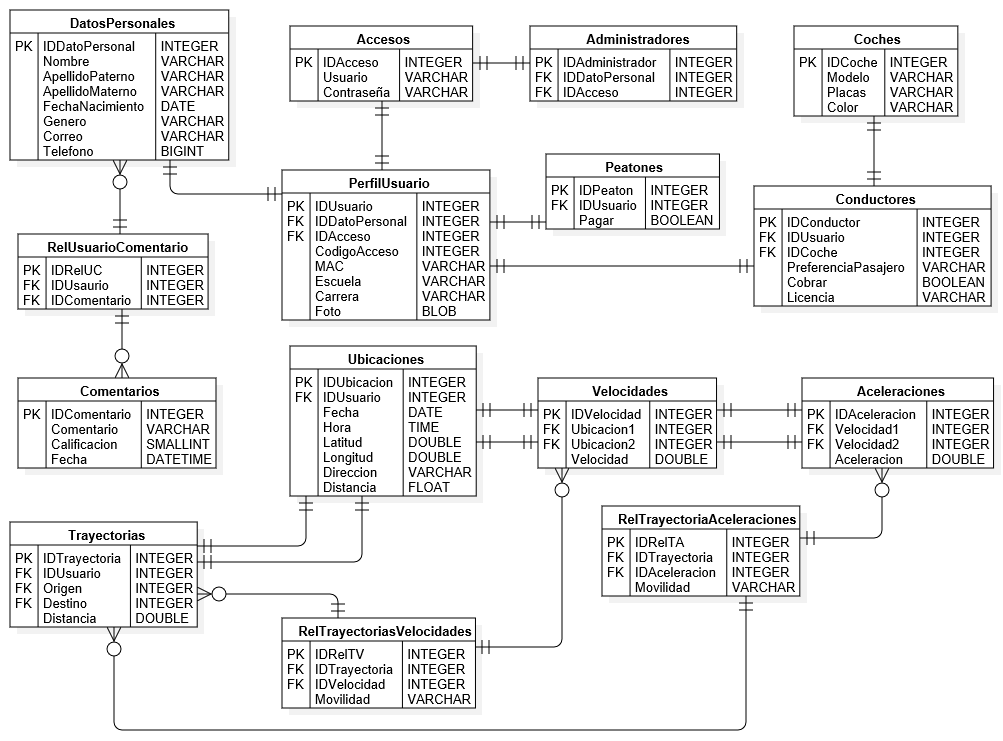
## **5.4 Base de datos del sistema propuesto**

En esta sección se presentan el diseño de la base de datos que conforma el sistema de recomendación de viajes compartidos propuesto en este proyecto terminal.

### **5.4.1 Diagrama de base de datos**

En esta sección se presenta el diagrama de clases del sistema de información propuesto en este proyecto terminal.

En la figura 32, se muestra el diagrama de base de datos del sistema propuesto



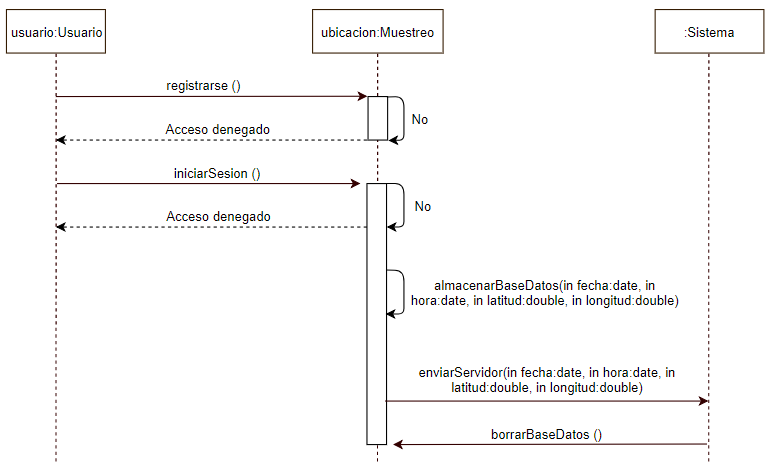
**Figura 32.** Diagrama de base de datos del sistema propuesto

## **5.5 Diagramas de secuencia del sistema propuesto**

En esta sección se presentan los diagramas de secuencia que conforman el sistema de recomendación de viajes compartidos propuesto en este trabajo terminal.

### **5.5.1 Diagrama de secuencia del algoritmo de muestreo**

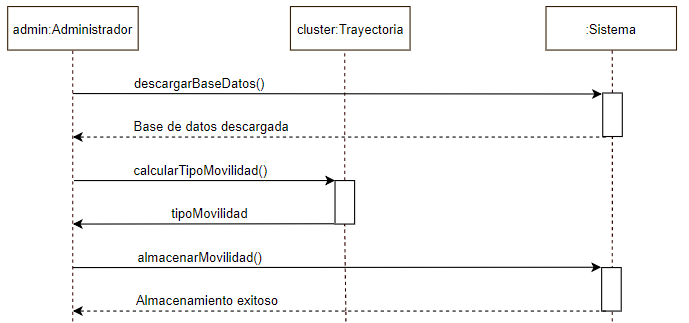
En esta sección se presenta en la figura 33 el diagrama de secuencia del algoritmo de muestreo.



**Figura 33.** Diagrama de secuencia del muestreo

### **5.5.2 Diagrama de secuencia del algoritmo de detección de patrones de movilidad individual**

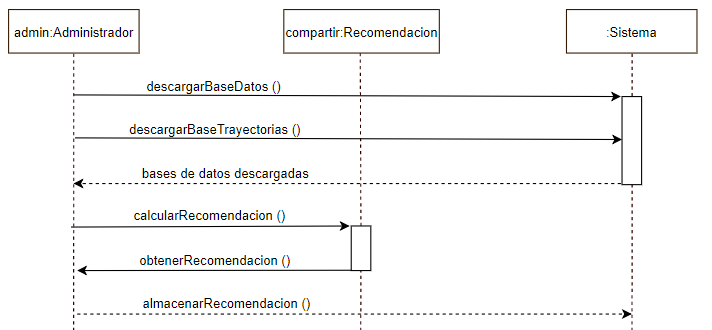
En esta sección se presenta en la figura 34 el diagrama de secuencia que describe la serie de pasos que se deben realizar para el algoritmo de detección de patrones de movilidad individual.



**Figura 34.** Diagrama de secuencia del algoritmo de detección de patrones de movilidad individual

### **5.5.3 Diagrama de secuencia del algoritmo de búsqueda de recomendaciones para compartir transporte**

En esta sección se presenta en la figura 35 el diagrama de secuencias del algoritmo de Búsqueda de recomendaciones de movilidad.



**Figura 35.** Diagrama de secuencia del algoritmo de búsqueda de recomendaciones para compartir transporte

## **5.6 Cálculo de KPIs y despliegue en el tablero de control**

En esta subsección se desarrolla el tablero de control para este proyecto terminal.

Para realizar la visualización de KPIs en el tablero de control, se emplearán las herramientas de Leaflet, JSP, CSS, JavaScript, Java y D3 para visualizar mapas, mapas de calor, líneas, círculos y graficas de pastel. Como datos de entrada se utilizarán las siguientes bases de datos:

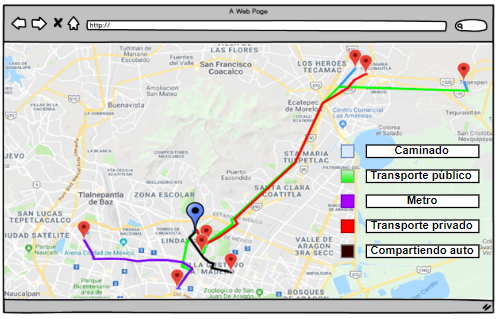
* **Base de datos del perfil de los usuarios** obtenida del registro de los usuarios de la aplicación móvil y almacenada en el servidor Web.
* **Base de datos de ubicación de los usuarios** obtenida a partir de la implementación del algoritmo de muestreo en los dispositivos móviles de los estudiantes de UPIITA.
* **Base de datos del tipo de movilidad** de cada usuario obtenida del algoritmo de patrones de movilidad.
* **Base de datos de viajes compartidos** obtenida del algoritmo de búsqueda de viajes compartidos.

A continuación, se mencionan los KPIs que serán desplegados en el tablero de control.

1. Número de usuarios que se han registrado en la aplicación móvil.
2. Número de usuarios que se registraron por escuela y carrera.
3. Número de usuarios de género masculino y femenino.
4. Número de usuarios peatones y conductores.
5. Rango de edades de los usuarios que se han registrado en la aplicación móvil.
6. Mapas de calor de las zonas con diferentes niveles de aglomeración que frecuentan los estudiantes. El grado de congestión de personas será desplegado con los siguientes colores:
   1. **Rojo:** Mucha aglomeración.
   2. **Amarillo:** Mucha aglomeración, menor a la tonalidad roja.
   3. **Verde lima:** Mucha aglomeración, menor a la tonalidad amarrilla.
   4. **Azul turquesa:** Mediana aglomeración, menor a la tonalidad verde lima.
   5. **Azul marino:** Poca aglomeración.
7. **Trayectorias origen destino** de los usuarios **limitado a un tiempo de muestreo** **indicado por el usuario en el tablero** **de control**.
8. Tipo de movilidad individual de los usuarios.
   1. **Azul oscuro:** caminando.
   2. **Verde oscuro:** medio de transporte.
   3. **Rojo:** detenido.
   4. **Amarillo:** Valor no considerado como un tipo de movilidad (*outlier)*.
9. Rutas candidatas a compartir automóvil.

En este trabajo se hace mención de trayectorias origen destino únicamente en las rutas sugeridas para compartir automóvil.

En la figura 36, se observa una propuesta del tablero de control, en donde cada uno de los colores indica algún tipo de movilidad, además de mostrar los lugares origen destino de los usuarios que llegan a la UPIITA (marcador de color azul):



**Figura 36.** Propuesta del tablero de control

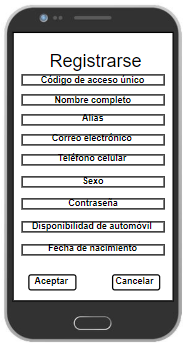
## **5.7 Diseño de vistas de la aplicación móvil**

En esta sección se bosqueja el diseño de las vistas para la aplicación móvil del sistema de información propuesto en este proyecto terminal.

En la figura 37, se muestra la vista principal de la aplicación móvil.

**Figura 37.** Vista principal

En la figura 38, se muestra la vista registrarse de la aplicación móvil.



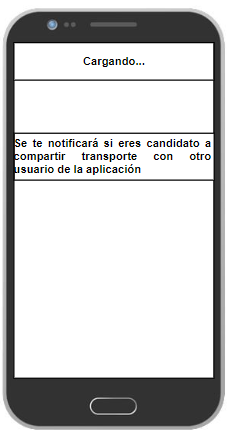
**Figura 38.** Vista registrarse

En la figura 39. Se muestra la vista iniciar sesión de la aplicación móvil.



**Figura 39.** Vista iniciar sesión

En la figura 40 se muestra la vista muestreo de la aplicación móvil.



**Figura 40.** Vista muestreo

# **6. Implementación**

En este capítulo se describe la implementación correspondiente de cada uno de los módulos del sistema, conformado por: aplicación móvil, servidor Web, procesamiento analítico y tablero de control. En la figura 41 se observa el esquema general y la secuencia de desarrollo del sistema:



**Figura 41.** Implementación del sistema para compartir automóvil

A continuación, se abordará de manera particular cada módulo implementado en el sistema.

## **6.1 Implementación de la aplicación móvil**

En esta sección se describe la implementación de la aplicación móvil.

### **6.1.1 Implementación de las pantallas de la aplicación móvil**

En esta subsección se presentan las pantallas realizadas para la aplicación móvil.

En la figura 42 se describe la pantalla **Portada** de la aplicación móvil.

|  |  |
| --- | --- |
|  | Al abrir la aplicación móvil, el usuario observará una pantalla con la descripción del nombre del sistema, llamado **Mobility Finder**, después de dos segundos, será redireccionado a la figura 43. |

**Figura 42.** Pantalla Portada de la aplicación móvil

En la figura 43 se describe la pantalla **Login** de la aplicación móvil, correspondiente al inicio de sesión.

|  |  |
| --- | --- |
|  | En esta pantalla se muestra el menú de autentificación del usuario. Si el usuario está registrado en el sistema, deberá ingresar su correo electrónico y contraseña presionando el botón .  El sistema validará la información ingresada, de ser correcta, será redireccionado a la figura 47.  Si el usuario no está registrado en el sistema deberá presionar el botón  para ser redireccionado a la figura 44. |

**Figura 43.** Pantalla Login de la aplicación móvil

En la figura 44 se describe la pantalla **Registro** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran los datos a ser ingresados por el usuario para ser registrado en el sistema. Las variables ingresadas son: nombre, apellido paterno, apellido materno, fecha de nacimiento, genero, correo, contraseña, teléfono, escuela y carrera.  Si el usuario selecciona la opción **pasajero**, será redireccionado a la figura 45. Si el usuario selecciona la opción **conductor**, será redireccionado a la figura 46. |

**Figura 44.** Pantalla Registro de la aplicación móvil

En la figura 45 se describe la pantalla **Pasajero** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran los datos de un usuario de tipo pasajero a ser ingresado. La variable ingresada es: ¿pagaría por el servicio?  Para terminar de registrarse, el usuario debe presionar el botón .  El usuario será redireccionado a la figura 43. |

**Figura 45.** Pantalla Pasajero de la aplicación móvil

En la figura 46 se describe la pantalla **Conductor** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran los datos de un usuario de tipo conductor a ser ingresados. Las variables son: ¿Cobraría el viaje?, marca de auto, modelo de auto, color del auto, número de placas, preferencia de pasajeros, número de licencia de conducir y número de asientos disponibles. Estas variables son cifradas para garantizar la privacidad de los datos del usuario conductor en el servidor Web.  Para terminar de registrarse, el usuario debe presionar el botón .  El usuario será redireccionado a la figura 43. |

**Figura 46.** Pantalla Conductor de la aplicación móvil

En la figura 47 se describe la pantalla **Muestreo** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran tres botones.  El botón  solicitará al usuario permisos para activar el GPS de su dispositivo móvil para posteriormente almacenar las coordenadas geográficas del usuario.  El botón  desactivará la opción de muestrear al usuario.  El botón  desplegará el siguiente submenú:    La opción **Perfil** redireccionará al usuario a la figura 48.  Las opciones **aceptar sugerencia**, **sugerencia Auto-Auto** y **sugerencia Auto-Peatón** redireccionan al usuario a un menú de sugerencias por día, como se muestra en la figura 49. |

**Figura 47.** Pantalla Muestreo de la aplicación móvil

En la figura 48 se describe la pantalla **Perfil** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran los datos ingresados por el usuario al sistema para registrarse.  Si el usuario presiona el botón  será redireccionado a la figura 47. |

**Figura 48.** Pantalla Perfil de la aplicación móvil

En la figura 49 se describe la pantalla **Menú Sugerencia por día** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran cinco botones, correspondientes a los días de la semana lunes, martes, miércoles, jueves y viernes. Los usuarios podrán seleccionar el día que aceptarán o visualizarán las sugerencias disponibles para compartir automóvil.  Si los usuarios seleccionaron la opción **Aceptar Sugerencia** de la figura 47, serán redireccionados a la figura 50.  Si los usuarios seleccionaron la opción **Sugerencia** **Auto-Auto** o **Sugerencia** **Auto-Peatón** de la figura 47 serán redireccionados a la figura 51 para visualizar sus sugerencias para compartir el automóvil disponible. |

**Figura 49.** Pantalla Menú Sugerencia por Día de la aplicación móvil

En la figura 50 se describe la pantalla **Aceptar Sugerencia** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestra las sugerencias disponibles para compartir automóvil entre usuarios conductor-conductor y conductor-peatón.  Los usuarios conductores sólo podrán aceptar por cada día un máximo de sugerencias menores o iguales a la cantidad de asientos disponibles que hayan registrado en su perfil.  Los usuarios peatones podrán seleccionar sugerencias posteriormente de que los usuarios conductores hayan aceptado primero.  La sugerencia para compartir automóvil será notificada al celular de los usuarios por medio de la red social WhatsApp. |

**Figura 50.** Pantalla Aceptar Sugerencia de la aplicación móvil

En la figura 51 se describe la pantalla **Menú Sugerencia** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran un menú acerca de los usuarios con los que podrá compartir automóvil, los datos que se desplegarán son los siguientes: fecha, día, hora de inicio, punto de inicio, hora de bajada, punto de bajada, posición peatón, distancia, tiempo, correo conductor, correo pasajero.  Si el usuario desea ver la ruta para compartir automóvil, deberá seleccionar el botón  para ser redireccionado a la figura 52. |

**Figura 51.** Pantalla Menú Sugerencia de la aplicación móvil

En la figura 52 se describe la pantalla **Mapa Sugerencia** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestran la ruta de sugerencia recomendada para los usuarios candidatos a compartir automóvil, la ruta muestra el punto más cercano del pasajero para abordar la ruta de conductor. |

**Figura 52.** Pantalla Mapa Sugerencia de la aplicación móvil

En la figura 53 se describe la pantalla **Notificación WhatsApp** de la aplicación móvil.

|  |  |
| --- | --- |
|  | En esta pantalla se muestra la notificación de sugerencia para compartir automóvil para los usuarios candidatos que aceptaron compartir su ruta con otros usuarios. |

**Figura 53.** Pantalla Notificación WhatsApp de la aplicación móvil

### **6.1.2 Implementación del muestreo en la aplicación móvil**

En esta subsección se aborda la codificación del muestreo para la aplicación móvil.

Para realizar el muestreo de la aplicación móvil, se desarrolló la codificación que se muestra en fragmento del código 1. En el primer paso, se configuró la solicitud de permisos para el uso del GPS y Wi-Fi en la aplicación móvil utilizando la función *ACCESS\_FINE\_LOCATION ()* y *ACCESS\_COARSE\_LOCATION ().* Posteriormente, utilizando un objeto de tipo LocationManager y LocationListener se solicitará la posición de las coordenadas geográficas utilizando las funciones *getLongitude ()* y *getLatitude ().* Para almacenar las ubicaciones de los usuarios de manera interna, se utiliza el método *addStudentDetail ().*

|  |
| --- |
| *//Verificar si tiene permisos lat y lon, GPS-> falta el de internet, Granted= si tiene permisos* **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***ACCESS\_FINE\_LOCATION***,Manifest.permission.***ACCESS\_COARSE\_LOCATION***,}, 1000); }  **protected void** onPreExecute(){   **mTextView**.setTextColor(Color.***BLUE***);  **mTextView**.setText(**mTextView**.getText() + **"\nInicializando muestreo o transferencia de datos"** +  **" al servidor Web, espere por favor .... "**);  *//Permiso de GPS Provider* LocationManager locationManager = (LocationManager) MainActivity.**this**.getSystemService(Context.***LOCATION\_SERVICE***);   **final** LocationListener locationListener = **new** LocationListener() {  *// Cuando cambie la ubicación* **public void** onLocationChanged(Location location) {**glon** = location.getLongitude();  **glat** = location.getLatitude();} *//onLocation*  };   **int** permissionCheck = ContextCompat.*checkSelfPermission*(MainActivity.**this**,  Manifest.permission.***ACCESS\_FINE\_LOCATION***);locationManager.requestLocationUpdates(LocationManager.***GPS\_PROVIDER***, 0, 0, locationListener); } *//on PreExecutive* |

**Fragmento del código 1.** Muestreo en la aplicación móvil

### **6.1.3 Implementación del almacenamiento interno en la aplicación móvil**

En esta subsección se bosqueja la codificación del almacenamiento interno para la aplicación móvil en el fragmento del código 2. Para realizar el almacenamiento interno de la ubicación de los usuarios se utiliza la función *addStudentDetail (),* donde se utilizan los parámetros de la fecha, hora, latitud y longitud muestreada del dispositivo móvil.

|  |
| --- |
| **public long** addStudentDetail(String fechaa, String horaa, **double** lat, **double** longi) {  SQLiteDatabase db = **this**.getWritableDatabase();ContentValues values = **new** ContentValues();  values.put(***FECHA***, fechaa);  values.put(***HORA***, horaa);  values.put(***LATITUD***, lat);  values.put(***LONGITUD***, longi);**long** insert = db.insert(***TABLA\_ESTUDIANTES***, **null**, values);  **return** insert; } |

**Fragmento del código 2.** Almacenamiento interno en la aplicación móvil

### **6.1.4 Implementación de la transferencia de datos al servidor Web de la aplicación móvil**

En esta subsección se aborda la codificación de la transferencia de datos de la ubicación de los usuarios para la aplicación móvil, obsérvese en el fragmente de código 3. En este programa se utilizó el lenguaje de programación Java para transferir el id del usuario, la fecha, la hora y las coordenadas geográficas de cada registro muestreado de la aplicación móvil utilizando una instrucción ***try-catch-finally*** para evitar posibles fallas al momento de transferir paquetes de datos, siendo estos los registros almacenados de manera interna en el dispositivo móvil. Además, se utilizaron los objetos ***HttpUrlConnection*** y la ruta del *script* empleado para realizar el almacenamiento de los datos en el gestor de bases de datos de MySQL en 000WebHost. La codificación utilizada para registrar a los usuarios al sistema informático es similar a este programa, por este motivo no se incluirá en el escrito.

|  |
| --- |
| BufferedReader reader = **null**; String servidor = **"http://olivares567.000webhostapp.com/sprueba4.php"**; String datosEnviados = **new** String();  **try** {datosEnviados = URLEncoder.*encode*(**"idU"**, **"UTF-8"**) + **"="** + URLEncoder.*encode*(**IDU**, **"UTF-8"**);  datosEnviados += **"&"** + URLEncoder.*encode*(**"lat"**, **"UTF-8"**) + **"="** +  URLEncoder.*encode*(String.*valueOf*(latStore), **"UTF-8"**);  datosEnviados += **"&"** + URLEncoder.*encode*(**"lon"**, **"UTF-8"**) + **"="** +  URLEncoder.*encode*(String.*valueOf*(lonStore), **"UTF-8"**);datosEnviados += **"&"** + URLEncoder.*encode*(**"fecha"**, **"UTF-8"**) + **"="** +  URLEncoder.*encode*(String.*valueOf*(fechaStore), **"UTF-8"**);  datosEnviados += **"&"** + URLEncoder.*encode*(**"hora"**, **"UTF-8"**) + **"="** +  URLEncoder.*encode*(String.*valueOf*(horaStore), **"UTF-8"**);  } **catch** (UnsupportedEncodingException e) { } **try** {  URL url = **new** URL(servidor);   HttpURLConnection con = (HttpURLConnection) url.openConnection();  con.setReadTimeout(60000);  con.setConnectTimeout(60000);  con.setRequestMethod(**"POST"**);  con.setInstanceFollowRedirects(**false**);   con.setDoOutput(**true**);  OutputStreamWriter writer = **new** OutputStreamWriter(  con.getOutputStream());  writer.write(datosEnviados);  writer.flush();   InputStream is = con.getInputStream();  reader = **new** BufferedReader(**new** InputStreamReader(is, **"UTF-8"**));  String data = **null**;  StringBuilder sb = **new** StringBuilder();  **while** ((data = reader.readLine()) != **null**) {  sb.append(data);  }} **catch** (Exception e) {  e.printStackTrace();} **finally** { **if** (reader != **null**) {  *// COMENTAR ESTO* **try** {  *//dbConnector.deleteTitle();* reader.close();**dbConnector**.delete();  numRegistros = **dbConnector**.CountRegisterData();  i = 0;} **catch** (IOException e) {  e.printStackTrace();} *//catch* } *// if reader !=* } *//finally* |

**Fragmento del código 3.** Transferencia de datos de la aplicación móvil al servidor Web

Para la conexión con el servidor Web, se han utilizado *scripts* desarrollados con el lenguaje de programación PHP, los cuales se mencionarán en la siguiente subsección.

### **6.1.5 Implementación de la sugerencia para compartir automóvil en la aplicación móvil**

En esta subsección se muestra la codificación de la sugerencia para compartir automóvil de la aplicación móvil en el fragmento del código 4.

|  |
| --- |
| **public class** MapActivity **extends** AppCompatActivity **implements** OnMapReadyCallback, TaskLoadedCallback {  **private** GoogleMap **mMap**;  **private** MarkerOptions **place0**, **place1**, **place2**;  Button **getDirection**;  **private** Polyline **currentPolyline**;**double latSubida** = Double.*parseDouble*( *datos*.**sLatEncuentro**.get(*datos*.**sIdMapa**) );  **double lonSubida** = Double.*parseDouble*( *datos*.**sLontEncuentro**.get(*datos*.**sIdMapa**) );  **double latBajada** = Double.*parseDouble*( *datos*.**sLatBajada**.get(*datos*.**sIdMapa**) );  **double lonBajada** = Double.*parseDouble*( *datos*.**sLonBajada**.get(*datos*.**sIdMapa**) );  **double latPeaton** = Double.*parseDouble*( *datos*.**sLatPeaton**.get(*datos*.**sIdMapa**) );  **double lonPeaton** = Double.*parseDouble*( *datos*.**sLonPeaton**.get(*datos*.**sIdMapa**) ); @Override  **protected void** onCreate(@Nullable Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***map\_activity***);  **getDirection** = findViewById(R.id.***btnGetDirection***); **final** LatLng centrarInicio = **new** LatLng(**latSubida**,**lonSubida**);   *//FUNCION DEL BOTON CADA VEZ QUE ES LLAMADO* **getDirection**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View view) {  **new** FetchURL(MapActivity.**this**).execute(getUrl(**place1**.getPosition(), **place2**.getPosition(), **"driving"**), **"driving"**);**mMap**.animateCamera(CameraUpdateFactory.*newLatLngZoom*(centrarInicio, 15) );  }  });**place0** = **new** MarkerOptions().position(**new** LatLng(**latPeaton**, **lonPeaton**)).title(**"Posición Peatón"**);**place1** = **new** MarkerOptions().position(**new** LatLng(**latSubida**, **lonSubida**)).title(**"Punto de subida"**);**place2** = **new** MarkerOptions().position(**new** LatLng(**latBajada**, **lonBajada**)).title(**"Punto de bajada"**);  MapFragment mapFragment = (MapFragment) getFragmentManager()  .findFragmentById(R.id.***mapNearBy***);  mapFragment.getMapAsync(**this**);  }  } |

**Fragmento del código 4.** Visualización del mapa de sugerencia para compartir automóvil en la aplicación móvil

Para mostrar la sugerencia de compartir automóvil en la aplicación móvil se ha utilizado la clase llamada FetchURL, disponible en la documentación de la API de Google, permite asignar la posición inicial y final de la ruta de tipo **conductor** (*driving*) a ser trazada. Obsérvese en el fragmento del código 4 línea 26.

|  |
| --- |
| FetchURL(MapActivity.**this**).execute(getUrl(**place1**.getPosition(), **place2**.getPosition(), **"driving"**), **"driving"**); |

La asignación de los marcadores que muestran el punto de subida y bajada de los usuarios que comparten automóvil, así como el punto más cercano del usuario pasajero para abordar con el usuario conductor se realiza utilizando el fragmento del código 4 línea 32.

|  |
| --- |
| **place1** = **new** MarkerOptions().position(**new** LatLng(**latSubida**, **lonSubida**)).title(**"Punto de subida"**); |

### **6.1.6 Implementación del esquema de seguridad en la aplicación móvil**

En esta subsección se expone la codificación del algoritmo AES-256 implementado en la autentificación de la aplicación móvil como esquema de seguridad. Obsérvese en el fragmento del código5.

Para generar una clave encriptada se ha utilizado la función*AESGenerateKey ()* y para encriptar la información se implementó la función *AESencrypt ().* Para desencriptar el texto cifrado, fue utilizada la función *AESdecrypt ().*

|  |
| --- |
| String passwordEncriptacion = **"contrasena"**;  **private** String encriptar(String datos, String password) **throws** Exception{  SecretKeySpec secretKey = generateKey(password);  Cipher cipher = Cipher.*getInstance*(**"AES"**);  cipher.init(Cipher.***ENCRYPT\_MODE***, secretKey);  **byte**[] datosEncriptadosBytes = cipher.doFinal(datos.getBytes());  String datosEncriptadosString = Base64.*encodeToString*(datosEncriptadosBytes, Base64.***DEFAULT***);  **return** datosEncriptadosString;}  **private** String desencriptar(String datos, String password) **throws** Exception{  SecretKeySpec secretKey = generateKey(password);  Cipher cipher = Cipher.*getInstance*(**"AES"**);  cipher.init(Cipher.***DECRYPT\_MODE***, secretKey);  **byte**[] datosDescoficados = Base64.*decode*(datos, Base64.***DEFAULT***);  **byte**[] datosDesencriptadosByte = cipher.doFinal(datosDescoficados);  String datosDesencriptadosString = **new** String(datosDesencriptadosByte);  **return** datosDesencriptadosString;}  **private** SecretKeySpec generateKey(String password) **throws** Exception{  MessageDigest sha = MessageDigest.*getInstance*(**"SHA-256"**);  **byte**[] key = password.getBytes(**"UTF-8"**);  key = sha.digest(key);  SecretKeySpec secretKey = **new** SecretKeySpec(key, **"AES"**);  **return** secretKey; } |

**Fragmento del código 5.** Cifrado de la aplicación móvil

## **6.2 Implementación del servidor Web**

En esta sección se describe la implementación del servidor Web utilizado en el sistema de información.

### **6.2.1 Implementación de la obtención de los datos del muestreo en el servidor Web**

En esta subsección se presenta la codificación de la obtención de los datos de las ubicaciones de los usuarios correspondientes a las variables **$idu** (id del Usuario)**, $fecha** (fecha muestreada)**, $hora** (hora muestreada)**, $lat** (latitud) **y $lon** (longitud) utilizando un script de PHP en el servidor Web. Obsérvese en el fragmento del código6.

|  |
| --- |
| <?php  $con=mysqli\_connect("localhost","id8412411\_db\_username","upiita","id8412411\_db");  // Si existe un error en la conexión con la base de datos, enviar error-1, de lo contrario, continuar la ejecución  if (mysqli\_connect\_errno($con)) {  echo "error-1";  }else {  $idU = $\_POST["idU"];  $lat = $\_POST["lat"];  $lon = $\_POST["lon"];  $fecha = $\_POST["fecha"];  $hora = $\_POST["hora"];  $comando = "INSERT INTO ubicacion (idU,fecha,hora,lat,lon)  VALUES ('$idU','$fecha','$hora','$lat','$lon')";  mysqli\_close($con);  }  ?> |

**Fragmento del código 6.** Almacenamiento de las ubicaciones de los usuarios en el Servidor Web

### **6.2.2 Implementación de la sugerencia para compartir automóvil en el servidor Web a la aplicación móvil**

En esta subsección se presenta la codificación para transferir las sugerencias de compartir automóvil del servidor Web a la aplicación móvil utilizando un script de PHP. En este programa se verifica el correo del usuario candidato a compartir automóvil para posteriormente enviar la información y la ruta de sugerencia a la aplicación móvil del usuario. Obsérvese en el fragmento del código7.

|  |
| --- |
| <?php  $con=mysqli\_connect("localhost","id8412411\_db\_username","upiita","id8412411\_db");  if (mysqli\_connect\_errno($con)) {  echo "error-1"; // NO HAY ACCESO  }else {  date\_default\_timezone\_set('America/Mexico\_City');  $hoy = date('Y-m-d H:i:s');  $correo = $\_POST["correo"];  $resultado = mysqli\_query($con, "select \* FROM sugerencia where CorreoConductor='$correo'" );    if($resultado) {  $datas = array();  while($row = mysqli\_fetch\_assoc($resultado)){  $datas[] = $row;  }  $contarR=count($datas);  //Concatenar array  $aux;    for ($i = 0; $i < $contarR; $i++) {  $aux=implode("!", $datas[$i]);  $vectorAux=$vectorFinal;  $vectorFinal=$vectorAux."!".$aux;  echo $vectorFinal."<br/>"."<br/>";  }    if ($contarR>=1){  $datos = $hoy . "#" . "1". "#" . $vectorFinal;  }else{  $datos = "no";  }  }  else {  $datos = "no"; // ERROR EN CONSULTA  }  mysqli\_close($con);  echo $datos;  } |

**Fragmento del código 7.** Transferencia de la sugerencia para compartir automóvil a la aplicación móvil

## **6.3 Implementación del procesamiento analítico**

En esta sección se describen la implementación del módulo de procesamiento analítico desarrollado en el sistema de información para compartir automóvil, conformado por las subsecciones: descarga de la base de datos, limpieza de datos, detección del patrón de movilidad individual, algoritmo de viajes compartidos y transferencia de datos al servidor Web.

### **6.3.1 Implementación de la descarga de la base de datos**

Para realizar la implementación del módulo de procesamiento analítico, se requiere descargar la tabla **ubicación** y **perfil**, las cuales se encuentran alojadas en el servidor Web, utilizando los comandos del fragmento de código 8, se precede a descargar las tablas en una computadora local.

|  |
| --- |
| cd C:\Program Files\MySQL\MySQL Server 5.7\bin  mysqldump -u u -h 213.190.6.169 -p u418198429\_gyzy perfil > C:/Users/edgar/Desktop/ExamenPT2/ProgramasPT2/perfil.sql  mysqldump -u u -h 213.190.6.169 -p u418198429\_gyzy ubicacion > C:/Users/edgar/Desktop/ExamenPT2/ProgramasPT2/ubicacion.sql |

**Fragmento del código 8.** Descargar la base de datos del servidor Web

### **6.3.2 Implementación de la limpieza de datos**

En esta subsección se muestra el código utilizado para eliminar los datos repetidos y coordenadas geográficas iguales a cero, como se ha redactado en el análisis de la subsección 5.1.2. Obsérvese en el fragmento del código9.

|  |
| --- |
| **for**(**int** j=0 ;j<=totalUsuarios; j++){ *//POR USUARIO* **for**(**int** k=0; k<=fechasinDuplicados.size()-1;k++) {  hayPrimero=**false**;  **try** {  archivo = **new** BufferedReader(**new** FileReader(nombreArchivo));  *//Leer hasta que ya no haya datos* **while** ((registro = archivo.readLine()) != **null**) {  *//StringTokenizer divide un string en substrings o tokens* StringTokenizer dato = **new** StringTokenizer(registro, **","**);  *//Lugar aumentar un dato más para descripción del lugar* String dato1 = dato.nextToken();*//idU del usuario  //fecha=dato.nextToken();* String dato2 = dato.nextToken();*//fecha* String dato3 = dato.nextToken();*//hora* String dato4 = dato.nextToken();*//latitud* String dato5 = dato.nextToken();*//longitud* lat = Double.*parseDouble*(dato4);  lon = Double.*parseDouble*(dato5);  *//Si son el mismo usuario* **if** (*parseInt*(dato1)==j){**if**((fechasinDuplicados.get(k)).equals(dato2)){  *//Limpieza de datos* **if** (hayPrimero == **false**) {  *//Si existen coordenadas erroneas* **if**(lat==0 && lon==0) {  antLat = lat;  antLon = lon;} *//continue;* **else** { hayPrimero = **true**;  *//si el primer dato no es 0, agregalo :V* **if**(j==0){  *//Existe und ato con el usuario 0* }**else**{ubicacion.get(0).add(dato1); *//IdU* ubicacion.get(1).add( dato2.replace(**" d. C."**,**""**) ); *//Fecha* ubicacion.get(2).add( dato3.replace(**" CDT"**,**""**) ); *//Hora* ubicacion.get(3).add(dato4); *//Lat* ubicacion.get(4).add(dato5); *//Lon* }  antLat = lat;  antLon = lon;  }   } **else** {  **if**( (lat!=antLat && lon!=antLon) && (lat!=0 && lon!=0) ){ **if**(j==0){}**else**{ubicacion.get(0).add(dato1); *//IdU* ubicacion.get(1).add( dato2.replace(**" d. C."**,**""**) ); *//Fecha* ubicacion.get(2).add( dato3.replace(**" CDT"**,**""**) ); *//Hora* ubicacion.get(3).add(dato4); *//Lat* ubicacion.get(4).add(dato5); *//Lon* }  }  antLat = lat;  antLon = lon;  }*//else* }*//fecha* **else**{}  } *//if son el mismo usuario* **else**{ } *//else son el mismo usuario* } *// while* } **catch** (FileNotFoundException e) {  System.***out***.println(**"No existe el archivo"**);  } **catch** (IOException e) {  System.***out***.println(**"Error de entrada/salida de datos"**);  } **finally** { *//son tareas que siempre ocurren, Cerrar bien el archivo es importante* **if** (archivo != **null**) {  **try** {  archivo.close(); *//cerrar el archivo* } **catch** (IOException e) {  System.***out***.println(**"Error al cerrar el archivo"**);  }*//catch* }*//if* }*//finally* } *//* } *//for para n usuarios* |

**Fragmento del código 9.** Limpieza de datos de la base de datos ubicación

En este algoritmo, se genera la base de datos **ubicacionLimpia** con formato CSV, siendo este utilizado posteriormente en las subsecciones 6.3.3 y 6.3.4, así como la sección 6.4.

### **6.3.3 Implementación del algoritmo de detección de patrones de movilidad individual**

En esta subsección se ha codificado el algoritmo diseñado en la subsección 5.1.2, para su realización, se ha utilizado el lenguaje de programación Scala, el marco de trabajo Spark y el gestor de bases de datos Cassandra. En el fragmento del código 10 se muestra el código de la fórmula de Haversine.

|  |
| --- |
| **def** distanciaCoord(lat1: Double, lng1: Double, lat2: Double, lng2: Double): Double = {  *//double radioTierra = 3958.75;//en millas* **val** radioTierra: Double = 6371 *//en kilómetros - radio equivolumen* **val** dLat: Double = Math.*toRadians*(lat2 - lat1) *// grado a radianes* **val** dLng: Double = Math.*toRadians*(lng2 - lng1)  **val** sindLat: Double = Math.*sin*(dLat / 2.0)  **val** sindLng: Double = Math.*sin*(dLng / 2.0)  **val** va1: Double = Math.*pow*(sindLat, 2) +  Math.*pow*(sindLng, 2) \* Math.*cos*(Math.*toRadians*(lat1)) \* Math.*cos*(Math.*toRadians*(lat2))  **val** va2: Double = 2 \* Math.*atan2*(Math.*sqrt*(va1), Math.*sqrt*(1 - va1))  **val** distancia: Double = radioTierra \* va2  distancia \* 1000 *//metros* } *//distanciaCoord* |

**Fragmento del código 10.** Fórmula de Haversine utilizando Scala

Para conectar el manejador de bases de datos Cassandra y el marco de trabajo de Spark, se ha utilizado la configuración de **SparkContext**, la cual permite crear las variables que utiliza este marco de trabajo. Obsérvese en el fragmento del código 11 línea 1.

|  |
| --- |
| **val** conf = **new** SparkConf().setMaster(**"local[\*]"**).setAppName(**"SparkCassandra"**)  .set(**"spark.cassandra.connection.host"**, **"localhost"**)  **val** sc = **new** SparkContext(conf) |

Posteriormente, se han utilizado el tipo de dato RDD (por sus siglas en inglés *Resilient Distributed Dataset* o Conjunto de datos Distribuidos Resilientes) para declarar cualquier tipo de variable (String, double, int). Además, permite la lectura de datos de las tablas de Cassandra de una manera sencilla. Obsérvese en el fragmento del código 11 línea 22.

|  |
| --- |
| **var** hora = sc.cassandraTable[Double](**"usuario"**, **"ubicacion"**).select(**"hora"**).where(**"idU = "** + i).collect() |

En el fragmento del código 11, se muestra el código del algoritmo de detección del patrón de movilidad del usuario. De este programa, se obtiene como salida la base de datos **movilidad** con formato CSV, siendo este utilizado posteriormente en la sección 6.4.

|  |
| --- |
| **val** conf = **new** SparkConf().setMaster(**"local[\*]"**).setAppName(**"SparkCassandra"**)  .set(**"spark.cassandra.connection.host"**, **"localhost"**)  **val** sc = **new** SparkContext(conf)  **var** antLat, antLon, distancia, acumuladaD, auxD, velocidad: Double = 0  **var** hayPrimero, hayPrimerV: Boolean = **false  var** T : ArrayList[Double] = **new** ArrayList[Double]() *// Diferencia de tiempo* **var** D : ArrayList[Double] = **new** ArrayList[Double]() *// Distancia* **var** V : ArrayList[Double] = **new** ArrayList[Double]() *// Velocidad* **var** TM : ArrayList[Int] = **new** ArrayList[Int]() *// Tipo de movilidad del usuario* **var** antHora : Double = 0  **var** j : Int =0 *//Esta variable recorre todos los vectores del RDD lat,lon, hora* **var** numFechasContador :Int = 0  **var** m : Int = 0  **var** contadorRenglon=1 *// CONTADOR PARA REGISTRAR # DE CADA DATOS EN CASSAND   // --- NÚMERO DE USUARIOS --- //* **var** numUsuarios = sc.cassandraTable[Int](**"usuario"**, **"perfil"**).select(**"idU"**).cassandraCount()  *println*(**"Número total de usuario: "**+ numUsuarios)   *// --- NÚMERO DE REGISTROS POR USUARIO --- //* **for** (i <- 1 to numUsuarios.toInt) {  **var** NumRegPorU = sc.cassandraTable[Int](**"usuario"**, **"ubicacion"**).select(**"idU"**).where(**"idU = "** + i).cassandraCount()  **var** contSubirCass=0 *// SOLO APLICA EN HORA Y FECHA POR USUARIO*  *// --- REGISTROS "FECHA" DE CADA USUARIO --- //* **val** totalRegFechaU = sc.cassandraTable[String](**"usuario"**, **"ubicacion"**).select(**"fecha"**).where(**"idU = "** + i).collect()**var** totalFechaDif = totalRegFechaU.distinct  *// --- TAMAÑO DE REGISTROS DIFERENTES "FECHA" POR USUARIOS ---//* **var** numTotalFechaDif = totalFechaDif.length   numFechasContador = 0 *// Inicializar nuevo contador de Filtrado fecha* j = 0 *// Inicializar nuevo contador para los vectores lat, lon hora* **var** lat = sc.cassandraTable[Double](**"usuario"**, **"ubicacion"**).select(**"lat"**).where(**"idU = "** + i).collect()  **var** lon = sc.cassandraTable[Double](**"usuario"**, **"ubicacion"**).select(**"lon"**).where(**"idU = "** + i).collect()  **var** hora = sc.cassandraTable[Double](**"usuario"**, **"ubicacion"**).select(**"hora"**).where(**"idU = "** + i).collect()   **for**(ii <- 1 to numTotalFechaDif ){**var** filFecha = totalRegFechaU.filter(x => { x == totalFechaDif( numFechasContador ) }) *//voy a tomar la fecha que se encuentre actualmente en ese vector* **var** NumFechaFil = filFecha.length  **for** (iii <- 1 to NumFechaFil) { *//voy a recorrer todos los registros de la fecha* **if**(hayPrimero == **false**){ *//Primer registro por cada día nuevo* T.add(0); D.add(0); V.add(0); TM.add(0) *// No hubo 1era movilidad* antLat = lat (j)  *//AQUI DEJA DE FUNCIONAR EL PROGRAMA* antLon = lon (j)  *println*(**"1"**)  hayPrimero = **true** } *//inicialización primer dato* **else** {  distancia = *distanciaCoord*(antLat, antLon, lat(j), lon(j))  acumuladaD = acumuladaD + distancia  D.add(distancia)  **if** (hayPrimerV == **false**) {  **if** (antHora > hora(j)) {  **var** auxhora = hora(j) + 8.6400 \* 10000000000000L  velocidad = (distancia / (auxhora / 1000000000 - antHora / 1000000000)) \* 3.6  } **else** {  velocidad = (distancia / (hora(j) / 1000000000 - antHora / 1000000000)) \* 3.6  }  hayPrimerV = **true** } *// if hayPrimerV* **else** {  **if** (antHora > hora(j)) {  **var** auxhora = hora(j) + 8.6400 \* 10000000000000L velocidad = ((acumuladaD - auxD) / (auxhora / 1000000000 - antHora / 1000000000)) \* 3.6  } **else** { velocidad = ((acumuladaD - auxD) / (hora(j) / 1000000000 - antHora / 1000000000)) \* 3.6  }  } *// else hayPrimerV = True   // --- Condiciones Outliers --- //* **if** ( (hora(j)/1000000000 - antHora/1000000000 ) > 300 ||  distancia > 2166 || abs(acumuladaD - auxD )> 2166 ) { TM.add(10) }**else if** (velocidad >= 0 && velocidad < 0.5) { TM.add(0)}**else if** (velocidad >= 0.5 && velocidad < 10){ TM.add(1) }**else if** (velocidad >= 10 && velocidad < 130){ TM.add(2) }**else if** (velocidad>= 130 ||  distancia > 2166 ||  (acumuladaD - auxD )> 2166){ TM.add(10) } **else** { } V.add(velocidad)  T.add( hora(j)/1000000000 - antHora/1000000000 )  antLat= lat(j)  antLon= lon(j)  auxD = acumuladaD   } *// else de hayPrimero = True* antHora = hora(j)  j=j+1 *//Recorrer todos los registros   //LIMPIAR VARIABLES --- Cuando termine este ciclo* **if**(iii == NumFechaFil ){ *// --- ALMACENAMIENTO EN CASSANDRA ---- //* **if**(NumFechaFil == 1){ *// Si solo existe un registro, no hay D,T,V* **val** insertarCassandra = sc.parallelize(  *Seq*((i, contadorRenglon, totalFechaDif(ii-1), hora(contSubirCass), lat(0), lon(0), D.get(0), T.get(0), V.get(0), TM.get(0) ) ) )  insertarCassandra.saveToCassandra(**"usuario"**, **"movilidad"**,  *SomeColumns*(**"id"**, **"renglon"**, **"fecha"**, **"hora"**, **"lat"**, **"lon"**, **"d"**, **"t"**, **"v"**, **"tm"**))  contadorRenglon += 1  contSubirCass += 1  }**else**{ **for**(iv <- 1 to NumFechaFil ) {  **val** insertarCassandra = sc.parallelize( *Seq*((i, contadorRenglon, totalFechaDif(ii-1), hora(contSubirCass), lat(contSubirCass),lon(contSubirCass), D.get(iv-1), abs( T.get(iv-1) ), V.get(iv-1), TM.get(iv-1))))  insertarCassandra.saveToCassandra(**"usuario"**, **"movilidad"**,  *SomeColumns*(**"id"**, **"renglon"**, **"fecha"**, **"hora"**, **"lat"**, **"lon"**, **"d"**, **"t"**, **"v"**, **"tm"**))  contadorRenglon += 1  contSubirCass += 1  } *//For ingresar todos los registros*  }*// else para almacenar datos en Cassandra* hayPrimero = **false**; hayPrimerV = **false** *//lIMPIAR UN ARRAYLIST* antHora = 0; antLat = 0; antLon = 0  TM.clear(); V.clear() ;T.clear() ;D.clear()  } *// if ya se terminó el for* } *//FOR recorrer todas las fechas de cada usuario* numFechasContador= numFechasContador+1 *// Aumentar el siguiente dato filtrado*  } *// FOR # de registros diferentes* } *// FOR Número de usuarios*  } |

**Fragmento del código 11.** Algoritmo de detección de patrones de movilidad individual utilizando Scala

### **6.3.3 Implementación del algoritmo de viajes compartidos**

En esta subsección se ha codificado el algoritmo diseñado en la subsección 5.1.3, para su realización, se ha utilizado el lenguaje de programación Scala, el marco de trabajo Spark y el gestor de bases de datos Cassandra. Obsérvese en el fragmento de código 12. Además, en este algoritmo se utiliza el fragmento del código 10.

Este algoritmo está diseñado para realizar sugerencias de compartir automóvil con otros usuarios a partir del análisis de sus rutas individuales diarias durante un periodo de tiempo (datos recabados durante un mes).

A continuación, se enlista el funcionamiento del algoritmo para compartir automóvil.

1. Por cada usuario, el algoritmo evalúa las rutas individuales de cada día de la semana durante un periodo de tiempo (en este caso se utiliza un mes), teniendo como resultado, una lista de las rutas cotidianas de cada usuario correspondientes a cada día de la semana.
   1. Caso de uso: Si en un mes un usuario, se ha muestreado mínimo tres semanas, donde las rutas de un mismo día de cada semana son similares en espacio y tiempo, entonces el algoritmo considera que es una ruta habitual de ese día de la semana del usuario.
2. El algoritmo comparara en espacio y tiempo las rutas habituales entre usuarios conductores con peatones y posteriormente conductores con conductores. Aquellas rutas que coinciden se usan para hacer una comparación entre los datos de tiempo para buscar coincidencias considerando el umbral definido de media hora.

|  |
| --- |
| **def** haversine2 (valoresc : Array[Tuple4[Double, Double,Double,Double]], valoresp : Array[Tuple4[Double,Double, Double,Double]]) : Array[Double] ={ *//regresa renglones* **var** ventiempo: Double= 1.8E12  **var** yact =0  **var** banderay = 0  **var** banderat = 0  **val** radioTierra: Double = 6371  **var** arregloauxp: Array[Double] = **new** Array[Double](valoresp.length)  **var** arregloresp: Array[Double] = **new** Array[Double](valoresp.length)  **var** filvaloresc = valoresc  **var** bandera\_acortarc = 0  **var** valueando = *Array* (0,1,2,3)   **for**(x <- 0 until ((valoresc.length)-1)){ *//este for reduce los puntos que no estén lejos de 170m* **var** dlat= (valoresc(bandera\_acortarc).\_2 - valoresc(x+1).\_2)\*(*Pi*/180)  **var** dlon = (valoresc(bandera\_acortarc).\_3 - valoresc(x+1).\_3)\*(*Pi*/180)  **var** ha =(Math.*pow*(Math.*sin*(dlat/2),2) + Math.*cos*(valoresc(bandera\_acortarc).\_2)\*Math.*cos*(valoresc(x+1).\_2)\*Math.*pow*(Math.*sin*(dlon/2),2))  **var** hav = 2 \* Math.*asin*(Math.*min*(1.0, Math.*sqrt*(ha)))  **var** haver = radioTierra\*hav\*1000 *//metros*   **if**(haver <170){ *// 10km/h si la distancia es menor a esa, entonces el conductor no actuaba como tal* filvaloresc = filvaloresc.filter(!\_.==(valoresc(x+1)))}**else**{  bandera\_acortarc=x+1  }  }  **var** arregloauxc: Array[Double] = **new** Array[Double](filvaloresc.length)  **var** arregloresc: Array[Double] = **new** Array[Double](filvaloresc.length)  **var** lxx : Array[Double] = **new** Array[Double](filvaloresc.length)   **for** (x <- 0 until filvaloresc.length ){ *//for del tamaño de la tupla del conductor* **var** puntitoc= filvaloresc(x).\_1  breakable{ **for**(y<- yact until valoresp.length){ *// for del tamaño de la tupla del peaton* **var** puntitop= valoresp(y).\_1  **var** anal =filvaloresc(x).\_2  **var** anal2 = valoresp(y).\_2  **var** anal1=filvaloresc(x).\_3  **var** anal11=valoresp(y).\_3  **val** dlat : Double = (filvaloresc(x).\_2 - valoresp(y).\_2)\*(*Pi*/180)  **val** dlon : Double = (filvaloresc(x).\_3 - valoresp(y).\_3)\*(*Pi*/180)  **var** ha =(Math.*pow*(Math.*sin*(dlat/2),2) + Math.*cos*(filvaloresc(x).\_2)\*Math.*cos*(valoresp(y).\_2)\*Math.*pow*(Math.*sin*(dlon/2),2))  **var** hav = 2 \* Math.*asin*(Math.*min*(1.0, Math.*sqrt*(ha)))  **var** haver = radioTierra\*hav\*1000 *//metros*   **if**(haver < 2000) {  *//arregloaux2(y)=haver //esta guardando el resultado, pero quiero guardar los puntos* arregloauxp(y) = puntitop *//guarda el punto de coincidencia del peaton en el arreglo aux del peaton -del renglon* arregloauxc(x) = puntitoc *//guarda el punto de coincidencia del conductor en el arreglo aux del conductor* lxx(x) = x *//se guarda la posicion de x en un array, para luego evaluar más adelante* banderay = 1  **if** ((filvaloresc(x).\_4 - valoresp(y).\_4).abs < ventiempo) {  banderat = 1  } }**else if**(haver >=2000 && banderay ==1){  *//debe romperse y seguir evaluando en la "y del for" donde se quedó y banderay se vuelve 0 para que siga con la ruta del conductor* yact=y  banderay=0 arregloresp= arregloauxp.distinct *//siempre tendrán un valor porque banderay está en 1* arregloresc= arregloauxc.distinct  break()  }**else**{  }  }*//for y* }*//breakable* arregloresp = (arregloresp ++ arregloauxp).distinct *// renglones que tienen una ruta* arregloresc = (arregloresc ++ arregloauxc).distinct  lxx }*//for x* lxx = lxx.distinct  *// lxx = lxx.filter(\_>0)//.distinct //arregloresc y lxx, tienen el mismo tamaño, uno conserva el renglón y el otro el número de la iteración del loop* arregloresc = arregloresc.filter(\_> 0) *//elimina el 0, se tiene* **var** sumaruta : Double = 0.0 *//para saber el valor de la ruta* **var** quiene =lxx(lxx.length-1)  **for**(x <- 0 until ((lxx.length)-1)){ **var** auxi = lxx(x).toInt  **var** auxii = lxx(x+1).toInt  **var** dlat= (filvaloresc(auxi).\_2 - filvaloresc(auxii).\_2)\*(*Pi*/180)  **var** dlon = (filvaloresc(auxi).\_3 - filvaloresc(auxii).\_3)\*(*Pi*/180)  **var** ha =(Math.*pow*(Math.*sin*(dlat/2),2) + Math.*cos*(filvaloresc(auxi).\_2)\*Math.*cos*(filvaloresc(auxii).\_2)\*Math.*pow*(Math.*sin*(dlon/2),2))  **var** hav = 2 \* Math.*asin*(Math.*min*(1.0, Math.*sqrt*(ha)))  **var** haver = radioTierra\*hav\*1000 *//metros* sumaruta = sumaruta + haver  }   **if**(sumaruta >2000 && banderat==1){ *//si la ruta es mayor a 2km, entonces hay una ruta similar en espacio y si cumple con la bandera de tiempo, entonces hay ruta en espacio y tiempo  valruta*=sumaruta  **return** arregloresc *//regresamos los renglones de los datos del usuario conductor* }**else**{  **return null** *//no regresa nada porque no cumplió con la ruta* } } |

**Fragmento del código 12.** Algoritmo para sugerir viajes compartidos utilizando Scala

De esta codificación, se obtiene como salida la base de datos **movilidad** con formato CSV. Posteriormente, será utilizado en la sección 6.3.4 y 6.4.

### **6.3.4 Implementación de la transferencia de datos al servidor Web**

En esta subsección se ha codificado la transferencia de la base de datos **sugerencia**, obtenida del algoritmo de viajes compartidos para ser almacenada en el servidor Web, con el objetivo de que los usuarios candidatos a compartir automóvil visualicen las rutas que comparten con otros usuarios. Obsérvese en el fragmento del código 13.

|  |
| --- |
| BufferedReader reader = **null**;  String servidor = **"http://olivares567.000webhostapp.com/subirSugerencia.php"**; String datosEnviados = **new** String();  **try** {  datosEnviados = URLEncoder.*encode*(**"subirDatos"**, **"UTF-8"**) + **"="** +  URLEncoder.*encode*( subirDatos,**"UTF-8"**);} **catch** (UnsupportedEncodingException e) { } **try** {  URL url = **new** URL(servidor);   HttpURLConnection con = (HttpURLConnection) url.openConnection();  con.setReadTimeout(60000);  con.setConnectTimeout(60000);  con.setRequestMethod(**"POST"**);  con.setInstanceFollowRedirects(**false**);   con.setDoOutput(**true**);  OutputStreamWriter writer = **new** OutputStreamWriter(  con.getOutputStream());  writer.write(datosEnviados);  writer.flush();   InputStream is = con.getInputStream();  reader = **new** BufferedReader(**new** InputStreamReader(is, **"UTF-8"**));  String data = **null**;  StringBuilder sb = **new** StringBuilder();  **while** ((data = reader.readLine()) != **null**) {  sb.append(data);  }} **catch** (Exception e) {  e.printStackTrace(); *//Imprime seguimiento del error (línea)* } **finally** { *//SE EJECUTA DESPUÉS DEL TRY CATCH* **if** (reader != **null**) {**try** {  reader.close();  } **catch** (IOException e) {  e.printStackTrace();} *//catch* } *// if reader !=* } *//finally* |

**Fragmento del código 13.** Transferencia de la base de datos sugerencia al servidor Web

## **6.4 Implementación del tablero de control**

En esta sección se describen la implementación del tablero de control de manera particular de los KPIs mencionados en la sección 5.4.4. En esta sección se requiere de la base de datos **ubicación**, **movilidad** y **sugerencia**. A continuación, se presentan cada uno de los KPIs que conforman el tablero de control.

En la figura 54 se presenta el número total de usuarios registrados en el sistema clasificado por escuelas y carreras.

En la figura 55 se aborda el número total de usuarios registrados en el sistema clasificados por género masculino y femenino, correspondientes a un total de 70 y 30 alumnos respectivamente.

|  |  |
| --- | --- |
|  |  |
| ***Figura 54.*** *Número total de usuarios registrados en el sistema clasificado por escuelas y carreras* | ***Figura 55.*** *Número total de usuarios registrados en el sistema clasificados por género masculino y femenino* |

En la figura 56 se muestra el número total de usuarios registrados en el sistema clasificados por peatones y conductores, correspondientes a un total de 18 y 82 alumnos respectivamente.

En la figura 57 se expone el rango de edades de los usuarios registrados en el sistema clasificado por:

         17 usuarios menores de 18 años

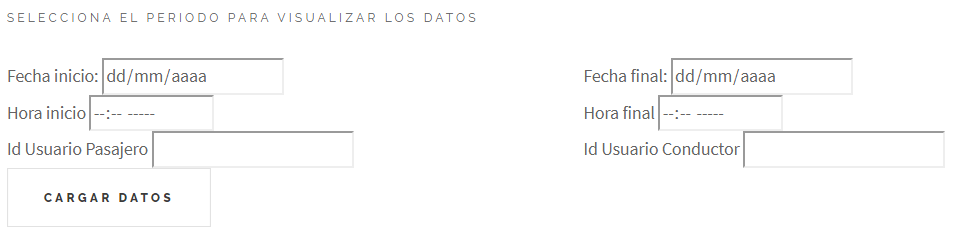
         47 usuarios entre 18 y 25 años

         32 usuarios entre 25 y 40 años

         4 usuario mayor de 40 años

|  |  |
| --- | --- |
|  |  |
| **Figura 56.** Número total de usuarios registrados en el sistema clasificados por peatones y conductores | **Figura 57.** Rango de edades de los usuarios registrado en el sistema |

En la figura 58 se presenta el menú del tablero de control con el cual un usuario selecciona la hora y fecha de inicio y final de las rutas que son de su interés; así como el identificador del usuario (Id del usuario conductor y pasajero) y la identificación del pasajero que le interesa visualizar sus datos.



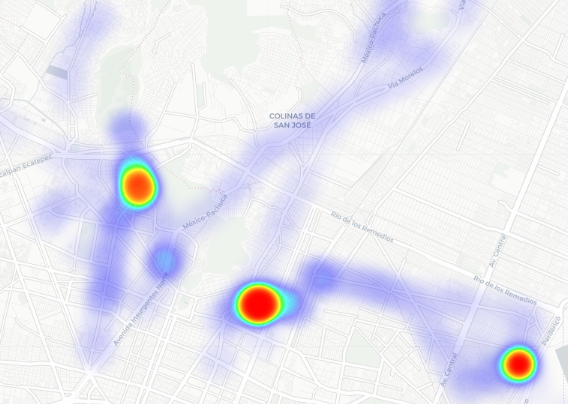
***Figura 58.*** *Selección del periodo de tiempo para la visualización de los datos en el tablero de control*

En la figura 59, se muestra el mapa de trayectorias del tablero de control.



**Figura 59.** Mapa de trayectorias

En la figura 60, se muestra el mapa de calor de las zonas con mayor flujo de movilidad de los usuarios registrados en el sistema, visualizado en el tablero de control.



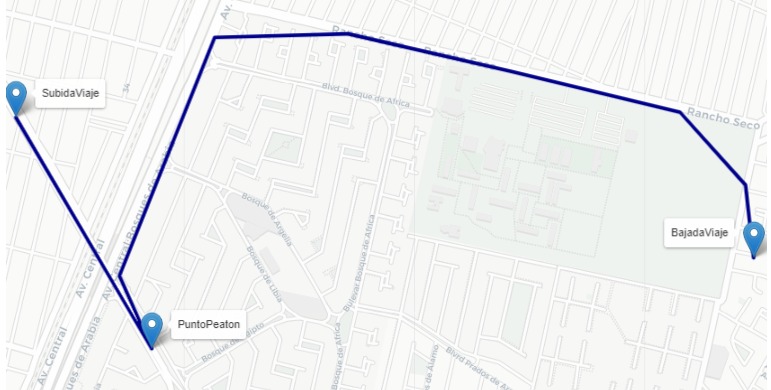
**Figura 60.** Mapa de calor

En la figura 61, se muestra un mapa con el patrón de movilidad individual de un usuario seleccionado en el menú del tablero de control. Cada círculo representa el sitio en que se realizó una muestra de este usuario.



**Figura 61.** Mapa para visualizar el patrón de movilidad individual del usuario

En la figura 62 se muestra un mapa en que se visualiza la ruta que deben compartir los usuarios de tipo peatón y conductor que fueron indicados en el menú del tablero de control.



**Figura 62.** Mapa para visualizar la sugerencia de compartir automóvil

# **7. Pruebas y resultados**

En este capítulo, se describen las pruebas y resultados obtenidos de la recolección de datos de las ubicaciones de 100 usuarios registrados en el sistema durante el transcurso de cuatro semanas.

Este capítulo se estructura de la siguiente manera. En la primera sección se describe el escenario de pruebas. En la segunda sección se describe los datos de prueba utilizados para mostrar la funcionalidad del sistema. En la tercera sección se presentan las pruebas obtenidas en este trabajo. En la cuarta sección se exponen los resultados obtenidos.

## **7.1 Escenario de pruebas**

Para probar el correcto funcionamiento del sistema, se instala la aplicación en dispositivos móviles con sistema operativo Android (6.0 o superior) de la población de UPIITA, en los cuales se les pide a los usuarios que acepten los términos y condiciones para pedir información de su localización y a utilizar cierta cantidad de servicios de comunicaciones, en la cual la ventana de tiempo del proyecto será por un mes, de tal manera que se recabe esta información para poder realizar un análisis que permitirá detectar patrones de movilidad vial y peatonal. Se espera tener como resultado dos módulos, el primero permitirá que los usuarios obtengan sugerencias de compartir el automóvil con respecto a otros usuarios de la misma aplicación que tengan horarios y desplazamientos similares; en el segundo módulo se mostrará en un tablero de control los KPIs establecidos en la sección 5.4.4 “Cálculo de KPIs y despliegue en el tablero de control”.

## **7.2 Datos de Prueba**

En esta subsección se desarrolló la preparación de datos de prueba, los cuales fueron utilizados como datos de entradas para probar el funcionamiento del sistema de sugerencia de viajes compartidos. En la tabla 33, se enlistan los usuarios que tienen patrones de co-movilidad. En la tabla 34 se enlistan los horarios y las rutas cotidianas de los usuarios mencionados en la tabla 33.

**Tabla 33.** Usuarios Propuestos para la preparación de datos de prueba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IdUsuario** | **Nombre** | **Rol** | **Mail** | **Contraseña** |
| 90 | Ulises Castillo | Conductor | [uc@gmail.com](mailto:uc@gmail.com) | uc |
| 91 | Teresa Alvarado | Peatón | ta@gmail.com | ta |
| 92 | Giselle Romero | Conductor | gr@gmail.com | gr |
| 93 | Andrea Peréz | Peatón | ap@gmail.com | ap |
| 94 | Saul Gonzalez | Conductor | sg@gmail.com | sg |
| 95 | Sergio Benitez | Peatón | sb@gmail.com | sb |
| 96 | Angel Vazquez | Conductor | av@gmail.com | av |
| 97 | Carlos Luna | Peatón | cl@gmail.com | cl |
| 98 | Daniel Castillo | Conductor | dc@gmail.com | dc |
| 99 | Ibrahim Ochoa | Peatón | io@gmail.com | io |
| 100 | Yotsan Fiesco | Conductor | yf@gmai.com | yf |

**Tabla 34.** Rutas cotidianas de los usuarios **propuestos** para la preparación de datos de prueba

|  |  |  |
| --- | --- | --- |
| **IdUsuario** | **Ubicación Origen** | **Empresa / Ubicación Destino** |
| 90 | Jardines de Morelos, 55070 Ecatepec de Morelos, México | Amazon. Juan Salvador Agraz 73, Lomas de Santa Fe, Contadero, Cuajimalpa de Morelos, 05348 Ciudad de México, CDMX |
| 91 | Jardines de Morelos, 55070 Ecatepec de Morelos, México | Amazon. Juan Salvador Agraz 73, Lomas de Santa Fe, Contadero, Cuajimalpa de Morelos, 05348 Ciudad de México, CDMX |
| 92 | Amazon. Juan Salvador Agraz 73, Lomas de Santa Fe, Contadero, Cuajimalpa de Morelos, 05348 Ciudad de México, CDMX | UPIITA. Av Instituto Politécnico Nacional 2580, La Laguna Ticoman, Gustavo A. Madero, 07340 Ciudad de México, CDMX |
| 93 | Amazon. Juan Salvador Agraz 73, Lomas de Santa Fe, Contadero, Cuajimalpa de Morelos, 05348 Ciudad de México, CDMX | UPIITA. Av Instituto Politécnico Nacional 2580, La Laguna Ticoman, Gustavo A. Madero, 07340 Ciudad de México, CDMX |
| 94 | Metro Ciudad Azteca. 55120, Cd Azteca 3ra Secc, Ecatepec de Morelos, Méx. | Torre BBVA Bancomer, Juárez 06600 Ciudad de México, CDMX |
| 95 | Metro Ciudad Azteca. 55120, Cd Azteca 3ra Secc, Ecatepec de Morelos, Méx. | Torre BBVA Bancomer, Juárez 06600 Ciudad de México, CDMX |
| 96 | AT&T. Av. Revolución #300, Escandón I Secc, Miguel Hidalgo, 01180 Ciudad de México, CDMX | Casa de los Azulejos. Av Francisco I. Madero 4, Centro Histórico de la Cdad. de México, Centro, Cuauhtémoc, 06500 Ciudad de México, CDMX |
| 97 | AT&T. Av. Revolución #300, Escandón I Secc, Miguel Hidalgo, 01180 Ciudad de México, CDMX | Casa de los Azulejos. Av Francisco I. Madero 4, Centro Histórico de la Cdad. de México, Centro, Cuauhtémoc, 06500 Ciudad de México, CDMX |
| 98 | Metro Lindavista. Lindavista Sur, 07300 Ciudad de México, CDMX | UPIITA. Av Instituto Politécnico Nacional 2580, La Laguna Ticoman, Gustavo A. Madero, 07340 Ciudad de México, CDMX |
| 99 | Metro Lindavista. Lindavista Sur, 07300 Ciudad de México, CDMX | UPIITA. Av Instituto Politécnico Nacional 2580, La Laguna Ticoman, Gustavo A. Madero, 07340 Ciudad de México, CDMX |
| 100 | Metro Ciudad Azteca. 55120, Cd Azteca 3ra Secc, Ecatepec de Morelos, Méx. | Torre BBVA Bancomer, Juárez 06600 Ciudad de México, CDMX |

## **7.3 Pruebas**

En esta sección se presentan las pruebas obtenidas durante un mes al desarrollar este proyecto terminal utilizando un tamaño de población de 100 usuarios.

### **7.3.1 Casos para obtener sugerencia de compartir automóvil**

En esta subsección se presenta un ejemplo del escenario válido para ser considerado como sugerencia para compartir automóvil para usuarios peatones y conductores. Este escenario cumple con la descripción realizada en la subsección 5.1.3.1 y 5.1.3.2.

En la figura 63, se presentan la ruta propuesta del tablero de control para que un usuario de tipo conductor y peatón o conductor y conductor compartan el automóvil. En la figura 64, se muestra la ruta de sugerencia para compartir el automóvil mostrándola en la aplicación móvil. En el desarrollo de este trabajo, se empleó la API de Google para presentarla en un dispositivo móvil, permitiendo el trazado de la ruta origen destino sugerida, donde la ruta predominante será la ruta del conductor para el caso de peatón con conductor y la ruta del conductor será la ruta del usuario que desee seguir siendo conductor para el caso de la sugerencia conductor con conductor.

|  |  |
| --- | --- |
|  |  |
| **Figura 63.** Trayectoria candidata para compartir automóvil en el tablero de control | ***Figura 64.*** *Trayectoria candidata para compartir automóvil en la aplicación móvil* |

### **7.3.2 KPIs del tablero de control**

En esta subsección se presentan los KPIs propuestos en la sección 5.6.

En la figura 65, se presentan los usuarios registrados por escuela. En la tabla 33, se describe la cantidad de estudiantes por escuela y carrera.



**Figura 65.** Escuelas registradas por los usuarios de la aplicación móvil

**Tabla 33.** Escuelas y carreras registradas en el sistema

|  |  |  |
| --- | --- | --- |
| **Escuela** | **Carrera** | **Número de usuarios** |
| UPIITA | Telemática | 64 |
| Mecatrónica | 12 |
| Biónica | 8 |
| ESCOM | Sistemas computacionales | 3 |
| ESFM | Física y matemáticas | 1 |
| UPIBI | Alimentos | 5 |
| ESIQIE | Química industrial | 2 |
| Química petrolera | 1 |
| ENMH | Médico cirujano y partero | 1 |
| ESCA | Contador público | 2 |
| Negocios internacionales | 1 |

## **7.4 Resultados**

En esta sección, se comentan los resultados obtenidos de la sección 7.3.

Dada una muestra de 100 usuarios de la población de UPIITA, donde 82 usuarios son peatones y 18 conductores, se obtuvieron 9 sugerencias de distintos usuarios, donde una sugerencia es de tipo conductor con conductor, logrando así, un 90% de efectividad de que los resultados obtenidos sean aplicables a la totalidad de la comunidad de esta institución.

# **Conclusiones**

En el presente trabajo, se realizó la implementación del sistema de información *Sistema de reconocimiento de patrones de movilidad vial y peatonal para ciudades inteligentes*. Con base en los resultados obtenidos y las pruebas realizadas con las herramientas propuestas en este trabajo se llegó a las siguientes conclusiones:

* El algoritmo para sugerir viajes compartidos, hace análisis de rutas de una complejidad de tipo cuadrático (n)(m) para el caso de conductor con peatón. En el caso de conductor con conductor, la complejidad del algoritmo sería (n)(n+m).

Donde n es el número de usuarios conductores y m es el número de usuarios peatones.

* El GPS tiene problemas para muestrear en lugares cerrados como son: edificios y centros comerciales, así como estaciones subterráneas del metro y lugares cercanos a cerros.
* El margen de error de los GPS varía en modelo, gama y el recurso de la batería que se solicite utilizar: Wifi (ahorro de batería) o GPS (máxima precisión).
* Con el desarrollado del tablero de control se tiene como propósito que las personas tomadoras de decisiones visualizar información clara y precisa de la movilidad de los usuarios que utilizan este sistema. Con la información recabada de las rutas de los usuarios, algunos ejemplos de soluciones que pueden realizar los tomadores de decisiones son:
  + Crear rutas alternas de transporte público o privado que se dirija a zonas concurridas de los usuarios.
  + Colocación de locales de servicios (cajeros, restaurantes, centros comerciales, centro de salud, centros de entretenimiento, entre otros).
* Los usuarios que usan una aplicación móvil, son muestreados en un periodo de tiempo, se pueden utilizar técnicas no supervisadas de aprendizaje automático para encontrar rutas habituales entre los usuarios para sugerirles compartir sus medios de transporte en algún tramo de sus rutas que cumplan con los requisitos para ser evaluados por el algoritmo de viajes compartidos.
* Con el desarrollo del algoritmo para realizar sugerencias de viajes compartidos se logró encontrar sugerencias para compartir sus medios de transportes usuarios de tipo conductor, adicionalmente a las sugerencias de tipo conductor con peatón.Considerando la sugerencia conductora con conductor, si los usuarios aceptan la sugerencia, se podría dar una alternativa a reducir el tránsito vehicular dada la muestra finita obtenida de la población de UPIITA.

Con el desarrollo de este sistema de información fue posible constatar la necesidad de contar con un tamaño de datos requeridos para considerarse una problemática de *Big Data* se deberían tener datos de al menos 50 Terabytes de información de rutas de personas.

# **Trabajos a Futuro**

A partir del 1 de enero del 2020 se aprobó una ley donde todas las plataformas de Internet (páginas Web y aplicaciones móviles) deberán pagar sus impuestos. Por lo tanto, los usuarios prestadores del servicio de compartir su automóvil particular, deberán estar dados de alta en hacienda para reportar sus ingresos, el costo de dicho servicio se verá incrementado en el consumidor final. Por el momento, los bancos y el SAT prorrogaron dicha implementación, debido a que no cuentan con la infraestructura necesaria todavía.

Como continuación de este proyecto, han surgido algunas líneas futuras para llevar a cabo para otros investigadores:

* Se desarrollará un módulo en la aplicación móvil que permita el cobro del servicio en efectivo o tarjeta de débito o crédito.
* Proporcionar escalabilidad al proyecto mediante la implementación de un clúster de *Spark*, almacenando la información en *HDFS* y utilizando un gestor de bases de datos para realizar consultas, como pueden ser: *MongoDB*, *Hive*, *Hbase o Impala*.
* Implementación de la base de datos de *Firebase* para realizar el monitoreo de los usuarios en tiempo real.
* Implementación de plataformas en la nube como son *Amazon Web Services (AWS), Google Cloud Platform (GCP) o Azure* de *Microsoft.*
* Implementación de la aplicación móvil en el sistema operativo *iOS* utilizando los lenguajes de programación *Swift*, *Flutter*, *React* *Native* o *Xamarin*.
* Implementación del tablero de control utilizando *Tableau* o Power *BI* o el uso de marcos de trabajo para desarrollo Web como son *Spring, Angular o ReactJS*.

# **Referencias**

[1] IEEE, “Smart Cities Community, IEEE,” ieee.org, 2018. [En línea]. Disponible en: https://www.ieee.org/membership-catalog/productdetail/showProductDetailPage.html?product=CMYSC764. [Último acceso: 02 Nov 2018].

[2] L. G. Anthopoulos, “The Rise of the Smart City,” Understanding Smart Cities: A Tool for Smart Government or an Industrial Trick?, *Springer International Publishing*, vol.2, pp. 5-41, 2017.

[3] S. Colado et al., “*Smart City. Hacia la gestión Inteligente*,” Primera edición, México: Alfaomega Grupo Editor, 2014, pp. 6,7,15.

[4] IEEE, “IEEE Smart Cities - IEEE Power and Energy Society,” ieee-pes.org, 2018. [En línea]. Disponible en:<https://www.ieee-pes.org/pes-communities/ieee-smart-cities>. [Último acceso: 27 Sep 2018].

[5] Seoul Solution, "[Juniper Research] Global Smart City Performance Index (2017)", 2017. [En línea]. Disponible en: https://seoulsolution.kr/en/content/7664. [Último acceso: 27- Sep- 2018].

[6] EasyPark Group, “EasyPark Group Smart Cities Index,” *Easyparkgroup.com*, 2017. [En línea]. Disponible en: https://easyparkgroup.com/smart-cities-index/. [Último acceso: 27 Sept 2018].

[7] I. Guerra González “Sistema de difusión de información comercial basado en geolocalización y perfil de usuario,” trabajo terminal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Ciudad de México, CDMX, 2012.

[8] B. Ruiz Agustín “Extracción y clasificación de información del tráfico vehicular a través de Twitter y su visualización en Google Maps,” trabajo terminal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Ciudad de México, CDMX, 2012.

[9] R. González Suárez, R. Hernández Tiscareño, V. J. Suárez Peñaloza “#HELP!T7,” trabajo terminal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Ciudad de México, CDMX, 2014.

[10] V. E. Nava Quezada “Aplicación de tráfico y navegación en la Ciudad de México basada en Minería de Textos”, proyecto terminal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Ciudad de México, CDMX, 2015.

[11] D. Sarabia Tiznado, S. J. García Sánchez “Sistema de monitoreo, generación de itinerarios y análisis de datos de rutas de Transporte público,” proyecto terminal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, IPN, Ciudad de México, CDMX, 2017.

[12] A. Moreno. “Smart cities, innovación y eficiencia urbanas: los nuevos modelos de transporte en México. El caso de león, Guanajuato,” *Ideas Concyteg*. vol. 1, pp 979 - 1000.

[13] JUNIPER RESEARCH, “SMART CITIES – WHAT’S IN IT FOR CITIZENS?,” JUNIPER RESEARCH, 2017. [En línea]. Disponible en: https://newsroom.intel.com/wp-content/uploads/sites/11/2018/03/smart-cities-whats-in-it-for-citizens.pdf. [Último acceso: 27 22 2018].

[14] Swarm, “Isochronic Singapore - Swarm,” *Isochrone.swarm.is*, [En línea]. Disponible en: https://isochrone.swarm.is/. [Último acceso: 27 Sept 2018].

[15] SENSEable City Lab, “LIVE Singapore!,” *Senseable.mit.edu*, [En línea]. Disponible en: http://senseable.mit.edu/livesingapore/. [Último acceso: 27 Sept 2018].

[16] Superpedestrian, “Copenhagen Wheel Technology & Features | Superpedestrian,” *Superpedestrian.com*, [En línea]. Disponible en: https://www.superpedestrian.com/en/tech. [Último acceso: 30 Sept 2018].

[17]    Centre for Advanced Spatial Analysis, University College London, #CityDashboard: London,” *Citydashboard.org*, 2012. [En línea]. Disponible en: http://citydashboard.org/london/. [Último acceso: 30 Sept 2018].

[18]    CARTO, “How The New York City Mayor’s Office takes a real-time pulse of the city with its interactive dashboard,” *Carto.com*, 2017. [En línea]. Disponible en: https://carto.com/blog/new-york-city-mayor-office/. [Último acceso: 07 Oct 2018].

[19] Barcelona City Council, “Sentilo,” *Sentilo.io* [En línea]. Disponible en: http://www.sentilo.io/wordpress. [Último acceso: 07 Oct 2018].

[20] Smart Travel News, “Geolocalización móvil para predecir los patrones de movilidad urbana,” *SmartTravelNews*, [En línea]. Disponible en: https://www.smarttravel.news/2017/01/27/geolocalizacion-movil-para-predecir-los-patrones-de-movilidad-urbana/. [Último acceso: 20 Oct 2018].

[21] Comunicaciones 360º, “Geolocalización móvil para predecir los patrones de movilidad móvil,” *MOCA*, 2017. [En línea]. Disponible en: http://hiperconexion.com/geolocalizacion-movil-para-predecir-los-patrones-de-movilidad-movil/. [Último acceso: 25 Oct 2018].

[22] Terrain Technologies, “Siade SaaS | Análisis y optimización del transporte público.,” *Siade.eu*, [En línea]. Disponible en: https://www.siade.eu/es/. [Último acceso: 26 Oct 2018].

[23] X. Chen, D. Shi, B. Zhao and F. Liu, “Mining Individual Mobility Patterns Based on Location History,” *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, Changsha, China, 2016.

[24] Waze, “Is Your Business on the Map?” *Waze Local*, [En línea]. Disponible en: https://www.waze.com/business/#make-your-ad-card. [Último acceso: 2018 Nov 02].

[25] Google, “Google Maps Platform,” *Google Cloud*, [En línea]. Disponible en: https://cloud.google.com/maps-platform/. [Último acceso: 02 Nov 2018].

[26] Uber, “Uber. Movilidad a tu manera,” *Uber.com*, [En línea]. Disponible en: https://www.uber.com/es/mx/. [Último acceso: 02 Nov 2018].

[27] BlaBlaCar, “¿Cómo funciona BlaBlaCar?,” *BlaBlaCar*, [En línea]. Disponible en: https://www.blablacar.mx/how-does-car-sharing-work. [Último acceso: 02 Nov 2018].

[28] T. Mitchel, “Machine Learning.” First Edition. United States of America: McGraw Hill, 1997, pp. 39.

[29] F. Provost y T. Fawcett, “Data Science for Business.” First Edition. United States of America: O’Reilly Media, Inc., 2013, pp. 4,48,78,163

[30] Instituto Nacional de Estadística y Geográfia, “Sistema de Información Geográfica”, INEGI, [En línea]. Disponible en: <https://www.inegi.org.mx/inegi/spc/doc/internet/sistemainformaciongeografica.pdf>. [Último acceso: 05 May 2019].

[31] Víctor Olaya, “Conceptos básicos de visualización y representación”, [En línea]. Disponible en: <http://volaya.github.io/libro-sig/chapters/Conceptos_basicos.html#Conceptos_basicos_visualizacion>. [Último acceso: 05 May 2019].

[32] Portal Educativo, “Coordenadas geográficas”, [En línea]. Disponible en: <https://www.portaleducativo.net/quinto-basico/684/coordenadas-geograficas>. [Último acceso: 05 May 2019].

[33] R.W. Sinnott, "Virtues of the Haversine", Sky and Telescope, vol. 68, no. 2, 1984, p. 159.

[34]    E. Estellés *et al*., “Crowdsourcing Fundamentals: Definition and Typology,” *Springer International Publishing*, vol. 1, pp. 33-48, 2015.

[35] M. O’Mahony, “Crowdsensing en smart cities”, Tesis, Madrid, 2018.

[36] E. Nadelhoffer, “*10 Best Practices for Building Effective Dashboards.*” Tableau Software. 2017, p. 2.

[37] I. Sommerville, “*Ingeniería de Software.*” 9ª edición. Naucalpan de Juárez, Estado de México.: Addison-Wesley, 2011, pp. 126,147.

[38] Smart Vision Europe Ltd, “What is the CRISP-DM methodology?,” *Smart Vision* - Europe, [En línea]. Disponible en: https://www.sv-europe.com/crisp-dm -methodology/. [Último acceso: 02 Nov 2018].

[39] C. Fontela, “Modelado de software para profesionales.” Primera Edición. Buenos Artes: Alfaomega Grupo Editor Argentino., 2011, pp. 28,29,30.

[40] “2013 01 28 POO utilizando JAVA”, apuntes de clases POO propiedad del Dr. Jesús Manuel Olivares Ceja, Laboratorio de Ciencia de Datos y Tecnologías de Software, Centro de Investigación en Computación, IPN, verano 2019.

[41] F. J. Ceballos, “*Java 2.*” 4ta Edición. Madrid, España: RA-MA Editorial, 2011, p. 401.

[42] Lucid Software, “Notación y símbolos para la creación de mapas de procesos”, 2019. [En línea]. Disponible en: https://www.lucidchart.com/pages/es/notacion-y-simbolos-para-la-creacion-de-mapas-de-procesos. [Último acceso: 23 Mar 2019].

[43] Android, “android,” Astrium, DigitalGlobe, [En línea]. Disponible en: https://www.android.com/. [Último acceso: 20 Abr 2019].

[44] Mobile App Marketing by PickASO, “Desarrollo de apps: diferencia entre Android y iOS”, 2019. [En línea]. Disponible en: <https://pickaso.com/2019/desarrollo-apps-diferencias-android-vs-ios>. [Último acceso: 20 Abr 2019].

[45] Xatakamovil, “Así es como A”, 2019. [En línea]. Disponible en: <https://www.xatakamovil.com/sistemas-operativos/asi-como-android-se-ha-comido-mercado-diez-anos>. [Último acceso: 23 Mar 2019].

[46] Teknowledge Software, “Eclipse vs Android Studio: Which IDE Is Better For Android Developers,” 2015. [En línea]. Disponible en: <https://teks.co.in/site/blog/eclipse-vs-android-studio-ide-better-android-developers/>. [Último acceso: 23 Mar 2019].

[47] Android Studio FAQs, “Android Studio: Requisitos mínimos,” 2016. [En línea]. Disponible en: <https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos>. [Último acceso: 23 Mar 2019

[48] Android, “Preguntas frecuentes sobre Kotlin en Android,” 2019. [En línea]. Disponible en: <https://developer.android.com/kotlin/faq?hl=es-419> [Último acceso: 23 Mar 2019].

[49] L. Darcey, S. Conder, “*Android Wireless Application Development.*” Third Edition. United States: Addison-Wesley, 2012, p.35.

[50] Google, “Maps,” Google Cloud, [En línea]. Disponible en: https://cloud.google.com/maps-platform/maps/. [Último acceso: 02 Nov 2018].

[51] 000webhost.com, “HOSTINGER 000webhost,” 000webhost.com [En línea]. Disponible en: <https://www.000webhost.com/features>. [Último acceso: 20 Abr 2019].

[52] Freehostia.com “Free Cloud Hosting,” 2019. [En línea]. Disponible en: <https://www.freehostia.com/free-cloud-hosting/>. [Último acceso: 20 Abr 2019].

[53] KnowledgeHut (2019), “Scala Vs Python Vs R Vs Java - Which language is better for Spark & Why?”. [En línea]. Disponible en: <https://www.knowledgehut.com/blog/programming/scala-vs-python-vs-r-vs-java>. [Último acceso: 20 Abr 2019].

[54] T. White, “Hadoop: The Definitive Guide,” First Edition. Gravenstein Highway North, Sebastopol: O’REILLY, 2009, p. 41.

[55] Solid IT (2019), “System Properties of Cassandra”. [En línea]. Disponible en: <https://db-engines.com/en/system/Cassandra%3BHBase%3BMySQL>. [Último acceso: 20 Abr 2019].

[56] Oracle (2019), “JavaServer Pages Technology”. [En línea]. Disponible en: <https://www.oracle.com/technetwork/java/index-jsp-138231.html>. [Último acceso: 20 Abr 2019].

[57] Mozilla and individual contributors (2019). “CSS”. [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>. [Último acceso: 20 Abr 2019].

[58] W3.CSS (2019). “JavaScript Tutorial”. [En línea]. Disponible en: <https://www.w3schools.com/js/>. [Último acceso: 20 Abr 2019].

[59] Mike Bostock (2019). “Data-Driven Documents”. [En línea]. Disponible en: <https://d3js.org/>. [Último acceso: 20 Abr 2019].

[60] OpenStreetMap (2019). “Leaflet”. [En línea]. Disponible en: <https://leafletjs.com/>. [Último acceso: 20 Abr 2019].

[61] Oracle (2019). “Oracle MySQL Cloud Service”. [En línea]. Disponible en: <https://www.oracle.com/mx/mysql/>. [Último acceso: 20 Abr 2019].

[62] A. Gómez, “*Enciclopedia de la Seguridad Informática.*” 2ª Edición. Madrid, España: RA-MA Editorial, 2014, pp. 32.

[63] Federal Information Processing Standards Publication (2001). “Advanced Encryption Standard (AES)”. [En línea]. Disponible en: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf>. [Último acceso: 20 Abr 2019].

[64] AES, “AES como Estándar Internacional de Cifrado”. [En línea]. Disponible en: <https://www.conaic.net/revista/publicaciones/Vol_V_Num1_Ene_Abr_2018/Articulo10.pdf>. [Último acceso: 20 Abr 2019].

[65] Oracle Corporation and/or its affiliates (2019). “11.8 Data Type Storage Requirements”. [En línea]. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html#data-types-storage-reqs-numeric>. [Último acceso: 20 Abr 2019].

[66] Interactive Programmers Community (2019). “Base de datos – calcular volumen”. [En línea]. Disponible en: <https://www.lawebdelprogramador.com/foros/Bases-de-Datos/557346-calcular-volumen.html>. [Último acceso: 20 Abr 2019].

[67] North Coast Media LLC (2019). “Innovation: Precise positioning using raw GPS measurements from Android”. [En línea]. Disponible en: <https://www.gpsworld.com/innovation-precise-positioning-using-raw-gps-measurements-from-android-smartphones/>. [Último acceso: 20 Abr 2019].

[68] Uber Technologies Inc. (2019). “Improving Uber’s Mapping Accuracy with CatchME”. [En línea]. Disponible en: <https://eng.uber.com/mapping-accuracy-with-catchme/>. [Último acceso: 20 Abr 2019].

[69] Investigación y Reconstrucción de Accidentes de Tráfico (2019). “Tabla relación velocidad de peatones caminando”. [En línea]. Disponible en: <https://causadirecta.com/especial/calculo-de-velocidades/tablas/tabla-relacion-velocidad-de-peatones-caminando>. [Último acceso: 20 Abr 2019].

[70] Creative Commons Atribution 4.0 International License (2019). “Datos y estadísticas de uso de transporte público en Ciudad de México, México”. [En línea]. Disponible en: <https://moovitapp.com/insights/es-419/Moovit_Insights_%C3%8Dndice_de_Transporte_P%C3%BAblico-822>. [Último acceso: 20 Abr 2019].

[71] Normas APA “Fórmula para calcular la muestra de una población”, 2019 [En línea]. Disponible en: <http://normasapa.net/formula-muestra-poblacion/>. [Último acceso: 23 Mar 2019].

# **Anexos**

## **1 Análisis para obtener la muestra de una población Finita**

En este anexo del trabajo terminal propuesto se presentan el análisis realizado para definir el número de personas que es recomendable entrevistar para conocer el grado de aceptación de un proyecto dada una población [71]. Observarse en 20.

(20)

|  |
| --- |
|  |

**Tabla 34.** Descripción de cada variable de la ecuación para calcular la muestra de una población Finita.

|  |  |  |
| --- | --- | --- |
| **Símbolo** | | **Descripción** |
|  | Tamaño de muestra buscado | |
|  | Tamaño de la población o universo | |
|  | Parámetro estadístico que depende del Nivel de confianza (Nc). Utiliza la distribución de | |
|  | Probabilidad de que ocurra el evento estudiado (éxito). Probabilidad esperada | |
|  | = Probabilidad de que no ocurra el evento estudiado. Fracaso | |
|  | Error de estimación máximo aceptado | |

La realización de este análisis, fue necesario investigar el tamaño de la población de UPIITA del periodo 2018, siendo este el último censo realizado por la escuela, el resultado obtenido fue de 2420 alumnos estudiando en este plantel.

El porcentaje de probabilidad para que ocurra este evento es del 90%, y de que fracase es 10%.

En esta ecuación se utiliza un nivel de confianza del 95%, utilizando la tabla 35 correspondiente a la tabla de , se selecciona el valor de respectivo al nivel de confianza seleccionado, siendo este 1.96. Por lo tanto, el error de estimación máximo aceptado es del 5%.

Se obtiene el resultado de 131 estudiantes que deben ser entrevistados para conocer si el proyecto terminal propuesto es viable para ser aplicado en la UPIITA.

**Tabla 35.** Distribución de

|  |  |
| --- | --- |
| **Nivel de confianza (%)** |  |
| 99.7 | 3 |
| 99.0 | 2.58 |
| 98.0 | 2.33 |
| 96.0 | 2.05 |
| **95.0** | **1.96** |
| 90.0 | 1.645 |
| 80.0 | 1.28 |
| 75.0 | 1.15 |
| 70.0 | 1.03 |

Sustituyendo los valores de la expresión 20, se obtiene el siguiente resultado de la expresión 21.

|  |
| --- |
|  |

(21)

## **2 Encuesta realizada a la población de UPIITA**

En este anexo del trabajo terminal se presenta la encuesta realizada a 131 estudiantes de la población de UPIITA durante el periodo 2019-1, para conocer el porcentaje de aceptación que se tendrían de los estudiantes para participar en este proyecto.

A continuación, se exponen cada una de las preguntas que se realizaron en la encuesta.

En la tabla 38 se muestra el porcentaje estudiantes de género hombre y mujer entrevistados, donde aproximadamente el 20% eran mujeres.

**Tabla 38.** Porcentaje de la población de UPIITA entrevistada de género hombre y mujer.

|  |  |  |  |
| --- | --- | --- | --- |
| **1. Género** | **Género** | **Cantidad** | **(%)** |
| **https://lh4.googleusercontent.com/QHT8vT5ZBs5xRxCbGEEujb7pEfKbyYJiYWrYGHg_AstjMqjuZJPlc1M2FhLfgjJJcntdsgYA0zbAatZpoPiGLix9RF0Dn1WDwqmTBIK1RKyGq-j-4RKbGnEzh0X9E4kxywNFechT** | **https://lh4.googleusercontent.com/QHT8vT5ZBs5xRxCbGEEujb7pEfKbyYJiYWrYGHg_AstjMqjuZJPlc1M2FhLfgjJJcntdsgYA0zbAatZpoPiGLix9RF0Dn1WDwqmTBIK1RKyGq-j-4RKbGnEzh0X9E4kxywNFechT** | 105 | 80.2 |
| **https://lh4.googleusercontent.com/QHT8vT5ZBs5xRxCbGEEujb7pEfKbyYJiYWrYGHg_AstjMqjuZJPlc1M2FhLfgjJJcntdsgYA0zbAatZpoPiGLix9RF0Dn1WDwqmTBIK1RKyGq-j-4RKbGnEzh0X9E4kxywNFechT** | 26 | 19.8 |
|  |  |  |

En la tabla 39 se muestra el porcentaje estudiantes de cada una de las carreras disponibles en la UPIITA.

**Tabla 39.** Porcentaje de la población entrevistada de cada una de las carreras disponibles en la UPIITA.

|  |  |  |  |
| --- | --- | --- | --- |
| **2. ¿Cuál es tu carrera?** | | **Cantidad** | **(%)** |
| **https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF** | **https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF** | 54 | 41.2 |
| **https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF** | 50 | 38.2 |
| https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF | 22 | 16.8 |
| **https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF** | 4 | 3.1 |
| **https://lh4.googleusercontent.com/2gLC5_a1rUizN77yZuNwmjkJ9z6QbV0YKjgK0KzQXpFQBso2O_LqBGLBpaumbJBwvkodvLIrA9eRvvKNpyNyuADTyJIy6Sc8_nrqb-Ibz6lwO-0q7pPnEx9zKwuSGVQNfEYzCzKF** | 1 | 0.8 |

En la tabla 40 se muestra el porcentaje del tiempo aproximado que tardan los estudiantes de la UPIITA en llegar a la escuela usando transporte público.

**Tabla 40.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a la escuela usando transporte público.

|  |  |  |  |
| --- | --- | --- | --- |
| **3. ¿Cuál es el tiempo aproximado que le toma llegar a la escuela en transporte público?** | **Tiempo** | **Cantidad** | **(%)** |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 6 | 4.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 9 | 6.9 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 5 | 3.8 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 39 | 29.8 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 3 | 2.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 25 | 19.1 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 24 | 18.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 15 | 11.5 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 5 | 3.8 |

En la tabla 41 se muestra el porcentaje del tiempo aproximado que tardan los estudiantes de la UPIITA en llegar a su hogar usando transporte público.

**Tabla 41.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a su hogar usando transporte público.

|  |  |  |  |
| --- | --- | --- | --- |
| **4. ¿Cuál es el tiempo aproximado que le toma regresar a su casa en transporte público?** | **Tiempo** | **Cantidad** | **(%)** |
| https://lh3.googleusercontent.com/wsm421vZUrPVS6--AI-SkNCN_cYU-bExCLKZF1PFKs7MmCPKtx50xVUkIWtwZXlKzxwNfYcvEK4jWwWgMqz2nZ0CuMOSAGEIPexFbkMg1HtleWRPDjLGv24C2oPRFdvNQ_5z5UGK | https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 6 | 4.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 7 | 5.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 6 | 4.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 26 | 19.8 |
| https://lh3.googleusercontent.com/wsm421vZUrPVS6--AI-SkNCN_cYU-bExCLKZF1PFKs7MmCPKtx50xVUkIWtwZXlKzxwNfYcvEK4jWwWgMqz2nZ0CuMOSAGEIPexFbkMg1HtleWRPDjLGv24C2oPRFdvNQ_5z5UGK | 5 | 3.8 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 30 | 22.9 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 24 | 18.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 17 | 13 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 10 | 7.6 |

En la tabla 42 se muestra el porcentaje del tiempo aproximado que tardan los estudiantes de la UPIITA en llegar a la escuela usando automóvil.

**Tabla 42.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar escuela usando automóvil.

|  |  |  |  |
| --- | --- | --- | --- |
| **5. ¿Cuál es el tiempo aproximado que le toma llegar a la escuela en automóvil?** | **Tiempo** | **Cantidad** | **(%)** |
| https://lh3.googleusercontent.com/SdYkbj1zspTL1x52WtJY-hMOlglTqamELFaF2kF84z9UsU1LycLAdwA-4Wu_uKkqR0w_5vNuhjmfgvZVSxjM5p8f1A7JqulpM_mIPgVJRvuTJPCafClAZ7vBh2QnOxLfkIKroh9W | https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 11 | 8.4 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 33 | 25.2 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 35 | 26.7 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 29 | 22.1 |
| https://lh3.googleusercontent.com/wsm421vZUrPVS6--AI-SkNCN_cYU-bExCLKZF1PFKs7MmCPKtx50xVUkIWtwZXlKzxwNfYcvEK4jWwWgMqz2nZ0CuMOSAGEIPexFbkMg1HtleWRPDjLGv24C2oPRFdvNQ_5z5UGK | 3 | 2.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 10 | 7.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 7 | 5.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 2 | 1.5 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 1 | 0.8 |

En la tabla 43 se muestra el porcentaje del tiempo aproximado que tardan los estudiantes de la UPIITA en llegar a su casa usando automóvil.

**Tabla 43.** Porcentaje del tiempo que tardan los estudiantes de la UPIITA en llegar a su casa usando automóvil.

|  |  |  |  |
| --- | --- | --- | --- |
| **6. ¿Cuál es el tiempo aproximado que le toma regresar a su casa en automóvil?** | **Tiempo** | **Cantidad** | **(%)** |
| https://lh4.googleusercontent.com/N8JLGmqsqNtsRMIRqAlNkcu7wmpIR00AdsliuIPaY2gfVit_co_1wqPwfwUGKdD1KGijpKZvA6on0lfoNg9Q55vXytr58EAjC9GoRtdwp2pCZn8jL8ZXhCKu3aYm1ssXD0gIuSeV | https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 11 | 8.4 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 27 | 20.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 34 | 26 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 31 | 23.7 |
| https://lh3.googleusercontent.com/wsm421vZUrPVS6--AI-SkNCN_cYU-bExCLKZF1PFKs7MmCPKtx50xVUkIWtwZXlKzxwNfYcvEK4jWwWgMqz2nZ0CuMOSAGEIPexFbkMg1HtleWRPDjLGv24C2oPRFdvNQ_5z5UGK | 3 | 2.3 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 13 | 9.9 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 6 | 4.6 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 4 | 3.1 |
| https://lh3.googleusercontent.com/-jHChuWvQOPYpypZ202FJIDsDaij9ChMYfgwuHng2Y6iQfWtGJVBvBGrZsRSCVhL14ryjhoDCm3e-qJ3t1ByiFYh75V803Xyo7hwtPkAn6iRnA8elq2DkApmwwwzSOYJ88KmWHbS | 2 | 1.5 |

En la tabla 44 se muestra el porcentaje de estudiantes de UPIITA que cuentan con automóvil propio para llegar a la escuela.

**Tabla 44**. Porcentaje de estudiantes de UPIITA que cuentan con automóvil.

|  |  |  |  |
| --- | --- | --- | --- |
| **7. ¿Cuenta con carro?** | **Respuesta** | **Cantidad** | **(%)** |
| **https://lh4.googleusercontent.com/QHT8vT5ZBs5xRxCbGEEujb7pEfKbyYJiYWrYGHg_AstjMqjuZJPlc1M2FhLfgjJJcntdsgYA0zbAatZpoPiGLix9RF0Dn1WDwqmTBIK1RKyGq-j-4RKbGnEzh0X9E4kxywNFechT** | https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 108 | 17.6 |
| https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 23 | 82.4 |
|  |  |  |

En la tabla 45 se muestra el porcentaje de estudiantes de UPIITA que compartirían su automóvil con otros estudiantes del plantel.

**Tabla 45.** Porcentaje de estudiantes de UPIITA que compartirían su automóvil.

|  |  |  |  |
| --- | --- | --- | --- |
| **7. ¿Compartirías tu automóvil para ir a la escuela o regresar a tu casa con otro estudiante de la UPIITA que tuviera rutas y horarios semejantes?** | **Respuesta** | **Cantidad** | **(%)** |
| https://lh6.googleusercontent.com/ZtzFxzGnFe9TuGWaQo9FBGemC9LU5_ucHBW3OMuOUqxarM-IlPJMtZawTdNytz9UOKRVJVYnvAXFL2triZt-5umpl3zYibT_liYkJ3X9-SrqbvFcDNwBtL9AGNYkfUC2quE6g1IJ | https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 22 | 95.7 |
| https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 1 | 4.3 |

En la tabla 46 se muestra el porcentaje de estudiantes de UPIITA que no disponen de automóvil privado pero que aceptaría compartir automóvil con algún estudiante que si tuviera.

**Tabla 46.** Porcentaje de estudiantes de UPIITA que no disponen de automóvil privado.

|  |  |  |  |
| --- | --- | --- | --- |
| **¿Aceptarías que algún estudiante de la UPIITA compartiera su automóvil contigo si tuviera rutas y horarios semejantes?** | **Respuesta** | **Cantidad** | **(%)** |
| https://lh5.googleusercontent.com/yi3X4jn7SOZAc53prXbxzp8z_eKZNBdSvtKCyTXR_u7SkzXR2N3AnWlULBLasSC50s1fSEwA_s7BJ79zOmvNyF3t_Hos5xRYNPi_eCldZBDtONz4r61vvtluHJ0tqIrhvdfXJ_qI | https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 101 | 93.5 |
| https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 7 | 6.5 |

En la tabla 47 se muestra el porcentaje de estudiantes de UPIITA aceptarían participar en este proyecto compartiendo sus datos en forma anónima.

**Tabla 47.** Porcentaje de estudiantes de UPIITA compartirían sus datos en forma anónima.

|  |  |  |  |
| --- | --- | --- | --- |
| **8. ¿Aceptarías participar en este proyecto, compartiendo tus datos en forma anónima? Este trabajo utilizará un esquema de seguridad para la protección de tu información** | **Respuesta** | **Cantidad** | **(%)** |
| https://lh6.googleusercontent.com/zHNuZHvyjAHWToNL8DGBohZ9sPxghDNnuz_627kvtm9qhsVqWUl-xb4KN923LotoGcJLvRaAogDGqk4bLp5Ej9QhCgUnIBLyGgznClLtIXea5qX62vQUaMKdB3ADAa2JcxQIg9Eq | https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 112 | 85.5 |
| https://lh5.googleusercontent.com/VCirXf2e6pVOH8X-YNH__SR_CpFCMmdwPZUKiw-hw4KW_rB6u3ZoiCUpGRhWrpc_Q25abcNXD_Fe59F3BOd-zI6xAe227znbITTyzdscoKVVNskhIZtuYSYLYLG0h2liB1gv3qPV | 19 | 14.5 |

## **Referencias Asesores**

**Dr. Jesús Manuel Olivares Ceja**

Titulado como licenciado en Ciencias de la Información de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales Administrativas (UPIICSA) del Instituto Politécnico Nacional (IPN) en 1992, el grado de Maestro en Ciencias de la Ingeniería Eléctrica con especialidad en Computación del Centro de Investigación y Estudios Avanzados (CINVESTAV) en 1996 y el de Doctor en Ciencias de la Computación del Centro de Investigación en Computación (CIC-IPN) en 2002. Ha desarrollado varias aplicaciones de software tanto en forma independiente como para empresas bancarias y comerciales. Es profesor-investigador en el Centro de Investigación en Computación del IPN adscrito al Laboratorio de Ciencia de Datos y Tecnología de Software.



**Dra. Benina Velázquez Ordoñez**

Titulada en licenciatura en Informática en el Instituto Tecnológico del ISTMO (I.T.R) en 2001, el grado de maestría en Ciencias con Especialidad en Ingeniería de Sistemas en 2011 y el grado de Doctor en Ingeniería de sistemas en la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco en 2017. Ha desarrollado varias investigaciones en el Centro de Investigación en Computación (CIC-IPN). Actualmente es encargada del Catálogo de Bienes Informáticos del IPN en la Dirección de Cómputo y Comunicaciones.

