



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN MECATRÓNICA

TEMA:

“GESTIÓN DE TIEMPO EN RUTAS DE TRANSPORTE SOBRE
DATOS DE POSICIONAMIENTO GLOBAL”

AUTOR: LUIS ARMANDO FARINANGO IPIALES

DIRECTOR: CARLOS XAVIER ROSERO CHANDI

IBARRA-ECUADOR

2022



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

| DATOS DEL AUTOR | | | |
|------------------------------------|--|------------------------|------------|
| CÉDULA DE IDENTIDAD: | 1003543061 | | |
| APELLIDOS Y NOMBRES: | FARINANGO IPIALES LUIS ARMANDO | | |
| DIRECCIÓN: | Av. 17 de Julio 5-21, Ibarra | | |
| EMAIL: | lafarinangoi@utn.edu.ec | | |
| TELÉFONO FIJO: | | TELÉFONO MÓVIL: | 0978785309 |
| DATOS DE LA OBRA | | | |
| TÍTULO: | “GESTIÓN DE TIEMPO EN RUTAS DE TRANSPORTE SOBRE DATOS DE POSICIONAMIENTO GLOBAL” | | |
| AUTOR: | LUIS ARMANDO FARINANGO IPIALES | | |
| FECHA (AAAA-MM-DD): | 2022-01-26 | | |
| SÓLO PARA TRABAJOS DE GRADO | | | |
| PROGRAMA: | PREGRADO | | |
| TÍTULO POR EL QUE OPTA: | INGENIERO EN MECATRÓNICA | | |
| ASESOR/DIRECTOR: | CARLOS XAVIER ROSERO C. | | |

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización se desarrolló sin violar derechos de autor de terceros, por lo tanto es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 26 días del mes de enero de 2022.



Luis Armando Farinango Ipiales
C.I.: 1003543061



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CERTIFICACIÓN

En calidad de director del trabajo de grado “GESTIÓN DE TIEMPO EN RUTAS DE TRANSPORTE SOBRE DATOS DE POSICIONAMIENTO GLOBAL”, presentado por el egresado LUIS ARMANDO FARINANGO IPIALES, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, enero de 2022.

Carlos Xavier Rosero
DIRECTOR DE TESIS

Agradecimiento

Quiero dejar expreso un profundo agradecimiento a la Universidad Técnica del Norte porque a través de sus directivos me dieron la oportunidad de ingresar a un mundo de conocimientos.

Al Dr. Carlos Xavier Rosero por ayudarme en mi formación de ingeniería hasta el final de mi recorrido académico, quien con su experiencia y conocimiento supo orientar y desarrollar siempre mi potencial estudiantil.

Luis Armando

Dedicatoria

La perseverancia y la paciencia siempre darán los mejores resultados. Quiero dedicar este trabajo de grado a Dios por brindarme la fuerza necesaria para culminar mi carrera y parte de mis metas, a mis padres Bibiana Ipiales e Isidro Farinango por ser mis guías con sus consejos y experiencias a lo largo de este recorrido, y por último a mis hermanas que siempre me demostraron apoyo incondicional.

Luis Armando

Resumen

El tráfico en las ciudades a menudo se ve afectado por condiciones estocásticas tales como existencia de semáforos, características de la carretera, cantidad de vehículos, variables climáticas, estilo de conducción, entre otras. En este contexto, la provisión de información precisa y a tiempo sobre posiciones, velocidades, tiempos de viaje y horas estimadas de llegada de una flota de vehículos, es valiosa para operadores de transporte y para usuarios. Los operadores podrían optimizar los despachos de los vehículos, y los usuarios podrían crear preferencias de viaje de forma dinámica.

En este trabajo se desarrolla un método para la gestión de tiempo en vehículos, específicamente autobuses, cuando el despacho se basa en horarios programados. Como entradas se consideran las trayectorias de los buses, las separaciones temporales entre puntos de verificación y los datos instantáneos tomados desde dispositivos de localización colocados en los vehículos. Las salidas obtenidas son los estados de navegación y las horas estimadas de llegada de los autobuses a las paradas y/o puntos de verificación.

El sistema ha sido implementado sobre una aplicación web y comprende principalmente un mapa sobre el que se muestra la navegación de una flota de autobuses. Además, comprende controles para selección de rutas y estaciones, y un centro de mensajes que indica la gestión de tiempo. El escenario considerado consiste en varias rutas de autobuses de servicio urbano despachadas en horarios programados, con puntos de verificación ajustados convenientemente a lo largo de las rutas. La aplicación web podría ser accedida a través de internet o se proyectaría en cada parada para que los usuarios puedan observar la información sobre la navegación de los vehículos, relativa a esa estación.

Los resultados del modelo confirman que la solución es efectiva para gestionar el tiempo en cada parada y que permite que los usuarios conozcan con precisión esta información.

Abstract

Traffic in cities is often affected by stochastic conditions such as traffic lights, road characteristics, number of vehicles, climatic variables, driving style, among others. In this context, the provision of accurate and timely information on positions, speeds, travel times and estimated arrival times of the vehicle fleet is valuable for transport operators and final users. Operators could optimize vehicle dispatches, and users could dynamically create travel preferences.

In this work, a method is developed for time management in vehicles, specifically buses, when dispatch is based on scheduled times. As inputs, the routes of the buses, the temporal separations between checkpoints and the instantaneous data taken from tracking devices placed on the vehicles are considered. The outputs obtained are the navigation states and the estimated times of arrival of the buses at the stops and/or checkpoints.

The system has been implemented on a web application and mainly comprises a map on which the navigation of a fleet of buses is shown. In addition, it includes control buttons for selecting routes and stations, and a text area acting as message center that indicates the time management. The scenario considered consists of several urban service bus routes dispatched at scheduled times, with conveniently adjusted checkpoints along the routes. The web application could be accessed through the internet or it would be projected at each stop so that users can observe the information on vehicle navigation relative to that station.

The results of the model confirm that the solution is effective in managing time at each stop and that it allows users to know precisely this information.

Índice general

| | |
|---|-------------|
| Índice general | X |
| Índice de figuras | XIII |
| Índice de cuadros | XIV |
| 1. Introducción | 1 |
| 1.1. Alcance | 1 |
| 1.2. Objetivos | 2 |
| 1.3. Estructura del documento | 2 |
| 2. Revisión Literaria | 3 |
| 2.1. Características técnicas de los sistemas de información de llegada de vehículos en tiempo real | 3 |
| 2.1.1. Tecnología subyacente de ubicación automática de vehículos | 3 |
| 2.1.2. Distribución de la información en tiempo real | 4 |
| 2.1.3. Uso de la información generada por el sistema | 5 |
| 2.2. Factores que afectan a la predicción de la hora de llegada | 6 |
| 2.3. Modelos de pronóstico de la hora de llegada | 7 |
| 2.3.1. Modelos de aprendizaje automático | 8 |
| 2.3.2. Modelos basados en datos históricos | 9 |
| 2.3.3. Modelos estadísticos | 10 |

| | |
|--|-----------|
| 3. Metodología | 12 |
| 3.1. Breve descripción de la propuesta | 12 |
| 3.2. Adquisición de datos en tiempo real | 13 |
| 3.2.1. Dispositivos de rastreo | 14 |
| 3.2.2. Base de datos de tiempo real | 15 |
| 3.3. Cuadro de trabajo de los autobuses | 16 |
| 3.4. Itinerario y planificación de cada ruta | 17 |
| 3.5. Programa para gestión de tiempo | 18 |
| 3.5.1. Front end | 18 |
| 3.5.2. Back end | 19 |
| 3.5.3. Visualización en el mapa | 20 |
| 3.5.4. Gestión de tiempo | 21 |
| 3.5.4.1. Cálculo de los tiempos de llegada | 22 |
| 3.5.4.2. Estados de los autobuses | 23 |
| 4. Resultados | 26 |
| 4.1. Implementación | 26 |
| 4.1.1. Aplicación Web | 26 |
| 4.1.2. Base de datos de tiempo real | 27 |
| 4.1.3. Dispositivo de rastreo | 28 |
| 4.2. Pruebas y resultados | 28 |
| 4.2.1. Escenario de pruebas | 28 |
| 4.2.2. Pruebas | 31 |
| 4.2.2.1. Cálculo del tiempo de llegada | 31 |
| 4.2.3. Análisis de resultados | 32 |
| 5. Conclusiones y trabajo futuro | 37 |
| 5.1. Conclusiones | 37 |
| 5.2. Trabajo futuro | 38 |

Índice de figuras

| | |
|--|----|
| 2.1. Factores que afectan a la predicción de la hora de llegada | 7 |
| 2.2. Modelos de pronóstico y subcategorías | 8 |
| 3.1. Método de rastreo satelital | 13 |
| 3.2. Diagrama de bloques de la propuesta | 14 |
| 3.3. Máquina de estados en base al tiempo | 23 |
| 3.4. Máquina de estados en base a la distancia | 24 |
| 3.5. Relación entre umbrales de distancia | 25 |
| 4.1. Front end de la aplicación web | 27 |
| 4.2. Consola de Firebase mientras sincroniza datos | 28 |
| 4.3. Consola que muestra al script de Python enviando tramas JSON a la base de datos | 29 |
| 4.4. Aplicación web en marcha | 31 |
| 4.5. Predicciones de llegada para la línea 1 | 32 |
| 4.6. Predicciones de llegada para la línea 2 | 33 |
| 4.7. Predicciones de llegada para la línea 3 | 34 |
| 4.8. Errores de predicción de llegada | 35 |

Índice de cuadros

| | |
|---|----|
| 3.1. Campos de la trama de datos JSON enviada por los dispositivos de rastreo . . . | 15 |
| 4.1. Rutas de los autobuses y sus características | 29 |

Lista de algoritmos

| | | |
|----|--|----|
| 1. | Datos de navegación enviados a la base de datos | 15 |
| 2. | Subrutinas de la aplicación y su dinámica de ejecución | 19 |
| 3. | Inicialización del mapa | 20 |
| 4. | Actualización de un bus sobre el mapa | 21 |
| 5. | Actualización de la información | 21 |
| 6. | Mensajes de gestión de tiempo | 21 |

Capítulo 1

Introducción

En esta sección se presenta la propuesta original del trabajo de grado. Se detallan el alcance deseado y los objetivos del proyecto. Se finaliza con la descripción de la estructura del documento.

1.1. Alcance

El propósito de este trabajo es desarrollar un método de gestión de tiempo en rutas de transporte sobre datos obtenidos a través de dispositivos del sistema de posicionamiento global (GPS, Global Positioning System) en vehículos, de tal forma que se pueda concebir una aplicación en software libre. Como entradas se tendrán la trayectoria que debe cumplir el vehículo, las ubicaciones de los puntos de verificación de cumplimiento de la trayectoria, las separaciones temporales entre estos puntos de verificación y datos instantáneos tomados desde los dispositivos GPS. Las salidas de este sistema serán los tiempos de desfase que existan entre las posiciones instantáneas y los puntos de verificación y por lo tanto, las horas estimadas de llegada.

La solución se realizará a través de software matemático libre, además se implementará de tal manera que se pueda visualizar su efecto sobre mapas virtuales. Por último, esta aplicación podrá ser implementada en plataformas de software libre que trabajen en la gestión de datos de flotas de transporte.

El trabajo comprende tres partes:

1. Análisis literario de técnicas para la predicción de llegada de vehículos.
2. Desarrollo teórico, que consiste en la generación de un algoritmo que estime la hora de llegada de vehículos.
3. Simulaciones a través de herramientas de software matemático.

1.2. Objetivos

El objetivo principal de este proyecto consiste en gestionar los tiempos de llegada de vehículos a puntos de verificación sobre rutas de transporte, usando datos de posicionamiento global. Los siguientes objetivos específicos son también realizados:

- Analizar los métodos de gestión de tiempo en transporte existentes para proponer una mejora respecto a éstos.
- Desarrollar una aplicación que se ajuste a las plataformas de software libre que trabajen en la gestión de datos en transporte.
- Realizar pruebas de funcionamiento del software matemático para el análisis de su desempeño.

1.3. Estructura del documento

El documento está formado por cuatro capítulos. El capítulo 2 presenta una revisión de la literatura concerniente a la predicción de tiempos de llegada en rutas de autobuses. La descripción del método desarrollado se realiza en el capítulo 3. En el capítulo 4 se muestran la implementación de la solución y las pruebas para verificar su rendimiento. En el capítulo 5 se concluye y además se bosquejan algunas posibles líneas de trabajo futuro.

Capítulo 2

Revisión Literaria

Se considera como gestión de tiempo en flotas de transporte de pasajeros y de carga, a la provisión de información precisa de navegación y a la predicción de la hora de llegada de los vehículos. Estas son dos cuestiones muy valiosas para los usuarios finales y para los departamentos de operación de transporte. En este capítulo, se citan varios métodos dentro de este contexto.

2.1. Características técnicas de los sistemas de información de llegada de vehículos en tiempo real

En esta sección se sintetiza información pertinente desde el documento de sistemas de información de llegada de autobuses en tiempo real realizado dentro del Programa de Investigación Cooperativa de Tránsito, en [1].

2.1.1. Tecnología subyacente de ubicación automática de vehículos

Consiste en sistemas de localización automática de vehículos (AVL, Automatic Vehicle Location), necesarios para determinar la información de llegada en tiempo real. Los sistemas AVL brindan información sobre la ubicación de cada vehículo y a veces, sobre velocidad, dirección

y cumplimiento del horario.

Dos tecnologías AVL muy utilizadas para determinar la ubicación del vehículo son el sistema de posicionamiento global (GPS, Global Positioning System) y el GPS diferencial (DGPS, Differential GPS). Además, se puede contar con sistemas de postes de señalización/balizas, a través de los cuales cada vehículo envía datos de ubicación y/u odométricos al despacho.

La frecuencia de actualización de la ubicación es fundamental para la precisión de las predicciones de llegada en tiempo real. En general, la información de ubicación se transmite desde cada vehículo a un sistema central en períodos que oscilan entre 30seg y 5min.

Además de proporcionar la información necesaria a los sistemas de información de llegada de autobuses en tiempo real, los sistemas AVL ofrecen muchos beneficios operativos y pueden integrarse con otros sistemas de transporte inteligente (ITS, Intelligent Transportation Systems). Esta integración permite proporcionar información vital (por ejemplo, desde contadores automáticos de pasajeros) y controlar la operación del vehículo (por ejemplo, darle prioridad sobre el resto de vehículos al manipular inteligentemente los semáforos). Estos beneficios incluyen facilitar el análisis del desempeño del servicio de tránsito en tiempo real e históricamente; recopilar la información necesaria para realizar la planificación del sistema, como los tiempos de ejecución y la programación; y proporcionar ubicaciones de vehículos para la gestión de emergencias.

2.1.2. Distribución de la información en tiempo real

El medio más utilizado para la distribución de información de llegada de autobuses en tiempo real es la señal electrónica, también conocida como señal de mensaje dinámico (DMS, Dynamic Message Sign), ubicada en cada parada de autobús. De todos los tipos de letreros electrónicos disponibles, el letrero LED es el más frecuente, seguido por el letrero LCD.

Un método adicional para proporcionar información de autobuses en tiempo real es mostrar las ubicaciones reales de los vehículos en un mapa del área de servicio que está disponible a través de internet y/o quioscos. Este método no proporciona información de llegada en tiempo real, pero requiere menos datos y es un método visual para mostrar a los clientes dónde

se encuentran actualmente sus vehículos. Los quioscos interactivos de pantalla táctil no sólo muestran un mapa de la ruta del vehículo y su hora estimada de llegada a esa parada, sino que se pueden utilizar para acceder a información meteorológica e información sobre los servicios y tarifas de la agencia. Además, la hora de llegada se proporciona en formato de audio. Las personas con discapacidad visual pueden simplemente presionar un botón en el quiosco para escuchar los tiempos estimados de llegada del autobús.

En general, los siguientes tipos de información mostrados en letreros electrónicos son los más frecuentes:

- Hora y fecha actuales
- Número de ruta y destino final del vehículo
- Tiempo de espera, ya sea en formato de cuenta regresiva o rango de tiempo
- Interrupciones del servicio u otros mensajes importantes

Las tecnologías de comunicación que se utilizan con mayor frecuencia para transmitir información a letreros electrónicos en las paradas de autobús son las comunicaciones celulares, principalmente datos en paquetes digitales celulares (CDPD, Cellular Digital Packet Data). Otras tecnologías en expansión incluyen comunicaciones dedicadas de corto alcance (por ejemplo, balizas) y líneas de red digital de servicios integrados a través de cable o fibra óptica. Tener una cobertura de comunicaciones adecuada en el área de servicio es fundamental para el éxito de la comunicación de información en tiempo real hacia los dispositivos de visualización.

Además de los letreros electrónicos en las paradas de autobús, actualmente existe la tendencia de mostrar la información en tiempo real a través de páginas web e incluso por medio de aplicaciones móviles.

2.1.3. Uso de la información generada por el sistema

Una de las principales preocupaciones sobre la implementación de varios tipos de tecnologías de tránsito es poder usar de manera efectiva la información que genera cada tecnología.

Es decir, existe la interrogante de si la información generada por los sistemas tiene un uso más allá de proporcionar la información al público, por ejemplo, para planificar mejor sus servicios.

En general, la información se utiliza para optimizar el servicio y las operaciones de tránsito. El uso de la información puede resultar en varias acciones:

- Realizar la planificación general del tránsito, que puede incluir planificación de operaciones, servicios, finanzas y gestión, y para desarrollar nuevos servicios.
- Cambiar los intervalos de ruta y/o los horarios.
- Modificar las estructuras de las rutas.
- Aumentar o disminuir el número de autobuses que sirven en una ruta.
- Mejorar la confiabilidad de los autobuses.
- Planificar nuevos servicios, tales como expandir el sistema a otras rutas y realizar envíos en tiempo real.

Otros usos de los datos incluyen monitorear el comportamiento de los conductores e investigar los comentarios de los clientes.

2.2. Factores que afectan a la predicción de la hora de llegada

La predicción de la hora de llegada depende de algunas variables, las más representativas se muestran en la Figura 2.1. La velocidad tiene gran impacto sobre el tiempo de llegada del vehículo, y a la vez depende de las condiciones de la carretera, de las condiciones climáticas y de los factores humanos. El tráfico es un factor impredecible pero tiene efecto sobre la llegada. La distancia se refiere a la separación en la carretera entre diferentes puntos de control o paradas, el tiempo de viaje se considera entre el inicio y el destino de la ruta, el tiempo de permanencia considera el tiempo perdido por el automotor sin cambiar de posición.

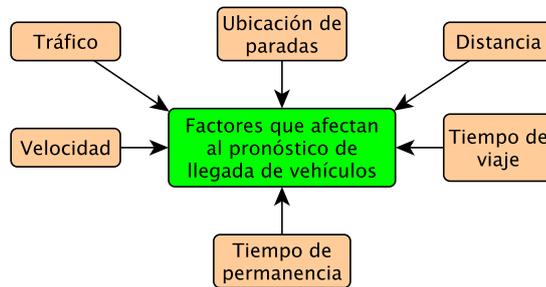


Figura 2.1: Factores que afectan a la predicción de la hora de llegada

2.3. Modelos de pronóstico de la hora de llegada

Muchas aplicaciones de sistemas AVL de identificación automática de vehículos (AVI, Automatic Vehicle Identification) y de conteo automático de pasajeros (APC, Automatic Passenger Counters), están recibiendo cada vez más atención en la gestión del transporte. Estos se consideran como los componentes clave de los sistemas ITS.

Recientemente, las agencias de tránsito se han dado cuenta de los beneficios operativos de la implementación de ITS. Por ejemplo, con base en las tecnologías detalladas, pueden adquirir información de autobuses en tiempo real para reducir el tiempo de viaje de los pasajeros y mejorar el nivel de gestión/servicio. Es así que existe un interés creciente en proporcionar información a los usuarios sobre la llegada de vehículos de pasajeros, a través de tecnologías emergentes de información y comunicación electrónicas. La disponibilidad de información en tiempo real puede ayudar a los pasajeros a programar de manera eficiente su hora de salida y tomar decisiones inteligentes para su viaje.

A través de la Figura 2.2 se organizan varias aproximaciones de pronóstico de llegada de vehículos encontradas en la literatura, las cuales son descritas en las subsecciones a continuación.

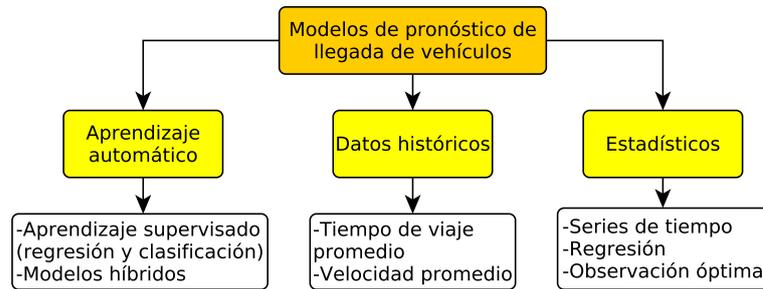


Figura 2.2: Modelos de pronóstico y subcategorías

2.3.1. Modelos de aprendizaje automático

En [2] se utilizan tres modelos para determinar los tiempos de llegada estimados (ETA, Estimated Times of Arrival): red neuronal artificial (ANN, Artificial Neural Network), regresión y un modelo estadístico basado en datos históricos. La predicción se basa en el tiempo de permanencia y en la congestión del tráfico; los principales parámetros utilizados son el tiempo de llegada, el tiempo de permanencia, el cumplimiento del horario, el tiempo de viaje y la distancia. El modelo ANN presenta mejores resultados con respecto al error porcentual absoluto medio (MAPE, Mean Absolute Percentage Error).

En [3] se presenta una técnica que utiliza un conjunto de modelos ANN para pronosticar la hora de llegada en función de diferentes condiciones meteorológicas. La minería de datos se usa para cambios estacionales y ANN para pronosticar el tiempo de llegada.

En [4] se predice el tiempo de llegada del autobús mediante el uso de una ANN y un filtro de Kalman. El modelo de predicción está en función del índice de tiempo, del retraso del autobús y del tiempo de viaje. El histórico de información proviene de la recopilación de datos GPS. La ausencia de datos se compensa a través del sistema de recolección automática de boletos (AFC, Automated Fare Collection). Modelos sub-ANN se utilizan para cada grupo de datos recopilados y todo se integra en un modelo jerárquico. Se concluye que ANN funciona mejor que el filtro de Kalman.

En [5] se desarrolla un modelo híbrido que combina la tenacidad de ANN con la fiabilidad del filtro de Kalman. Este modelo predice la hora de llegada del autobús a una parada individual

dentro de una ruta. Combina los datos tomados desde receptores GPS tales como ubicación y tiempo, y registra la velocidad media teniendo en cuenta la velocidad histórica de viaje. El resultado muestra la precisión satisfactoria en la predicción.

En [6] se presenta un modelo basado en máquina de vectores de soporte (SVM, Support Vector Machine) para la previsión de la hora de llegada del autobús. Este método utiliza tres índices nuevos que incluyen cobertura GPS, tasa de liberación y tasa de precisión para evaluar el servicio de predicción, y luego usa SVM para entrenar el modelo para la evaluación de múltiples índices. Los resultados muestran que el método es intuitivo y completo y que puede resolver el problema con precisión.

En [7] se predice el tiempo de viaje del autobús al incluir un factor de olvido que asigna peso a los datos recientes resultantes de las cantidades variables basadas en el tiempo de funcionamiento del bus. Se considera que la velocidad del autobús replica las condiciones del tráfico. Se aplica la prueba de Grubbs para eliminar valores atípicos de los datos de entrada. El modelo propuesto se evalúa con los datos de una ruta real y los resultados muestran que la SVM con el factor de olvido presenta ciertos beneficios en comparación con ANN con la prueba de Grubbs, y con el filtro de Kalman.

2.3.2. Modelos basados en datos históricos

Se utilizan para predecir el tiempo de viaje actual y futuro del autobús usando como base el tiempo de viaje histórico tomado de jornadas anteriores. Los modelos son más confiables en las áreas donde el patrón de tráfico es reactivamente estable, lo que significa que la congestión es mínima. Estos modelos realizan su trabajo usando el tiempo promedio de viaje y principalmente la velocidad promedio.

En [2], la predicción se basa en el tiempo de permanencia y en la congestión del tráfico según el modelo histórico. Se hace énfasis en la ubicación del bus y en la marca de tiempo como características principales [9]. Se establece que el modelo de datos históricos se desempeña 2.5 veces mejor que un modelo de regresión ANN y 2 veces mejor que un modelo de regresión SVM, logrando una precisión del 75.65%.

En [14] la hora prevista de llegada del autobús se muestra en un tablero electrónico inteligente. De acuerdo con el tiempo de llegada, el viajero elige la línea más corta. El modelo está diseñado para calcular el tiempo de viaje por sección, el tiempo de demora en las paradas y el tiempo de llegada del autobús utilizando un modelo dinámico para cada componente. Todos los componentes predicen colectivamente la hora de llegada del autobús con mayor precisión. Después de la predicción, el tiempo de viaje se muestra en tableros en cada estación.

En [15] se desarrolla un modelo dinámico para la predicción del tiempo de llegada del autobús utilizando datos GPS. El modelo dinámico pronostica el tiempo de llegada del autobús usando la distancia entre dos paradas y los parámetros históricos del tiempo de viaje. Al comparar el tiempo de predicción con el tiempo de llegada real, resulta que el error para 16 paradas es menor al 2%.

2.3.3. Modelos estadísticos

Los modelos de regresión predicen las variables dependientes como función lineal de las variables independientes. Calculan el efecto simultáneo de factores que son independientes entre sí y afectan a las variables dependientes. Considerando que el tiempo de llegada del autobús también se ve afectado por el comportamiento del conductor, las señales de tránsito, la congestión del tráfico, etc., estos son factores tratados como variables independientes por muchos investigadores. Además, estos modelos utilizan el tiempo de permanencia, la distancia, el número de paradas, los descriptores meteorológicos, etc., también como variables independientes.

En [10] se presenta un modelo de regresión automática de series de tiempo. El tiempo de viaje se divide en tres partes: tiempo de viaje libre, tiempo de retraso de parada, y tiempo de retraso de intersección de carreteras. El tiempo de viaje se pronostica utilizando la fusión del modelo de retraso con modelos de promedio integrados auto regresivos y modelos de intersección y retraso. La tasa de error encontrada es inferior al 20% en comparación con la hora de llegada real y la hora prevista.

En [11] se proponen tres algoritmos: regresión lineal, predicción del k-vecino-más-cercano (kNN, k-Nearest Neighbor) y regresión de kernel. Estos algoritmos se basan en el tiempo de

viaje, la ubicación, la distancia entre paradas y el retraso. La predicción del tiempo de llegada del autobús utilizando el modelo de regresión de kernel proporciona un rendimiento sólido en comparación con el kNN.

Los modelos con observación óptima se utilizan cuando se tiene alguna evidencia incierta sobre el sistema dinámico y es fácil hacer una suposición fundamentada de lo que se realizará a continuación con el sistema. La función básica del modelo proporciona alguna estimación del sistema actual, así como también predice el valor futuro de las variables. En este contexto, el filtro de Kalman se usa en [12] para pronosticar la hora de salida y de llegada del autobús usando el sistema APC. Se utilizan dos filtros de Kalman, uno para la duración de la ruta y otro para calcular el tiempo de llegada del pasajero a su destino. Este modelo predictivo tiene un mejor rendimiento que el modelo de regresión y que el ANN.

En [13] se aplica el filtro de Kalman utilizando parámetros de tiempo de permanencia, tiempo de funcionamiento, velocidad y ubicación, para predecir la hora de llegada del autobús. Se divide el tiempo total de viaje en dos componentes principales denominados tiempo de ejecución y tiempo de permanencia. La estimación del tiempo de viaje se realiza mediante el filtro de Kalman; la estimación del tiempo de permanencia se realiza mediante la estimación del tiempo de permanencia a posteriori y el factor de ganancia. El tiempo de llegada estimado se compara con los valores reales, así, este modelo muestra una tasa de error menor al 16.7%.

Capítulo 3

Metodología

Este capítulo detalla el sistema desarrollado. Se enfoca en dos partes importantes: la descripción del escenario de trabajo de los autobuses y la formulación de la propuesta para gestión de tiempo.

3.1. Breve descripción de la propuesta

El esquema de la Fig. 3.1 muestra el método de rastreo sobre el cual se soporta el presente proyecto. El autobús obtiene sus datos de navegación a través de la tecnología GPS. Los mismos son enviados mediante internet móvil (usando red celular) hacia un servidor remoto que podría ser físico o virtual (web hosting). En el servidor corre la aplicación que monitorea la navegación de los vehículos y gestiona su tiempo, que podría ser una página web. Los usuarios finales pueden conectarse a través de computadores o dispositivos móviles hacia la aplicación en el servidor con el objetivo de visualizar los resultados de navegación y de gestión.

En la Fig. 3.2 se observa el diagrama de bloques que describe brevemente la propuesta. Los *dispositivos de rastreo* colocados en los N buses, distribuidos en diferentes líneas, transmiten sus datos de navegación hacia una *base de datos en tiempo real* disponible en la nube. Desde ésta, el *programa para gestión de tiempo* toma la información necesaria para ejecutar su procesamiento. El *cuadro de trabajo* indica la distribución diaria de los buses en las diferentes rutas o líneas

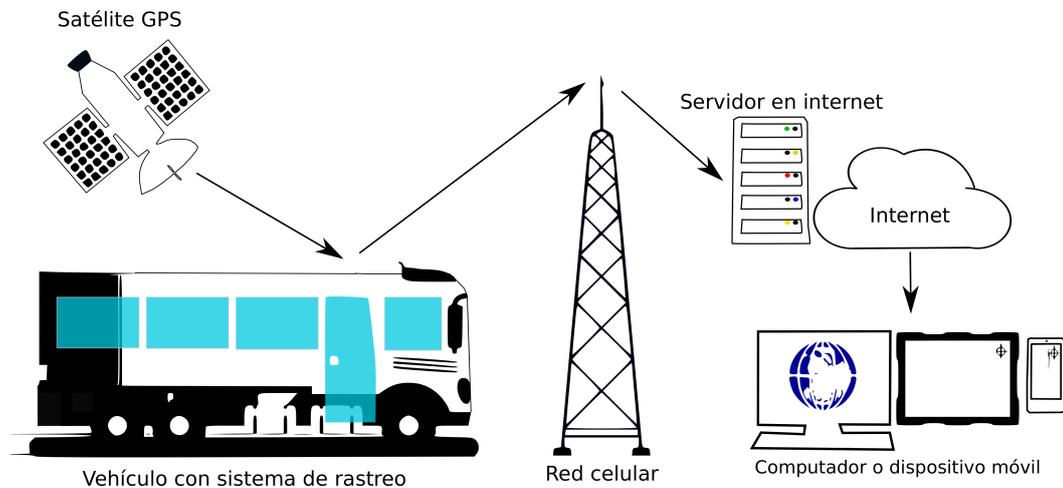


Figura 3.1: Método de rastreo satelital

de trabajo. El *itinerario de cada ruta* consiste en el detalle de las paradas por las que pasa cada autobús en una línea en particular (latitud y longitud). La *planificación de cada ruta* contiene información de las horas de inicio de cada trayecto y de los períodos de tiempo que los autobuses tienen para trasladarse de una parada a la siguiente.

En cuanto al *programa para gestión de tiempo*, éste es una aplicación web que corre en un computador local o sobre un servidor en la nube. El programa cumple con dos funciones: *visualización en el mapa*, y *gestión de tiempo*. La primera función muestra sobre un mapa virtual el trazado de las rutas de los autobuses, además de indicar la posición actual de cada uno de éstos. La segunda, comprende la determinación de los estados de navegación de los autobuses, la predicción de las horas de llegada a cada parada, y la visualización de mensajes de información para los usuarios.

3.2. Adquisición de datos en tiempo real

Si bien los dispositivos de rastreo y la base de datos no forman la parte central de la solución presentada, es necesario hablar de ellos dentro del contexto del proyecto. Esto, debido a que participan en la conformación del escenario de trabajo sobre el cual se desarrolla la presente

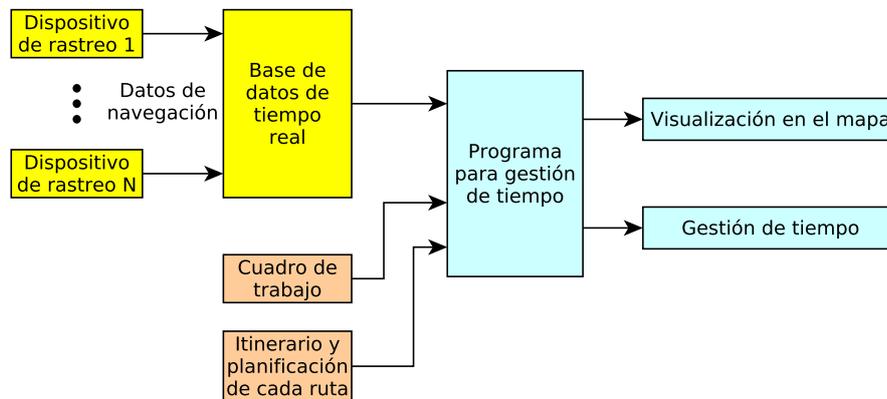


Figura 3.2: Diagrama de bloques de la propuesta

propuesta.

3.2.1. Dispositivos de rastreo

Los dispositivos de rastreo toman la información de navegación adquirida a través de la tecnología GPS, y la envían hacia la base de datos usando internet móvil.

Cada dispositivo se basa en hardware/software libre y ejecuta un script de Python 3 [16], siguiendo el Algoritmo 1. Los datos de navegación son obtenidos con una frecuencia máxima de 1Hz desde un módulo GPS. De estos datos, sólomente ciertos campos son escogidos, como se detalla en la Tabla 3.1. Posteriormente, se crea una trama JSON (JavaScript Object Notation) que es enviada a través de internet móvil a la base de datos. El tiempo de muestreo (periodo) con el cual se ejecuta este algoritmo es ajustable.

En cuanto a la base de datos de tiempo real, se usa el formato ligero de intercambio de datos JSON [17]. Éste es un formato de texto completamente independiente del lenguaje, pero utiliza convenciones que son familiares para diferentes lenguajes, entre los que se encuentran JavaScript y Python. Se usa el método PATCH para la actualización de datos, que consiste en actualizar los datos en la base, borrando los anteriores para el mismo usuario. Cada vehículo corresponde a un usuario.

Es necesario indicar que la comunicación entre el script de Python y la base de datos de

Algoritmo 1 Datos de navegación enviados a la base de datos

```
1: procedure GPS_DB ▷ Rutina con periodo ajustable
2:   datosGPS ← lee_GPS()
3:   tramaJSON ← GPS_a_JSON(datosGPS)
4:   mensaje_hacia_DB(tramaJSON)
```

Tabla 3.1: Campos de la trama de datos JSON enviada por los dispositivos de rastreo

| Campo | Identificación | Latitud | Longitud | Velocidad | Hora |
|-----------------------|-----------------------|------------------------|------------------------|------------------|-----------------------|
| Nombre en JSON | id | lat | lng | speed | time |
| Ejemplo | 1 | 0.3552967 965965472 | -78.127613 26316692 | 20.2 | 1626882 120.903906 |

tiempo real se realiza gracias al uso de la librería Python Firebase, disponible en [18]. Consiste en una interfaz de programación de aplicaciones (API, Application Programming Interface), que es un conjunto de instrucciones de programación para acceder a esta base de datos basada en web.

3.2.2. Base de datos de tiempo real

La sincronización en tiempo real permite que las aplicaciones (usuarios) accedan a los datos desde cualquier dispositivo, web o móvil, con facilidad, y los ayuda a trabajar en conjunto. En cuanto a la base de datos, se usa Firebase Realtime Database [19] que es NoSQL, y entre cuyas características más relevantes destacan las siguientes:

- Posee sincronización en tiempo real para datos JSON.
- Se aloja en la nube y permite almacenar y sincronizar datos entre diferentes dispositivos en tiempo real.
- Tiene un kit de desarrollo de software (SDK, Software Development Kit) para dispositivos móviles y web, para que se puedan crear aplicaciones sin la necesidad de servidores.
- Se puede ejecutar código de backend que responda a eventos activados por la base de datos.

Como todas las demás API específicas de transferencia de estado representacional (REST, Representational State Transfer), Firebase ofrece un cliente para actualizar (PATCH, PUT), crear (POST) o eliminar (DELETE) los datos almacenados.

3.3. Cuadro de trabajo de los autobuses

Las empresas de transporte en Ibarra-Ecuador se organizan a través del cooperativismo, donde cada autobús pertenece a un propietario asociado. Esto influye fuertemente en la dinámica de trabajo al punto en que para cada línea, el despacho de los autobuses se realiza respetando un horario fijo de partida y siguiendo un itinerario inflexible de llegada a cada parada.

En el esquema mencionado las decisiones de despacho no se toman en función de la optimización de tiempo y combustible, tampoco en relación a la demanda de servicio. Por el contrario, la gestión se orienta a la prevención y control de retrasos intencionales de cada autobús¹. Para detallar el escenario de despacho, a continuación se realiza una descripción formal del cuadro de trabajo.

Sea $n \in \{1, 2, 3, \dots, N\}$ la identificación de un autobús dentro de la compañía de transporte que comprende un máximo de N buses, a_n representa a un autobús dentro de la empresa. Además, considere $i \in \{1, 2, 3, \dots, I\}$ como la identificación de las I líneas de trabajo (trayectorias) en las que podría dar servicio el autobús a_n . Entonces, a_n^i representa al autobús de identificación (disco) n que sirve en la línea i .

Una jornada de trabajo puede tener una extensión en el rango de horas. La condición implícita en la identificación a_n^i es que el autobús a_n sólo sirve a la línea i a lo largo de una jornada de trabajo. Por ejemplo, la matriz

$$Q = \begin{bmatrix} a_1^1 & a_6^1 & a_{10}^1 & 0 \\ a_2^2 & a_5^2 & a_9^2 & 0 \\ a_3^3 & a_4^3 & a_7^3 & a_8^3 \end{bmatrix} \quad (3.1)$$

muestra una flota de $N = 10$ buses (elementos no nulos de la matriz), distribuidos en $I = 3$

¹El retraso intencional de un autobús permite esperar a que más pasajeros lleguen a las paradas, perjudicando al siguiente autobús.

líneas. Los autobuses 1, 6 y 10 trabajan en la línea 1. Los autobuses 2, 5 y 9 sirven en la línea 2. Finalmente, los autobuses 3, 4, 7 y 8 trabajan en la línea 3. Además, en cada fila de la matriz Q , los buses se encuentran ordenados convenientemente de izquierda a derecha, indicando el orden de partida. Por ejemplo, en la línea 1, el bus 1 empezará la ruta. Tendrá que pasar cierto tiempo para que el bus 6 empiece la trayectoria. Posteriormente, partirá el bus 10.

3.4. Itinerario y planificación de cada ruta

Sea $s \in \{1, 2, 3, \dots, S\}$ la identificación numérica de cada parada dentro de una línea o trayecto. El número 1 indica el inicio del trayecto, mientras que S identifica su final. La variable p_s^i representa a la parada s que pertenece a la línea i . Por ejemplo, la matriz

$$P = \begin{bmatrix} p_1^1 & p_2^1 & \cdots & p_8^1 \\ p_1^2 & p_2^2 & \cdots & p_7^2 \\ p_1^3 & p_2^3 & \cdots & p_9^3 \end{bmatrix} \quad (3.2)$$

muestra en cada una de sus 3 filas el conjunto de paradas por línea. En la primera fila que corresponde a la línea 1 se tiene un conjunto de 8 paradas. En la segunda fila, para la línea 2 se tiene 7 paradas. La línea 3 está constituida por 9 paradas.

Por otro lado, el itinerario de inicio del trayecto muestra la separación temporal entre los buses ordenados consecutivamente en el cuadro de trabajo. En su concepción más sencilla consiste en un período de tiempo fijo que se respetará a lo largo de la jornada para una línea en particular. Por ejemplo, para la matriz Q en (3.1), la matriz de tiempos para inicios de trayecto es

$$D = \begin{bmatrix} d^1 \\ d^2 \\ d^3 \end{bmatrix}, \quad (3.3)$$

donde sus elementos representan el tiempo en segundos cada el cual parte un vehículo dentro la línea. Así, considerando (3.1), para la línea 1, el autobús a_1^1 parte en el instante t , el autobús a_1^6 parte en el instante $t + d^1$ y el autobús a_1^{10} parte en el instante $t + 2 \cdot d^1$.

La planificación de la ruta contiene las horas programadas de llegada a cada una de las paradas p_s^i . Considerando la identificación numérica de cada parada dentro de una ruta usada en (3.2), $t_{s-1,s}^i$ representa la separación temporal (en segundos) entre las paradas p_{s-1}^i y p_s^i , dentro de la ruta i . Por ejemplo, la matriz

$$T = \begin{bmatrix} t_{1,2}^1 & t_{2,3}^1 & \cdots & t_{7,8}^1 \\ t_{1,2}^2 & t_{2,3}^2 & \cdots & t_{6,7}^2 \\ t_{1,2}^3 & t_{2,3}^3 & \cdots & t_{8,9}^3 \end{bmatrix}, \quad (3.4)$$

muestra la planificación de 3 rutas. En la primera fila, que corresponde a la línea 1, un bus que empieza el recorrido a una hora determinada en la parada de inicio p_1^1 , llegará a la parada p_2^1 luego de un tiempo $t_{1,2}$. Pasará por la parada p_3^1 luego de $t_{1,2} + t_{2,3}$ y finalizará en la parada p_8^1 luego de $t_{1,2} + t_{2,3} + \cdots + t_{7,8}$.

3.5. Programa para gestión de tiempo

La solución se plantea como una aplicación web cuya ventaja intrínseca es que se puede ejecutar en un navegador y podría ser accedida localmente o en un servidor remoto.

Para realizar la aplicación web se usa el lenguaje de marcas de hipertexto (HTML, Hyper-Text Markup Language) que es el componente más básico de la Web. Además de HTML, se utiliza JavaScript [20] para diseñar/programar cómo se comporta la página cuando ocurre un evento. De esta manera se asegura contenido interactivo para mejorar la experiencia del usuario y se agrega dinamismo a la página web.

3.5.1. Front end

El lado del usuario procura mostrar una interfaz sencilla de usar, atractiva y funcional. El formulario web básicamente consta de los siguientes elementos:

- *Selección de la ruta* a través de un botón de opción.

- *Selección de la parada* dentro de la ruta seleccionada, a través de un botón de opción.
- *Hora actual* indicada sobre una línea de texto.
- *Mensajes sobre la gestión de tiempo* a través de un campo de texto multilínea.
- *Mapa* que contiene las rutas con sus respectivas paradas. Además, muestra dinámicamente la posición de cada uno de los autobuses.

3.5.2. Back end

De manera convencional el back end de un sitio web consiste en un servidor, una base de datos y una aplicación.

En este caso el servidor no es relevante porque la aplicación web puede ejecutarse localmente. En cuanto a la base de datos, no existe una adosada a la aplicación para ingreso/lectura de la información; sin embargo, se deja como tema pendiente para desarrollos futuros. Por otro lado, sí se usa una base de datos de tiempo real para adquirir la información obtenida desde los dispositivos de rastreo, tal como se detalla en la sección 3.2.

La aplicación consta de procesos que se ejecutan periódicamente y de otros disparadas por eventos, como se muestra en el Algoritmo 2.

Algoritmo 2 Subrutinas de la aplicación y su dinámica de ejecución

- | | |
|---|---|
| 1: procedure HORA_ACTUAL | ▷ Cada 1000ms |
| 2: Muestra hora local | |
| 3: procedure ACTUALIZACIÓN_INFORMACIÓN | ▷ Cada 1000ms |
| 4: Mide y calcula distancias y tiempos de los autobuses en relación a una parada | |
| 5: procedure MENSAJES_GESTIÓN_TIEMPO | ▷ Cada 2000ms |
| 6: Muestra estados y tiempos de llegada de los buses con respecto a una parada | |
| 7: procedure INICIALIZACIÓN_MAPA | ▷ Al iniciar la aplicación |
| 8: Ajusta visualización del mapa y grafica líneas y paradas | |
| 9: procedure ACTUALIZACIÓN_AUTOBUSES | ▷ Disparada por cambio en la base de datos de tiempo real |
| 10: Crea, actualiza o borra un vehículo sobre el mapa | |
| 11: Adquiere los valores instantáneos de navegación desde la base de datos de tiempo real | |
-

3.5.3. Visualización en el mapa

Esta función es desempeñada por las subrutinas disparadas por eventos: *inicialización del mapa* y *actualización de autobuses*; véase el Algoritmo 2. Se apalanca en la API de Javascript de Google Maps [21] que permite integrar las funcionalidades de los mapas en aplicaciones web y dispositivos móviles.

El Algoritmo 3 muestra el proceso de inicialización del mapa en la aplicación. Una característica particular es que las rutas y las paradas de los autobuses son dibujadas a partir de tablas estáticas de latitudes y longitudes.

Algoritmo 3 Inicialización del mapa

- | | |
|---|----------------------------|
| 1: procedure INICIALIZACIÓN_MAPA | ▷ Al iniciar la aplicación |
| 2: Ajusta tamaño | |
| 3: Establece centro | |
| 4: Configura tipo | |
| 5: Dibuja rutas según tablas latitudes y longitudes | |
| 6: Dibuja paradas según tablas latitudes y longitudes | |
-

En el Algoritmo 4 se detalla la actualización de autobuses sobre el mapa. Es importante notar que para que se actualice la información de un autobús, éste se debe encontrar en servicio según el cuadro de trabajo (3.1). Además, al finalizar el algoritmo, los datos tomados desde la base de datos de tiempo real, además de otros calculados, son guardados en un búfer para su uso en la gestión de tiempo. Por otro lado, para disparar este proceso se pueden identificar tres eventos:

- Un bus se adiciona a la base de datos de tiempo real.
- Un bus se desconecta de la base de datos (en este caso se remueve el marcador).
- Existe algún cambio en la información de cualquier vehículo dentro de la base de datos.

Algoritmo 4 Actualización de un bus sobre el mapa

```
1: procedure ACTUALIZACIÓN_AUTOBUSES    ▷ Cambio en la base de datos de tiempo real
2:   Actualiza contador de buses conectados
3:   Extrae datos desde cadena JSON
4:   if bus trabaja hoy then
5:     Calcula orientación
6:     Grafica marcador con posición y orientación en el mapa
7:     Calcula velocidad promedio
8:     Guarda datos de navegación instantánea en el búfer
```

3.5.4. Gestión de tiempo

Esta función es realizada por las subrutinas periódicas: *actualización de la información* y *mensajes de gestión de tiempo*; véase el Algoritmo 2.

A través del Algoritmo 5 se explica el proceso de actualización de la información. Esta subrutina usa la información instantánea recogida por el Algoritmo 4.

Algoritmo 5 Actualización de la información

```
1: procedure ACTUALIZACIÓN_INFORMACIÓN    ▷ Cada 1000ms
2:   Verifica ruta y parada escogidas a través de los botones de opción
3:   Calcula distancia de todos los buses a esa parada
4:   Maneja máquina de estados en función de distancias y tiempos (Figs. 3.3 y 3.4)
5:   Crea mensaje para algoritmo de mensajes
```

El proceso de visualización de mensajes de gestión de tiempo se explica a través del Algoritmo 6. Consiste en una pila descendente de líneas que se actualizan siempre y cuando el mensaje entrante corresponda a un bus nuevo y no a uno del que ya se mostró información previamente. Si el mensaje es sobre un bus recurrente, sólo se actualizará el mensaje previamente existente.

Algoritmo 6 Mensajes de gestión de tiempo

```
1: procedure MENSAJES_GESTIÓN_TIEMPO    ▷ Cada 2000ms
2:   if mensaje sobre un bus existente then
3:     Actualiza el mensaje existente
4:   else
5:     Muestra nuevo mensaje
```

3.5.4.1. Cálculo de los tiempos de llegada

Las marcas de tiempo de un autobús están disponibles cada ciertos segundos para ubicaciones espaciadas irregularmente y son tomadas desde las mediciones GPS. En los datos recopilados desde el bus, k representa la distancia desde el inicio de la ruta en metros y t_k es el tiempo de viaje medido en segundos.

Considerando las paradas p_s^i en (3.2), la matriz en (3.4) contiene los tiempos de separación entre paradas consecutivas. El tiempo de llegada previsto para el bus desde la parada p_1^i hasta la parada p_s^i , a lo largo de la línea i , está dado por

$$t_s^i = t_{1,2}^i + t_{2,3}^i + \cdots + t_{s-1,s}^i. \quad (3.5)$$

Dadas las marcas de tiempo t_1, \dots, t_k para un autobús en particular hasta la ubicación k , el tiempo que necesita el bus para llegar a la parada s desde la actual posición k , se calcula a través de

$$t_{k,s} = t_s^i - t_k. \quad (3.6)$$

Ahora, es necesario saber si el bus se encuentra a tiempo o retrasado y para ello se usa la velocidad promedio \bar{v}_k calculada a partir de las medidas de velocidad tomadas desde los datos GPS, v_1, \dots, v_k , así

$$\bar{v}_k = \frac{1}{n} \sum_{j=1}^k v_j. \quad (3.7)$$

La distancia entre la ubicación actual del bus k y la próxima parada p_s^i a lo largo de la ruta se define como

$$d_{k,s} = p_s^i - k. \quad (3.8)$$

En su forma más simple, el tiempo estimado de llegada se calcula como

$$\hat{t}_{k,s} = \frac{d_{k,s}}{\bar{v}_k}. \quad (3.9)$$

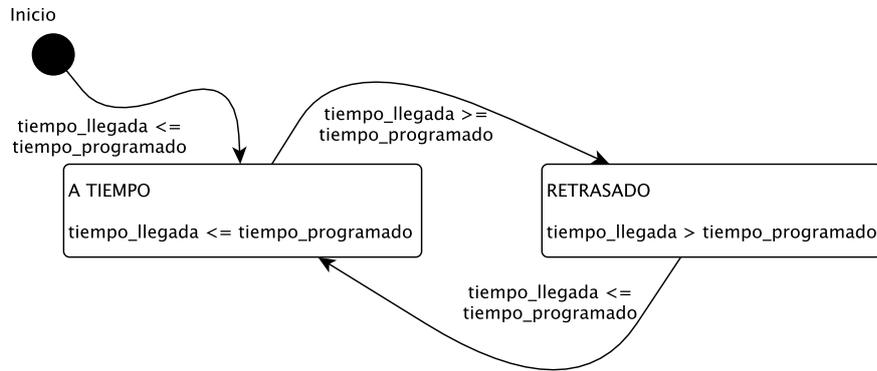


Figura 3.3: Máquina de estados en base al tiempo

Finalmente, la condición de retraso del autobús está dada por

$$\hat{t}_{k,s} > t_{k,s}, \quad (3.10)$$

que significa que a la velocidad promedio actual \bar{v}_k , al autobús le tomará un tiempo mayor que el estipulado para llegar a la parada p_s^i . Sin embargo, esta es una condición que puede cambiar dinámicamente si el autobús acelera para recuperar tiempo perdido.

Una suposición implícita de este enfoque es que el retraso actual permanecerá constante durante el resto del viaje. En la práctica podría no ser realista. Por ejemplo, los retrasos a menudo se deben a malas condiciones climáticas, carreteras con mucho tráfico o una gran cantidad de pasajeros que se subieron al autobús. En todos estos casos, es probable que el autobús acumule retrasos adicionales durante el resto del viaje.

3.5.4.2. Estados de los autobuses

La gestión de tiempo de los autobuses es ilustrada a través de la máquina de estados de la Fig. 3.3. Si el tiempo de llegada calculado es mayor al tiempo de llegada programado, significa que el bus pasa al estado *retrasado*. Podría volver al estado *a tiempo* si el tiempo de llegada es menor o igual que el tiempo programado.

En relación a la distancia del bus a la parada p_s^i , si esta es mayor que el primer umbral

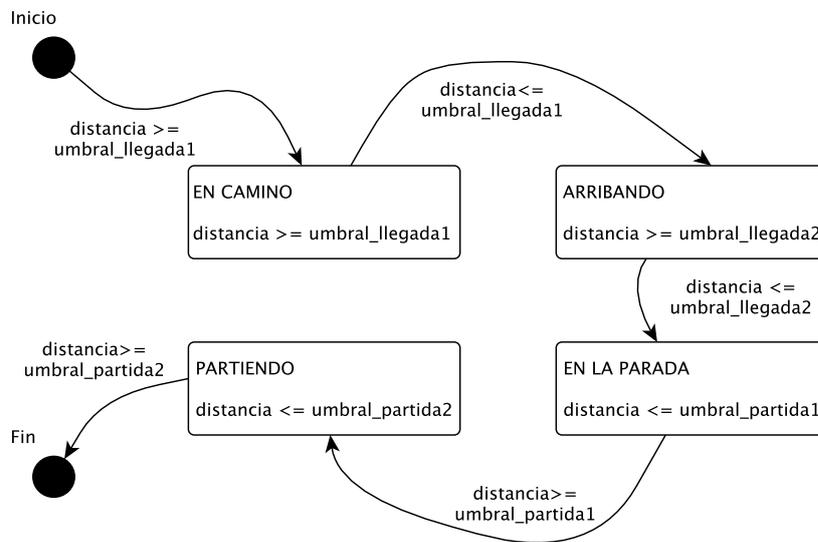


Figura 3.4: Máquina de estados en base a la distancia

de llegada, el bus se encuentra en el estado *en camino*; al estado *arribando* se puede pasar siempre y cuando la distancia sea menor que este umbral. Posteriormente, se pasa al estado *en la parada* cuando la distancia es menor que el segundo umbral de llegada y subsecuentemente al estado *partiendo* cuando la distancia se incrementa y sobrepasa el primer umbral de partida. Finalmente, cuando la distancia sobrepasa el segundo umbral de partida este bus no volverá a gestionarse en esta parada hasta que no termine toda la ruta.

La Fig. 3.5 muestra la relación entre los umbrales de distancia mostrados en la Fig. 3.3. A pesar de que sus valores se pueden modificar, deben mantener la misma relación indicada en la gráfica.

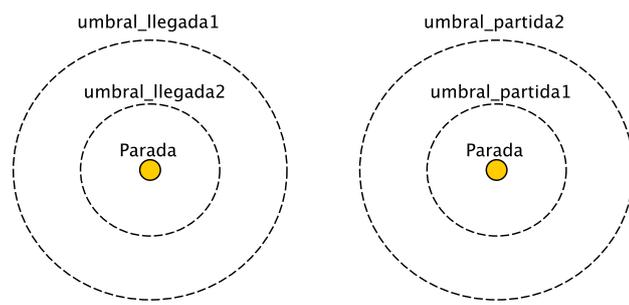


Figura 3.5: Relación entre umbrales de distancia

Capítulo 4

Resultados

En este capítulo, el método propuesto se materializa a través de una aplicación web realizada con HTML y JavaScript que se conecta con una base de datos de tiempo real en Firebase¹; se describe de una manera general la implementación realizada. Además, se explican las pruebas para comprobar su rendimiento y se analizan los resultados.

4.1. Implementación

4.1.1. Aplicación Web

La parte de interacción de la aplicación web (front end), explicada en la sección 3.5.1 puede apreciarse en la Fig. 4.1. Se distinguen los botones de opción para la selección de rutas y paradas. Además, se observa la línea de texto con la hora y fecha actuales, así como también el tablero para mensajes de gestión de tiempo. Es notorio el mapa que contiene las rutas con sus respectivas paradas y los marcadores de cada uno de los vehículos rastreados.

¹El código completo del proyecto se encuentra disponible en la cuenta de Github del tutor <https://github.com/carlosxavierrosero/Gesti-n-en-rutas-de-transporte>

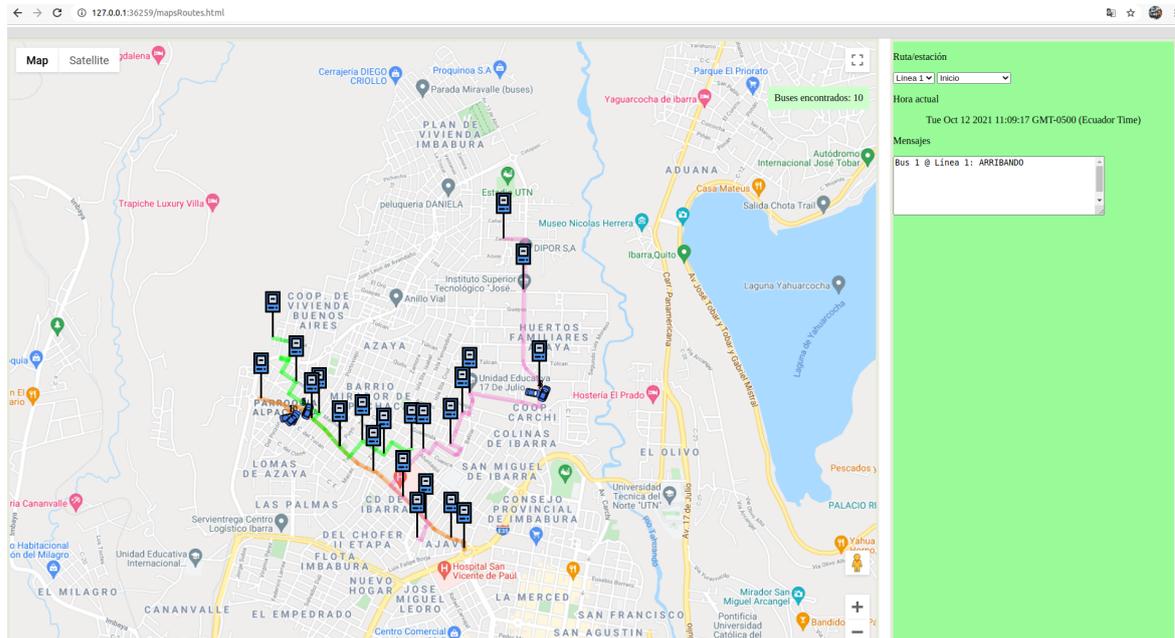


Figura 4.1: Front end de la aplicación web

4.1.2. Base de datos de tiempo real

En la Fig. 4.2 se observa la consola de la base de datos de tiempo real de Firebase alojada en la nube. Como se detalla en la sección 3.2.2, cada dispositivo de rastreo le envía su información de navegación en formato JSON. A la vez, estos datos se sincronizan en tiempo real con la aplicación web.

Para poner en marcha esta base de datos, se sigue la ruta de implementación descrita en [22] que consiste en:

1. Crear un proyecto de Firebase y registrar la aplicación web en éste.
2. Crear una base de datos de tiempo real.
3. Configurar las reglas de la base de datos.
4. Especificar la URL de la base de datos cuando se inicialice la SDK de JavaScript en el código de la aplicación web.

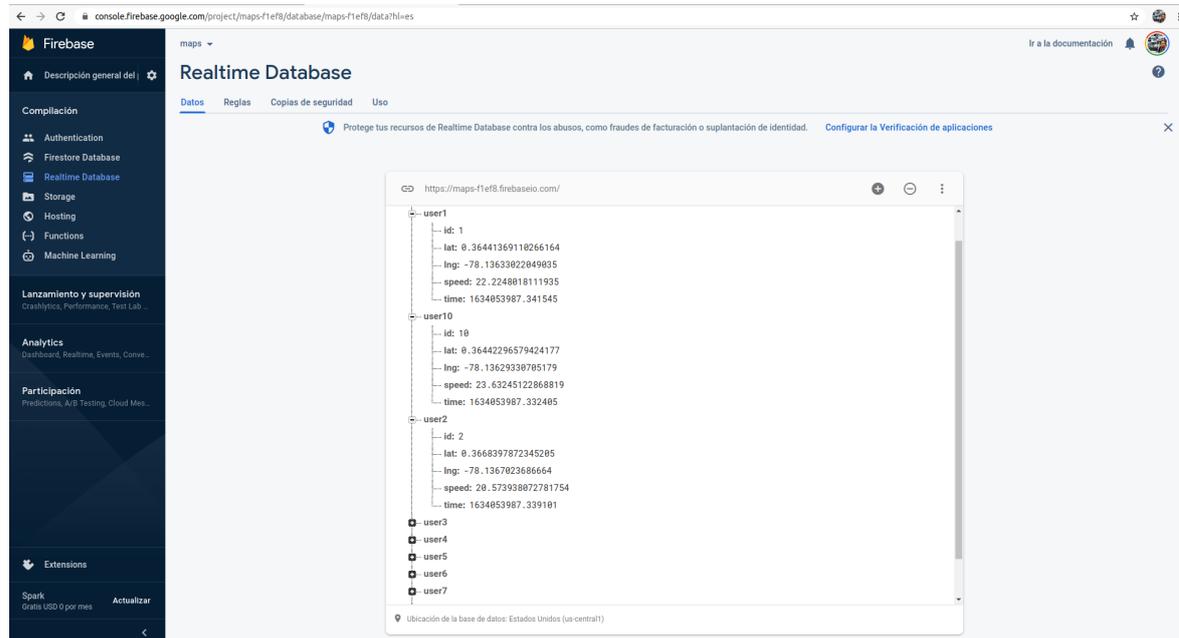


Figura 4.2: Consola de Firebase mientras sincroniza datos

4.1.3. Dispositivo de rastreo

Se utiliza un script de Python, programado con el Algoritmo 4 y corriendo en un computador con sistema operativo Linux. Este programa envía datos de navegación de varios autobuses circulando a lo largo de diferentes rutas. La información consta de los campos detallados en la Tabla 3.1. En la Fig. 4.3 se observa la consola de comandos con las tramas JSON enviadas por el script de Python.

4.2. Pruebas y resultados

4.2.1. Escenario de pruebas

Sobre el mapa de la aplicación web se han trazado 3 rutas de corta longitud, experimentales, con sus respectivas paradas, cuyas características se detallan en la Tabla 4.1. Las coordenadas de estas rutas y paradas se han ingresado de forma estática en el código de la aplicación web.

Las paradas en las 3 líneas se distribuyen usando (3.2), así

$$P = \begin{bmatrix} p_1^1 & p_2^1 & \cdots & p_8^1 \\ p_1^2 & p_2^2 & \cdots & p_7^2 \\ p_1^3 & p_2^3 & \cdots & p_9^3 \end{bmatrix},$$

$$P = \begin{bmatrix} \text{Inicio Manta} & \text{Parque} & \text{Santa Isabel} & \text{IESS} & \text{Juan Pablo II} & \text{Ignacio Burbano} & \text{Final} & 0 \\ \text{Inicio Zumba} & \text{Antinarcóticos} & \text{Parque} & \text{Iglesia} & \text{Santa Isabel} & \text{Final} & 0 & 0 \\ \text{Inicio ITCA} & \text{Ibarra} & \text{Vaca Flores} & \text{San Cristóbal} & \text{Guaranda} & \text{Uruguay} & \text{IESS} & \text{Final} \end{bmatrix}. \quad (4.1)$$

Por otro lado, se tienen 10 autobuses numerados del 1 al 10, distribuidos en las 3 líneas de acuerdo al criterio en (3.1),

$$Q = \begin{bmatrix} a_1^1 & a_6^1 & a_{10}^1 & 0 \\ a_2^2 & a_5^2 & a_9^2 & 0 \\ a_3^3 & a_4^3 & a_7^3 & a_8^3 \end{bmatrix} = \begin{bmatrix} \text{bus 1} & \text{bus 6} & \text{bus 10} & - \\ \text{bus 2} & \text{bus 5} & \text{bus 9} & - \\ \text{bus 3} & \text{bus 4} & \text{bus 7} & \text{bus 8} \end{bmatrix}. \quad (4.2)$$

La matriz de tiempos de inicio de trayecto para los buses ordenados consecutivamente en el cuadro de trabajo Q , sigue el criterio mostrado en (3.3), así

$$D = \begin{bmatrix} d^1 \\ d^2 \\ d^3 \end{bmatrix} = \begin{bmatrix} 600 \\ 600 \\ 300 \end{bmatrix} \text{seg.} \quad (4.3)$$

Finalmente, se rellena (3.4) con la planificación de las 3 rutas con las horas programadas de llegada a cada una de las paradas, tal como se muestra a continuación

$$T = \begin{bmatrix} t_{1,2}^1 & t_{2,3}^1 & \cdots & t_{7,8}^1 \\ t_{1,2}^2 & t_{2,3}^2 & \cdots & t_{6,7}^2 \\ t_{1,2}^3 & t_{2,3}^3 & \cdots & t_{8,9}^3 \end{bmatrix} = \begin{bmatrix} 120 & 90 & 90 & 90 & 90 & 90 & 60 & 0 \\ 160 & 100 & 90 & 90 & 90 & 90 & 0 & 0 \\ 130 & 190 & 160 & 90 & 90 & 120 & 120 & 120 \end{bmatrix} \text{seg.} \quad (4.4)$$

Los elementos con 0 se usan para ajustar las dimensiones de la matriz.

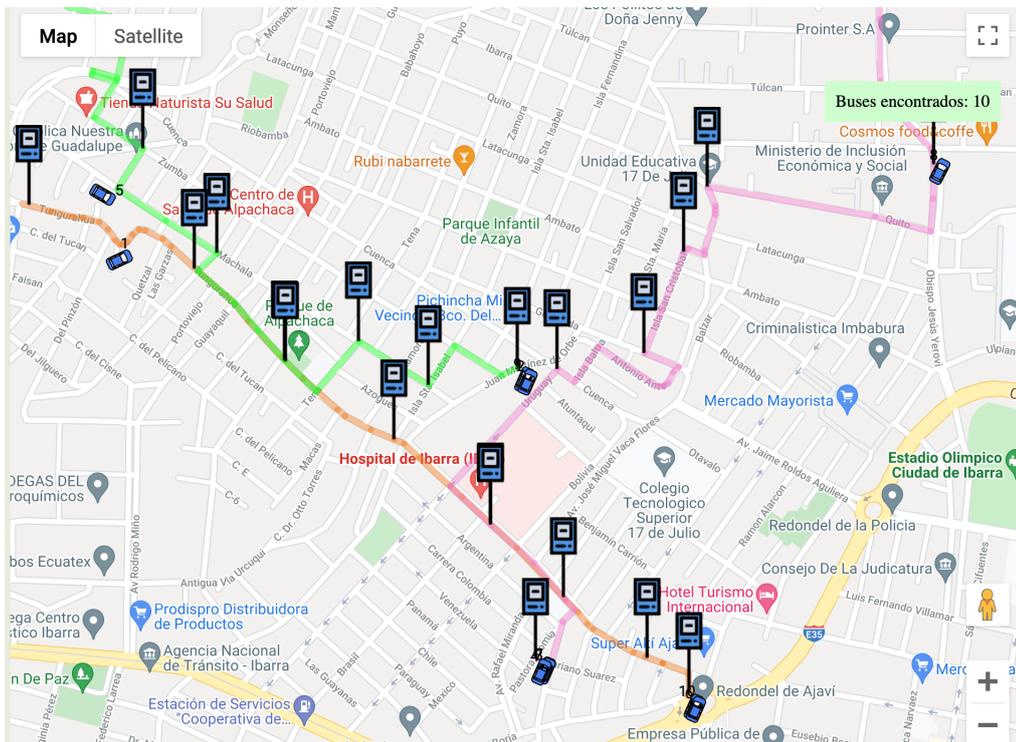


Figura 4.4: Aplicación web en marcha

4.2.2. Pruebas

La prueba de rigor consiste en correr la aplicación y mirar cómo los vehículos son actualizados sobre el mapa en función de sus datos GPS. En la Fig. 4.4 se observa que han sido encontrados 10 buses; los buses 1, 5 y 8, asignados a las líneas 1, 2 y 3 se encuentran recorriendo sus respectivas rutas. Además, las máquinas de estados de tiempo y distancia de las Figs. 3.3 y 3.4, y los umbrales de distancia (Fig. 3.5), son ejecutados adecuadamente por el programa.

4.2.2.1. Cálculo del tiempo de llegada

Las Figuras 4.5, 4.6 y 4.7 muestran las predicciones sobre los recorridos hechos por los buses a lo largo de las rutas 1, 2 y 3, respectivamente. Las líneas cortadas verticales indican la separación en metros entre las diferentes paradas. Las líneas rojas muestran el desplazamiento-tiempo que idealmente recorre el autobús para cumplir con la planificación. Las líneas azules

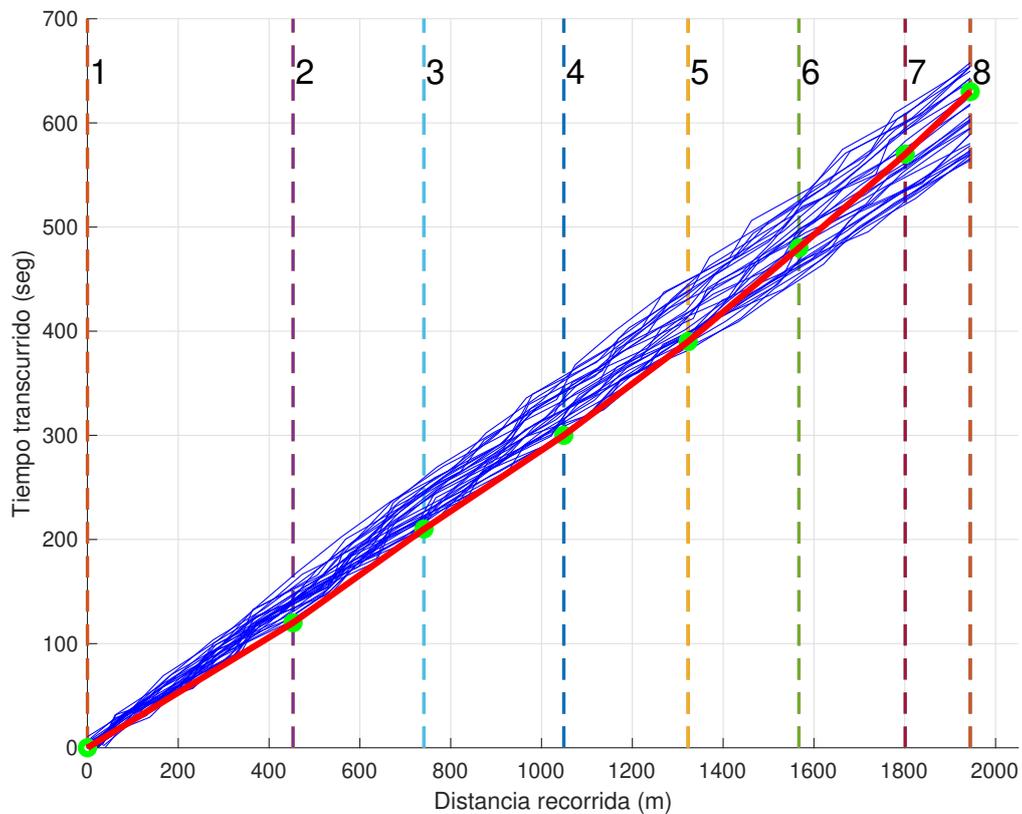


Figura 4.5: Predicciones de llegada para la línea 1

indican las predicciones de tiempos de llegada a las paradas, realizadas instantáneamente para 3 buses recorriendo 10 veces cada ruta. Entre las tres gráficas se representa un total de 90 recorridos.

La Figura 4.8 muestra el error medio absoluto de predicción de tiempo de llegada de 30 recorridos por cada ruta.

4.2.3. Análisis de resultados

En la predicción del tiempo de llegada, se puede notar visualmente la incertidumbre y variabilidad de las predicciones. Hay que considerar que los datos GPS tienen incertidumbre y por lo tanto, la velocidad media calculada en una ventana de tiempo también contiene incertidumbre.

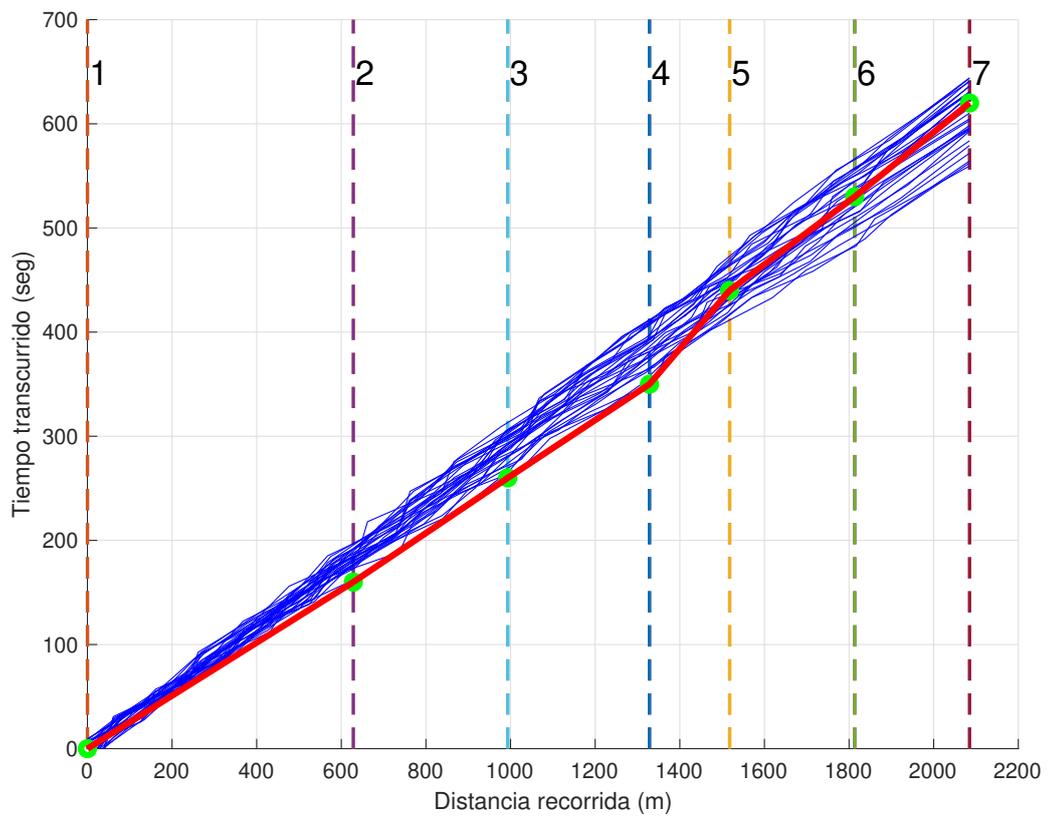


Figura 4.6: Predicciones de llegada para la línea 2

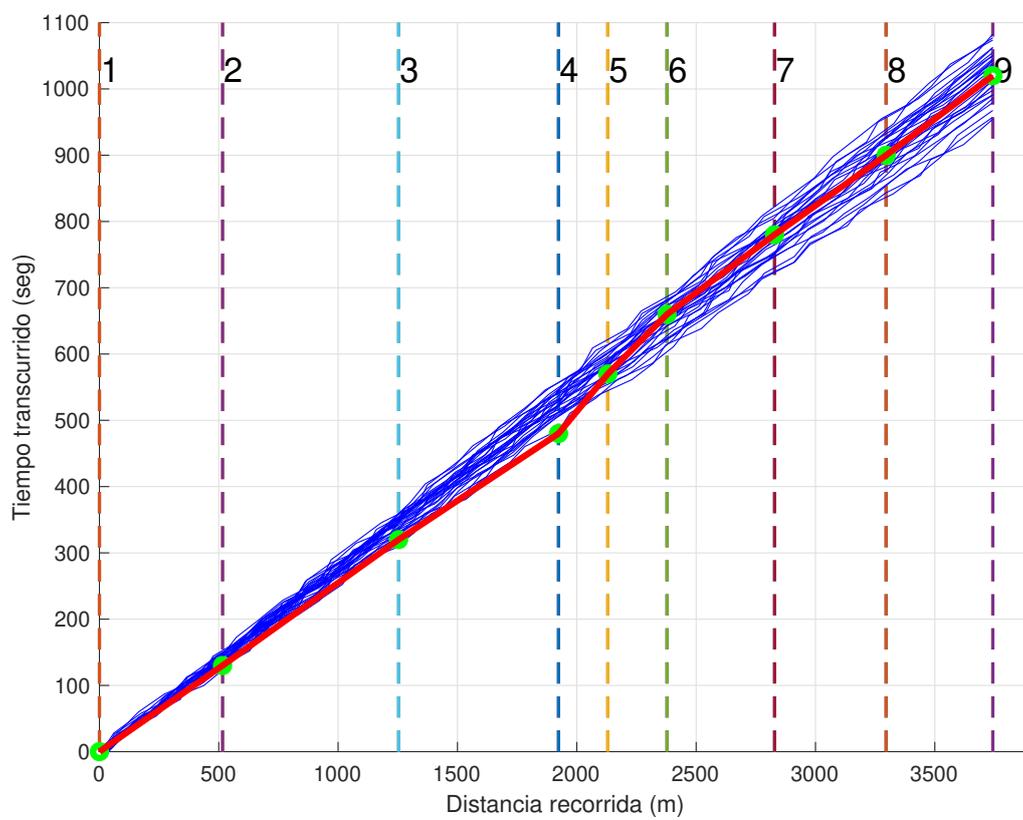


Figura 4.7: Predicciones de llegada para la línea 3

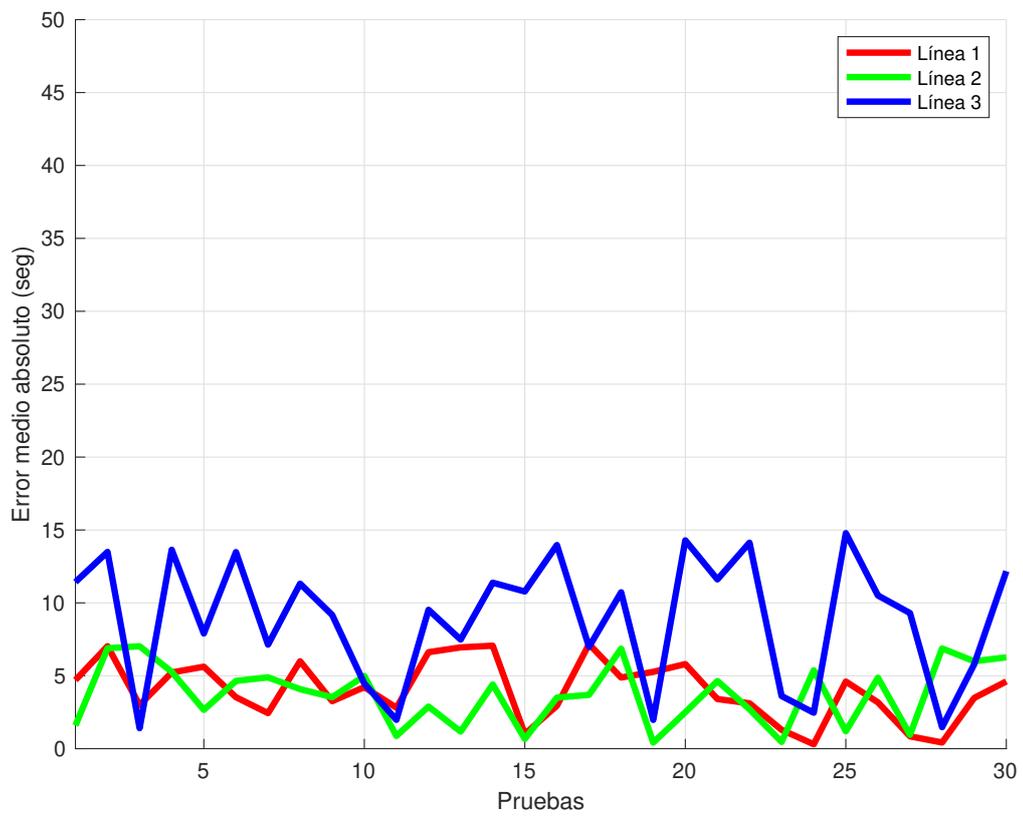


Figura 4.8: Errores de predicción de llegada

Al depender las predicciones de valores ruidosos, también son ruidosas.

Finalmente, se puede ver que el error aumenta cuando el recorrido es más extenso. También se puede notar que, a corto plazo, la predicción es muy estable.

Capítulo 5

Conclusiones y trabajo futuro

Este capítulo muestra las conclusiones del proyecto y esboza algunas líneas posibles de trabajo futuro.

5.1. Conclusiones

Se ha propuesto un método para gestionar los tiempos de llegada de vehículos a puntos de verificación sobre rutas de transporte, usando datos de posicionamiento global.

En este contexto, se han analizado varios métodos de predicción de tiempo de llegada encontrados en la literatura. Así, se ha concebido un método híbrido que organiza la llegada del autobús a la parada a través de una máquina de estados; las condiciones de guarda y de salto dependen de la distancia a la parada y del tiempo estimado de llegada a la misma. Además, para estimar los retrasos se ha planteado un método simple en base a una tabla de horarios de llegada a los puntos de verificación.

Se ha implementado una aplicación web usando software libre. Al someterla a pruebas, se ha determinado su aceptable desempeño en la gestión, tanto de los mensajes referentes a los estados de los autobuses, como en la predicción de los tiempos de llegada.

5.2. Trabajo futuro

El sistema propuesto se podría escalar para mostrar los tiempos de espera, ya sea en formato de cuenta regresiva o de rango de tiempo. Además, se podría gestionar información sobre interrupciones del servicio u otros mensajes importantes.

Las pruebas se podrían escalar a líneas de autobuses que permitan corroborar los resultados obtenidos a través de los experimentos realizados en este estudio.

Bibliografía

- [1] C.L. Schweiger, “TCRP Synthesis 48: Real-Time Bus Arrival Information Systems. A Synthesis of Transit Practice”, Washington: Transportation Research Board, 2003.
- [2] R. Jeong and R. Rilet, (2004), “Bus Arrival Time Prediction Using Artificial Neural Network Model”, in *IEEE Intelligent Transport Systems Conference*, Washington. D.C, USA, 2004.
- [3] R. D. Raut, and V. K. Goyal, “Public transport bus arrival time prediction with seasonal and special emphasis on weather compensation changes using RNN,in *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 1, no. 6, pp. 378-382, 2012.
- [4] Y. Lin, X. Yang, N. Zou and L. Jia, “Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China”, in *Journal of Transportation Engineering*, vol. 139, no. 11, November 2013.
- [5] M. Zaki, I. Ashour, M. Zorkany and B. Hesham, “Online Bus Arrival Time Prediction Using Hybrid NeuralNetwork and Kalman filter Techniques”, in *International Journal of Modern Engineering Research*, vol. 3, no. 4, pp. 2035-2041, Jul-Aug. 2013.
- [6] Z. He, H. Yu, Y. Du and J. Wang, “SVM based Multi-index Evaluation for Bus Arrival Time Prediction”, in *International Conference on ICT Convergence (ICTC)*, pp. 86-90, Jeju, South Korea, 14-16 Oct. 2013.

- [7] B. Yu, T. Ye, X. Tian and G. Ning, "Bus Travel-Time Prediction with a Forgetting Factor", in *Journal of Computing in Civil Engineering*, vol. 28, no. 3, May 2014.
- [8] A. Khamparia and B. Pandey, "Knowledge and Intelligent computing methods in e-learning", in *International Journal of Technology Enhanced Learning*, vol. 7, no. 3, pp. 221-242, 2015.
- [9] S. Maiti, A. Pal, A. Pal, T. Chattopadhyay and A. Mukherjee, "Historical Data based Real Time Prediction of Vehicle Arrival Time", in *IEEE 17th International Conference on Intelligent Transportation systems (ITSC)*, Qingdao, China, October 8-11, 2014.
- [10] P. Li, D. Wang, C. Wang and D. Liu, "Identification of Traffic Accident Prone Location Based on Vehicle Speed Variability Indices", in *Tenth International Conference of Chinese Transportation Professionals (ICCTP)*, pp. 466-477, Beijing, China, Aug. 2010.
- [11] M. Sinn, J. W. Yoon, F. Calabrese and E. Bouillet, "Predicting arrival times of buses using real-time GPS measurements", in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 1227-1232, Anchorage, AK, 2012.
- [12] A. Shalaby and A. Farhan, "Prediction Model of Bus Arrival and Departure Times Using AVL and APC Data", in *Journal of Public Transportation*, vol. 7, no. 1, pp. 41-61, 2004.
- [13] R. P. S. Padmanaban, L. Vanajakshi and S. C. Subramanian, "Estimation of bus travel time incorporating dwell time for APTS applications", in *2009 IEEE Intelligent Vehicles Symposium*, pp. 955-959, Xi'an, 2009.
- [14] J. Zhang, L. Yan, Y. Han and J. Zhang, "Study on the Prediction Model of Bus Arrival Time", in *2009 International Conference on Management and Service Science*, pp. 1-3, Wuhan, 2009,
- [15] M. Zhang, F. Xiao and D. Chen, "Bus Arrival Time Prediction Based on GPS Data", in *Fourth International Conference on Transportation Engineering*, pp. 1470-1475, Chengdu, China, Oct. 19-20, 2013.

- [16] The Python Software Foundation. Python 3.1. <https://www.python.org/download/releases/3.0/>, 2021. Python 3.1 is a continuation of the work started by Python 3.0.
- [17] JSON Org. <https://www.json.org/json-en.html>, 2021. JSON (JavaScript Object Notation) is a lightweight data-interchange format.
- [18] Ozgur. Python Firebase. <http://ozgur.github.io/python-firebase/>, 2021. Python interface to the Firebase's REST API
- [19] Firebase Realtime Database. <https://firebase.google.com/products/realtime-database/>, 2021. Firebase es una base de datos para almacenamiento y sincronización de datos en tiempo real.
- [20] JavaScript (JS). <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, 2021. JavaScript es un lenguaje de programación ligero, interpretado o compilado justo a tiempo con funciones de primera clase.
- [21] Google Maps Platform. <https://mapsplatform.google.com/maps-products/#maps>, 2021. Google Maps ayuda a los usuarios a explorar el mundo con productos de mapas personalizados y detallados.
- [22] Instalación y configuración en JavaScript. <https://firebase.google.com/docs/database/web/start?hl=es>, 2021. Instalación y configuración de la base de datos de tiempo real de Firebase en JavaScript.