



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo Terminal

Prototipo de Sistema Informativo de Rutas en Transporte

Público (Caso Metro DF)

2014-A020

Presenta:

Cielo Torres Carlos Gustavo

Director:

Chávez Lima Eduardo



ESCOM

Junio 2015



**ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA**



No.TT 2014-A020

30 de junio 2015

Documento Técnico

**“Prototipo de Sistema Informativo para Rutas de Transporte
Público (Caso Metro)”**

Presenta:

Cielo Torres Carlos Gustavo

Director:

Chávez Lima Eduardo

RESUMEN

La ciudad de México, siendo la cuarta aglomeración urbana más grande del mundo [1], cuenta con diversos medios de transporte; sin embargo, muchas veces estos son insuficientes contra el flujo de gente que día a día viaja por la ciudad. La aglomeración de pasajeros, accidentes, el clima y otros factores traen como consecuencia un servicio lento y desagradable para el público. Este documento presenta la propuesta para el desarrollo de un prototipo de una aplicación móvil que le indique al usuario qué ruta dentro del Sistema de Transporte Colectivo de la Ciudad de México le conviene tomar, mediante el procesamiento de información, tal como flujo de pasajeros y ruta más corta. Esto será posible gracias a las tecnologías implementadas en los dispositivos móviles, los cuales pueden dar información acertada y son de acceso fácil al usuario.

Palabras clave: Comunicación inalámbrica, dispositivos móviles, algoritmos de enrutamiento, búsqueda heurística, ingeniería de transporte, aplicación cliente-servidor

e-mail: o0ox_xo0o@live.com



ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE FORMACIÓN INTEGRAL E
INSTITUCIONAL



COMISIÓN ACADÉMICA DE TRABAJO TERMINAL

México, D.F. a 6 de julio de 2015

DR. FLAVIO ARTURO SÁNCHEZ GARFÍAS
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
PRESENTE

Por medio del presente, se informa que el alumno que integra el TRABAJO TERMINAL 2014-A020, titulado "Prototipo de Sistema Informativo para Rutas de Transporte (Metro DF)" concluyeron satisfactoriamente su trabajo.

Los discos (DVDs) fueron revisados ampliamente por su servidor y corregidos, cubriendo el alcance y el objetivo planteados en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE

PROF. EDUARDO CHÁVEZ LIMA

ADVERTENCIA

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000

Contenido:

| | |
|---------------------------------------|------|
| Capítulo..... | Pág. |
| Resumen..... | V |
| Lista de figuras..... | XII |
| Lista de tablas..... | XIII |
| Lista de Símbolos y abreviaturas..... | XIV |
| Introducción | 1 |
| 1. Capítulo 1..... | 5 |
| 2. Capítulo 2..... | 21 |
| 3. Capítulo 3..... | 41 |
| 4. Capítulo 4..... | 47 |
| 5. Capítulo 5..... | 72 |
| Conclusiones | 91 |
| Trabajo a futuro..... | 92 |
| Referencias..... | 93 |

Lista de figuras

| Ref. Figura..... | Pág. |
|--|------|
| Figura I-1: Logotipo Waze 2014 | 1 |
| Figura I-2: Grafo representativo de un problema resuelto con A* | 3 |
| Figura 1-1: Grafo ejemplificando método de costos mínimos. | 6 |
| Figura 1-2: Convoy modelo NM-79 del STC DF..... | 8 |
| Figura 1-3: Mapa de rutas alternas a apoyo de línea 12. | 10 |
| Figura 1-4: Mapa Metrobús Ciudad de México | 15 |
| Figura 1-5: Mapa del tren Ligero de la ciudad de México..... | 16 |
| Figura 1-6: Mapa del tren Suburbano México | 17 |
| Figura 1-8: Mapa SITEUR | 18 |
| Figura 2-1: Tiempo en segundos que tardan en realizarse $f(n)$ operaciones, haciendo un millón por segundo | 22 |
| Figura 2-2: Grafo de un modelo de carreteras | 23 |
| Figura 2-3: Ejemplo de grafo con matriz de adyacencia | 24 |
| Figura 2-4: Tipo de grafo en el que se puede aplicar el algoritmo Dijkstra | 26 |
| Figura 2-5: Ejemplo algoritmo Prim | 27 |
| Figura 2-6: Grafo de un modelo ferroviario. | 28 |
| Figura 2-7: Diagrama de flujo del algoritmo de enrutamiento a utilizar en el proyecto | 29 |
| Figura 2-8: Ejemplo de back tracking en búsqueda a lo ancho. | 30 |
| Figura 2-9: Grafo a resolver por método propuesto | 31 |
| Figura 2-10: Árbol inverso generado por algoritmo | 32 |
| Figura 2-11: Mapa de la red del STC | 35 |
| Figura 2-12: Grafo construido de la red del STC | 36 |
| Figura 2-13: Fragmento del grafo construido del STC | 37 |
| Figura 2-14: Ejemplificación de nodo inicial y nodo destino dentro del grafo del STC | 38 |
| Figura 2-15: Árbol inverso generado por algoritmo | 39 |
| Figura 2-16: Árbol inverso mostrando las rutas encontradas | 39 |

| | | |
|---------------------|--|----|
| Figura 4-1: | Comparativa de uso de sistemas operativos de usuarios en smartphones . | 47 |
| Figura 4-2: | Ejemplo de mapa en Google Maps . | 48 |
| Figura 4-3: | Diagrama de subsistemas . | 55 |
| Figura 4-4: | Diagrama de casos de uso del subsistema cálculo de ruta más corta . | 56 |
| Figura 4-5: | Diagrama de casos de uso de subsistema comparación de afluencia predicha en rutas encontradas . | 57 |
| Figura 4-6: | Diagrama de casos de uso de subsistema gestión de reportes. . | 58 |
| Figura 4-7: | Diagrama de caos de uso de gestión de usuarios. . | 59 |
| Figura 4-8: | Diagrama de casos de uso de gestión de rutas populares . | 60 |
| Figura 4-9: | Diagrama UML del sistema . | 61 |
| Figura 4-10: | Arquitectura general cliente-servidor . | 62 |
| Figura 4-11: | Arquitectura del sistemat . | 63 |
| Figura 4-12: | Modelo relacional de la base de datos. . | 64 |
| Figura 5-1: | Pantalla de Login . | 73 |
| Figura 5-2: | Pantalla principal de sistema..... | 74 |
| Figura 5-3: | Consulta de información de estaciones..... | 75 |
| Figura 5-4: | Consulta de reportes sobre anomalías en estaciones..... | 77 |
| Figura 5-5: | Interacción con mapa fuera del flujo. . | 78 |
| Figura 5-6: | Selección de punto inicial. . | 79 |
| Figura 5-7: | Despliegue de id estación. . | 80 |
| Figura 5-8: | Despliegue de ruta optima. . | 82 |
| Figura 5-9: | Despliegue de información ruta optima y ruta alterna..... | 83 |
| Figura 5-10: | Despliegue de formulario de reportes. . | 87 |
| Figura 5-11: | Despliegue de reporte recién creado. . | 90 |

Lista de tablas

| Ref. Tabla | Pág. |
|---|-------------|
| Tabla 2-1: Extracción de datos de costo en cada ruta calculada por algoritmo | 32 |
| Tabla 2-2: Tiempo promedio de recorrido de estación a estación de la línea 2..... | 34 |
| Tabla 2-3: Comparación de tiempos de las rutas encontradas por algoritmo | 40 |
| Tabla 3-1: Afluencia de pasajeros en Jul-Sep 2014 de la línea 1 del STC Metro DF | 42 |
| Tabla 3-2: Entrenamiento de método de promedios móviles..... | 45 |
| Tabla 3-3: Predicción de afluencia de pasajeros en periodo siguiente de las rutas obtenidas por algoritmo de enrutamiento del capítulo 2..... | 46 |
| Tabla 4-1: Actores del sistema..... | 50 |
| Tabla 4-2: Entrenamiento de método de promedios móviles..... | 45 |
| Tabla 4-1: Objetivos del sistema..... | 50-52 |
| Tabla 4-3: Requisitos de almacenamiento de información | 52-55 |
| Tabla 4-4: Pantallas del sistema..... | 65-71 |

Lista de Símbolos y abreviaturas

| Abreviatura | Término |
|--------------------|----------------|
|--------------------|----------------|

| | |
|-----|---------------------------------------|
| STC | Sistema de Transporte Colectivo Metro |
|-----|---------------------------------------|

| | |
|-----------|------------------|
| <i>DF</i> | Distrito Federal |
|-----------|------------------|

Introducción

La ciudad de México, al ser una de las más grandes y pobladas del mundo, requiere un sistema de transportes eficiente, rápido y seguro, pero lamentablemente no cuenta con él. Sistemas de transporte tales como el metro, el metrobús, y el tren suburbano suelen ser día a día la elección de cientos de miles de habitantes de las zonas aledañas, y a pesar de ser opciones baratas y factibles, suelen ser saturadas a ciertas horas, lo que provoca que los usuarios lleguen tarde a su destino.

El proyecto propuesto es una aplicación móvil que ayude a planificar recorridos en el Sistema de Transporte Colectivo Metro, con base en datos como la ruta más corta, las horas pico, e inclusive, recibir información en tiempo real, como cierre de estaciones, fallos de vías, justo como la aplicación Waze, con la diferencia que Waze sólo aplica para usuarios con automóvil [2]. Recibir la información en tiempo real es posible gracias a que la aplicación tendrá la opción de conectarse a internet y el usuario podrá reportar alguna anomalía en alguna estación, si el servidor detecta varios clientes que reportan el mismo problema, dígase un suicidio o problemas en la vías, aportará esa información a los demás clientes que se conecten, previniéndolos de tomar esa ruta, o al menos advertirles que pueden tardarse un poco más en su recorrido.



Fig. 1-1 Logotipo Waze 2014

Es verdad que hoy día existen más aplicaciones con propósito similar, pero no ofrecen el enfoque matemático que nuestro proyecto implementa, ya que se aplicarán algoritmos de enrutamiento y grafos para calcular la distancia más corta, dándole al usuario opciones más claras para tomar una decisión para tomar una ruta en específico. Hay principalmente 2 aplicaciones con fines similares: la primera es “Metro de México” la cual implementa Google Maps como apoyo, pero no proporciona información clara sobre la ruta más conveniente, sólo despliega el mapa para que

el usuario decida con base en lo que ve en él [3].. La segunda de igual forma utiliza Google Maps como apoyo y es un poco más clara, ya que traza la mejor ruta, pero sus cálculos de la ruta más corta son imprecisos y ambiguos sin tomar en cuenta que deja a un lado varios factores externos como las horas pico, sin mencionar que no indica el tiempo de la ruta trazada [4].

El objetivo general es desarrollar un sistema prototipo que permita a través de un dispositivo móvil, emitir información útil sobre qué ruta dentro del Sistema de Transporte Colectivo de la Ciudad de México le conviene tomar, basada en varios parámetros como la ruta más corta, el horario, flujo de gente, e inclusive algunos factores en tiempo real como avería de un tren, estaciones o algún desafortunado suicidio. Como se ha mencionado, esto es posible gracias a que el usuario podrá aportar ésta información y prevenir a los demás usuarios sobre algún evento que puede retrasar su viaje; es posible que el usuario que reporte el atraso no sepa la razón por la cual el metro se tarda, pero eso vendrá indicado en la información que reciba el servidor, la cual le indicará a los demás clientes que se han reportado retrasos por varios usuarios por razones desconocidas, los demás usuarios tendrán la decisión final. En general, el desarrollo se dividirá en los siguientes módulos:

1. Desarrollo del Módulo Cliente que solicitará las peticiones o información: La aplicación tendrá la posibilidad de conectarse al servidor, el cual proporcionará los datos en tiempo real, sin mencionar que mostrará el mapa del metro con ayuda de Google Maps, y con eso tener una percepción más realista de la distancia que se va a recorrer. Sin embargo, eso no indica que la aplicación no será funcional sin acceso a internet, ya que el módulo del cliente ya tendrá precargada información necesaria como la distancia entre estación y estación para así hacer el cálculo con el algoritmo de enrutamiento que será empleado. Se debe aclarar que, la información que proporcione el servidor sólo prevendrá a los usuarios sobre acontecimientos externos que pueden retrasarlos, pero no tendrán lugar como variables dentro del algoritmo de enrutamiento, ya que es muy difícil hacer cálculos exactos con ésta clase de información, el usuario sólo debe considerarlo como información “extra”.
2. Desarrollo del Módulo Servidor que procesará la información y mandará al cliente el resultado de ese proceso. El servidor recibirá los reportes que envíen los usuarios, y al recibir e-mail: o0ox_xo0o@live.com

varios reportes similares de la misma estación, alertará a los demás clientes que se conecten un evento puede estar retrasando los trenes en cierta zona. Se debe tomar en cuenta que la red de datos celulares tiende a fallar en el metro, y es posible que no en todas las estaciones se pueda enviar un reporte, así que como se mencionó en el punto anterior, ésta información debe considerarse como un extra, pero que de alguna forma, pueda mejorar el recorrido.

3. Implementación de algoritmos de enrutamiento dentro del ámbito de transporte. El algoritmo de enrutamiento a utilizar está basado en A*, el cual es una búsqueda heurística y es implementado en videojuegos.

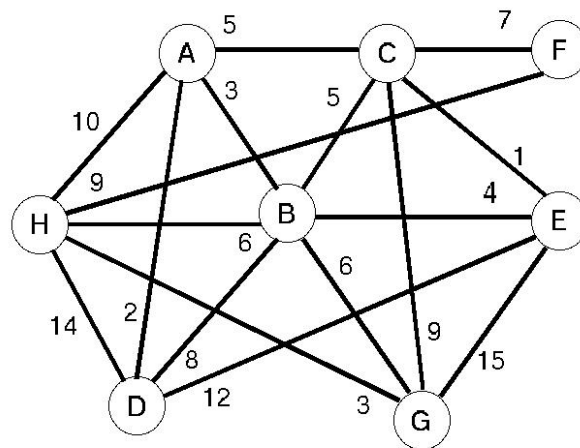


Fig. I-2 Grafo representativo de un problema resuelto con A*

El sistema de transporte de la Ciudad de México es un sistema bastante caótico, sin embargo es necesario para la mayor parte de la población; por éste motivo se propone el desarrollo de éste proyecto para facilitar la movilidad de los usuarios.

Tomando en cuenta también que hoy en día la capacidad de adquirir un teléfono móvil con transferencia de datos es muy común, de este modo, una mayor cantidad de usuarios podrán tener acceso a la información que la aplicación proporcionará.

El desarrollo del proyecto integra conocimientos relacionados con el área de Sistemas computacionales, inteligencia artificial e ingeniería de transporte.

En cuanto a la metodología, debido a tendencias actuales, la metodología híbrida del desarrollo de software en México es predominante, ya que la gran mayoría de las empresas desarrolladoras de software son jóvenes, y tienen proyectos que pueden irse lanzando y mejorando con la ayuda de la opinión del usuario; en nuestro caso, ya que la aplicación será dirigida al público en general (que cuente con un dispositivo móvil y frecuente el transporte colectivo) podemos lanzar diferentes actualizaciones, implementando mejoras a nuestro trabajo, pero sin olvidar los modelos tradicionales que hemos estudiado a lo largo de la carrera[5].

Dentro del sistema se otorgara mayor peso a la metodología de desarrollo evolutivo, ya que se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado. La ventaja de un proceso del software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar en forma creciente. [6]

En general, la aplicación funcionará en base a cliente-servidor, siendo la aplicación en Android el cliente, la cual se encargará de realizar el cálculo de la ruta más corta en base a distancia y tiempo promedio; en dado caso que el usuario esté conectado a Internet, se enviará una petición al servidor, el cual enviará información de parámetros externos como afluencia de gente e incidentes que los mismos usuarios notifiquen.

Capítulo 1: Marco General

Antes de centrarnos en el análisis, diseño e implementación del sistema que proponemos, es importante saber algunos antecedentes que debemos tomar en cuenta, ya que al estudiarlos podemos ver el alcance que tiene nuestro proyecto. En este panorama general haremos un breve estudio de temas relacionados con el STC y de algunas disciplinas relacionadas.

Problema de transporte y distribución

La Ingeniería de Transporte se ocupa de “la aplicación de los principios tecnológicos y científicos a la planificación, el proyecto funcional, la operación y la administración de las diversas partes de cualquier modo de transporte, con el fin de proporcionar a las personas y mercancías una movilidad segura, rápida, confortable, conveniente, económica y compatible con el medio ambiente” [8]. El ingeniero de transporte se ocupa de la planificación, el diseño, la construcción y la administración de un sistema de transporte, mediante el uso de técnicas de Ingeniería Civil, Economía, Investigación de Operaciones, etc.

Existen varios grupos que se dedican a la ingeniería de transporte, algunos de ellos estudian problemas de tráfico, logística y transporte de carga, transporte urbano y sistemas de información para el transporte. Otros grupos estudian pavimentos, diseño estructural, diseño geométrico, construcción, conservación y operación de obras. Además hay otros investigadores que estudian otros temas del transporte, o bien aspectos relacionados con el transporte, como vehículos y medio ambiente. [7]

En este documento abordamos una problemática que se estudia ya hace varios años dentro del campo de ingeniería de transporte; estos problemas ya han sido abordados desde enfoques diferentes, los ingenieros industriales por ejemplo, atacan esta problemática con métodos propuestos por la investigación de operaciones.

El problema del transporte o distribución es un problema de redes especial en programación lineal que se funda en la necesidad de llevar unidades de un punto específico llamado Fuente u Origen hacia otro punto específico llamado Destino. Los principales objetivos

de un modelo de transporte son la satisfacción de todos los requerimientos establecidos por los destinos y claro está la minimización de los costos relacionados con el plan determinado por las rutas escogidas.

El contexto en el que se aplica el modelo de transporte es amplio y puede generar soluciones atinentes al área de operaciones, inventario y asignación de elementos.

El procedimiento de resolución de un modelo de transporte se puede llevar a cabo mediante programación lineal común, sin embargo su estructura permite la creación de múltiples alternativas de solución tales como la estructura de asignación o los métodos heurísticos más populares como Vogel, Esquina Noroeste o Mínimos Costos.[9]

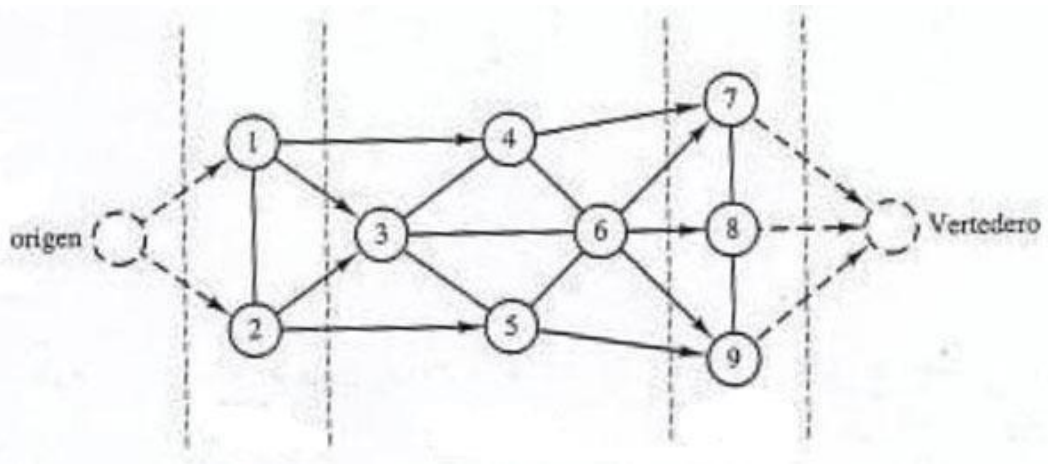


Fig. 1-1 Grafo ejemplificando método de costos mínimos

Estos problemas son resueltos comúnmente con el método simplex, una vez que se ha desarrollado el grafo, y han demostrado ser bastante efectivos para este tipo de problemas., sabiendo ya las rutas que debemos tomar, cuanto nos costará (tiempo, dinero etc.) y cómo influiría alguna anomalía en alguna de las rutas.

Como podemos ver, los métodos que se utilizan para resolver los problemas de transporte están muy relacionados con teoría de grafos y no son tan diferentes a los que nosotros proponemos, entonces podemos concluir que nuestra propuesta no es descabellada.

Breve historia del STC Metro DF

Para entender un poco sobre la importancia de resolver los problemas de transportes, debemos tomar como antecedente algo de la historia del metro, ya que es red a la que se dirige el estudio de este documento. La historia del STC Metro va ligada a la de la Ciudad de México estrechamente debido a importantes y trascendentales transformaciones realizadas en ella para su construcción, durante el gobierno del presidente Adolfo López Mateos (1958-1964) ya se tenía un anteproyecto de realizar un Metro en la Ciudad de México con el fin de resolver los problemas de vialidad que comenzaban a presentarse durante esa época. Sin embargo, el Gobierno del Distrito Federal decidió abandonar este proyecto para centrarse en la construcción de otra gran obra: el Anillo Periférico.

Para 1967 la Ciudad de México ya tenía graves problemas viales, principalmente en el Centro Histórico donde circulaban la mayoría de los automóviles. Es por eso que el G.D.F. decidió crear el 25 de abril el Sistema de Transporte Colectivo (STC) Metro planeando así construir un tren subterráneo en la ciudad. En el momento de su construcción, el país no contaba con los recursos suficientes para llevar a cabo un proyecto tan grande, es por eso que pidió un préstamo al gobierno francés por 856.5 y 142.5 millones de francos. Entonces se inició la construcción del Metro a cargo de la empresa de Ingenieros Civiles y Asociados (ICA), la cual firmó un contrato con el STC Metro en junio de 1967.

Para definir el recorrido de la primera línea del Metro, fueron tomadas en cuenta las áreas de mayor concentración de autos y camiones. La misma fue inaugurada el 4 de septiembre de 1969 y recorrería desde la estación “Zaragoza” a la estación “Chapultepec”, para así cubrir la demanda de pasajeros y disminuir el tráfico vial.

Para la década de los setentas, el Metro ya había cobrado gran importancia y afluencia en los pobladores de la ciudad, entonces se decidió expandir la red con la construcción de las líneas 2 y 3, inauguradas respectivamente el 1 de agosto y 20 de noviembre de 1970 por el Presidente Gustavo Díaz Ordaz, estas irían de “Tacuba” a “Taxqueña”; y de “Tlatelolco” a “Hospital General”.

Con 3 líneas, la Ciudad de México contaba con una red de transporte subterránea con una extensión de 42.2 Km., y 45 estaciones ubicadas en los puntos más concurridos de la ciudad en ese entonces, trasportando cerca de 600 millones de personas al año, ahorrando tiempo en su traslado y por sólo un peso el boleto.



Fig. 1-2 Convoy modelo NM-79 del STC DF

En la década de los 80, la Comisión de Vialidad y Transporte Urbano elaboró el proyecto para la construcción de las líneas 4 y 5. La línea 4 inaugurada el 29 de agosto de 1981 presentó muchos cambios en la concepción del Metro ya que ahora se introducían las vías y estaciones elevadas. La línea 5 inaugurada en agosto de 1982 en el norte de la ciudad, cubría el tramo de “Politécnico” a “La Raza”, dando una nueva opción a los estudiantes del IPN de llegar a Zacatenco.[12]

No obstante los estudiantes universitarios también gozaron de los servicios del Metro con la ampliación de la línea 3 en agosto de 1983, la cual iría de “Indios Verdes” hasta “Universidad”. La línea 1, después de 15 años de su inauguración se expande hasta “Tacubaya” y “Pantitlán”, en el norte la línea 2 llega hasta la estación de “Cuatro Caminos” facilitando a los pobladores de la zona Metropolitana para transportarse al Distrito Federal. En esta década la creciente demanda de usuarios en la zona Norte y Oriente, obliga la creación de las líneas 6 y 7 cuyas terminales fueron “El instituto del Petróleo” a “Martín Carrera”, y de “Tacuba” a “El Rosario” respectivamente.

Fueron inauguradas el 21 de diciembre de 1983 y el 29 de noviembre de 1988. Para estas fechas el Metro ya contaba con 114.7 Km. Y 105 estaciones, transportando a 1 542 millones de usuarios al año.

A mediados de los años noventa, el Metro se convirtió en un factor importante en el transporte de los habitantes de la Ciudad de México así como de los del área Metropolitana, que también necesitaban un transporte que los llevara a sus lugares de trabajo, ubicados mayormente, en el D.F. Es por eso que se decidió crear la línea A que recorre de “Pantitlán” a “La Paz”, la cual 3 años después ya transportaba al 6% total de los usuarios del Metro. Poco después en 1994 se inauguró la línea 8, de “Garibaldi” a “Constitución de 1917”, la cual, era la tercera línea con mayor exención de recorrido.

Por último, en el año 2000 se inaugura la última línea construida hasta el momento por el STC Metro: la línea B, que va desde “Buenavista” hasta “Ciudad Azteca”, que conecta el Centro Histórico con municipios del Estado de México (tales como Nezahualcóyotl y Ecatepec).

En 2006 se inició como un nuevo proyecto la construcción de la línea 12 del Metro la cual conecta las zonas de Tláhuac y Mixcoac con una extensión de 24 Km. De recorrido. Esta misma fue abierta en 2013, pero por problemas en su tramo elevado, se tuvo que cerrar por tiempo indefinido, las fallas se atribuyen al diseño, según expertos. [10]



Fig. 1-3 Mapa de rutas alternas a apoyo de línea 12

El Metro de la ciudad de México se ha convertido en el medio de transporte más usado por millones de capitalinos, quienes encuentran en él: comodidad, rapidez y eficiencia. Por esa razón es que se ha decidido desarrollar el sistema propuesto sobre esta red, la más importante de nuestro país.

Factores externos que influyen sobre los viajes en el STC

En esta investigación analizaremos factores cuantificables que se relacionan directamente con el tiempo de viaje de un usuario común del metro, tales como la distancia y la afluencia de pasajeros. Pero existen también factores externos que modifican el tiempo total en el trayecto, que a diferencia de los ya mencionados, no se pueden cuantificar; en este documento los llamaremos “factores secundarios” los dividimos en dos tipos:

Factores predecibles: Como se mencionó anteriormente, los factores secundarios no son cuantificables, pero eso no quiere decir que algunos de ellos no puedan ser predichos; podemos mencionar dos de ellos:

- Clima lluvioso
- Horas pico entre semana

En cuanto al factor de las horas pico, según la página oficial del metro [12] (y a experiencia propia) tenemos 3 periodos en los que hay mayor cantidad de pasajeros utilizando el servicio:

- De 7:00 am a 8:30 am
- De 2:00 pm a 3:00 pm
- De 6:00 pm a 8:00 pm

Por supuesto, que en algunas estaciones la hora pico puede ser diferente, pero estamos contemplando un panorama general. En el caso del clima lluvioso, existen servicios que nos indican el clima dependiendo de nuestra ubicación; en este proyecto no se implementarán dichos servicios, pero es importante tomarlo en cuenta para trabajo a futuro

Factores impredecibles: Así como existen factores secundarios que pueden predecirse, existen muchos otros que no, a continuación mencionamos algunos de ellos:

- Suicidios
- Fallos en los trenes o vías
- Manifestaciones
- Conductores mal capacitados
- Activación de alarma de emergencia

Para atacar esta parte del problema, los usuarios podrán dar reportes sobre estas situaciones y subirlas al servidor por medio de la aplicación móvil a través de internet. El sistema contará con un medidor de confianza: si varios usuarios reportan la misma situación, el reporte será etiquetado como confiable, de lo contrario, su nivel será bajo. Los incidentes pueden ocurrir en cualquier momento, y no hay una forma de sacar un patrón entre ellos.

Ejemplos populares de incidentes ocurridos en el STC Metro DF

No sólo debemos analizar la parte de ingeniería de transporte del metro, sino que, como se mencionó en la parte inicial de este documento, también hay que analizar qué clase de incidencias o anomalías han ocurrido en este sistema de transporte tan grande y veterano, pues como sabemos, estas influyen directamente en la duración del recorrido que tomamos.

Con tantos años de existencia, han ocurrido diferentes eventos que van fuera de lo contemplado y que no son predecibles ni cuantificables. Algunas historias dignas de recordar, aunque otras se quisieran borrar de la memoria.

A continuación un recuento de sucesos que a diario se presentan en vagones y andenes del Metro y otros que marcarán de por vida la seguridad de este transporte:

El choque de convoyes, 20 de octubre de 1975: Este fue el percance más grave en la historia del Metro de la ciudad de México. El impacto de dos trenes tuvo como saldo la muerte de 39 personas, 119 heridos y la detención del conductor Carlos Fernández, quien por “su imprudencia” fue sentenciado a doce años de prisión:

“El accidente se debió a una imprudencia y descuido del Carlos Fernández Sánchez, al conducir el convoy número 10 de la estación Chabacano a la de Viaducto, en la que embistió al tren número 08”, declaró en su momento Octavio Sentíos, regente de Distrito Federal.

Aunque la culpa se atribuyó al chofer, diversos empleados aseguraron que todo fue un boicot orquestado por los titulares del entonces Departamento del Distrito Federal (DDF) y del STC Metro para demostrar que era necesario instalar el sistema de pilotaje automático:

“Fue un sabotaje. Su idea no era sacrificar inocentes, era nomás un alcance, un enganche de trenes, pero les falló”, declaró Héctor Manuel Zavala Bucio, integrante del sindicato de la empresa.

Balacera en Metro Balderas, 18 de septiembre del 2009: Dos hombres, uno de ellos aparentemente trastornado, desencadenaron una balacera en el interior del andén de la línea tres del Metro Balderas. El saldo fue de un policía y un pasajero muertos, además de cinco personas heridas:

“Todo se desató a las 17 horas con 14 minutos. El policía de la Bancaria e Industrial, Víctor Manuel Miranda, le gritó: “¡Tranquilo, qué te pasa!” a Luis Felipe Hernández, quien pretendía realizar pintas en los muros del Metro Balderas. El uniformado intentó someter al agresor, pero éste sacó un revólver .38 y tras un breve forcejeo, lo mató”, publicó EL UNIVERSAL.

Caídas a vías sin consecuencias fatales, 24 y 29 de octubre del 2009: Los noticieros de Televisión Azteca dieron a conocer un par de accidentes de similares características que se presentaron en las vías del Metro de la ciudad de México. En ellos se aprecia que la rápida reacción del conductor y el aviso de los usuarios fueron fundamentales para que se salvara la vida de dos personas, quienes cayeron por accidente a los rieles.

Hinchas del América arrojan a policía a las vías, 7 de febrero del 2010: Los hechos ocurrieron al filo de las 20:00 horas en la estación del Metro Hidalgo, cuando la porra denominada 'Disturbio' abordó el Metro después de haber presenciado el encuentro entre los equipos América e Indios.

Al llegar al andén de la estación, se suscitó un conato de enfrentamiento con policías capitalinos que les pedían que mantuvieran el orden. Fue en ese momento cuando un grupo de aproximadamente 60 aficionados acorraló a un policía y lo arrojó a las vías, segundos antes de que arribara un convoy el cual tuvo que detener su marcha.

Tras el percance se logró el aseguramiento de Juan Pablo Chombo Ramírez, quien fue señalado como el culpable de arrojar al oficial a las vías.

Conductor ebrio abre puertas del lado equivocado, 26 de septiembre del 2010: El chofer Francisco de la Cruz Mijangos fue detenido por conducir en estado de ebriedad, luego que desde 2007 se desempeñara como Gerente Inspector de Vías en el Metro, cargo por el que recibió adiestramiento especializado.

Los pasajeros del convoy se percataron que algo marchaba mal cuando en la estación Aculco de la línea 8, el conductor abrió las puertas de los vagones, pero no las que daban al andén, sino las que estaban dirigidas a las vías. Nadie resultó herido.

Actividades extremas en las instalaciones del Metro: A diario miles de jóvenes, amantes de la adrenalina, utilizan las instalaciones del Metro para demostrar sus habilidades en actividades extremas y peligrosas.[11]

*No se incluye ninguna ilustración debido a la crudeza de algunas de las imágenes recopiladas por los periódicos de los eventos anteriormente mencionados. Al mencionarlos no se pretende fomentar el morbo en el lector.

Al leer sobre estos ejemplos, nos damos cuenta de la importancia que tiene el reportar incidentes de esta naturaleza, ya que podemos prevenir que muchos usuarios no crucen con situaciones desagradables que puedan marcarlos de por vida.

Otras redes de transporte eléctrico en donde se podría implementar sistema planificador

La red del STC Metro no es la única red de transporte que existe en México, existen también otras redes en las cuales se podría implementar el sistema que se desarrollará en base a este documento, ya que estas son bastante similares entre sí. A continuación se mencionan algunas de estas redes de transporte con su respectivo mapa:

Metrobús: Es un sistema de transporte, basado en autobuses de capacidad y tecnología de punta, que brinda movilidad urbana de manera rápida y segura por medio de la integración de una infraestructura preferente, operaciones rápidas y frecuentes, sistema de pago automatizado y excelencia en calidad en el servicio.

Es un modo de transporte BRT (Bus Rapid Transit) que combina estaciones, vehículos, servicios y alta tecnología en un sistema integral con una identidad positiva. Tiene diversos componentes distintivos que juntos conforman un sistema integral. [13]

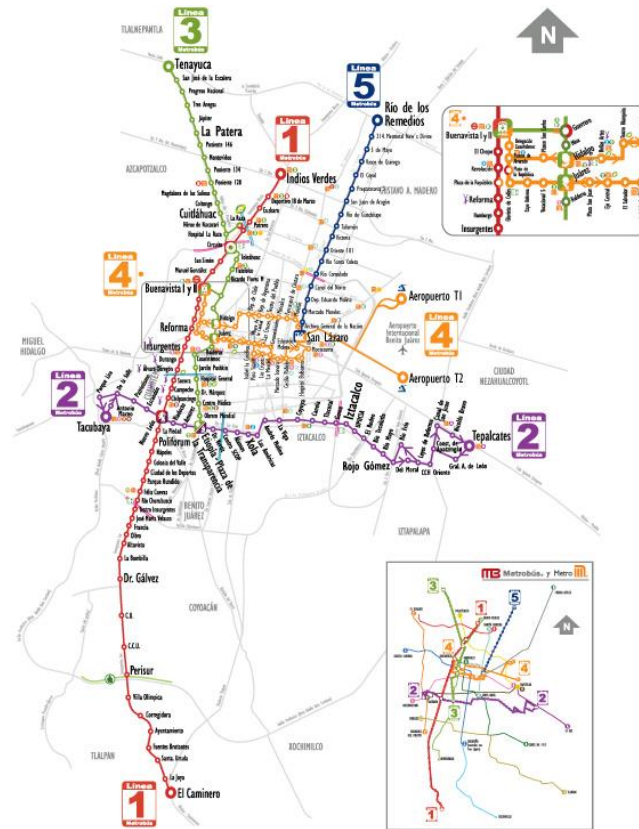


Fig. 1-4 Mapa Metrobús Ciudad de México

Tren Ligero: El Tren Ligero forma parte de la red del Servicio de Transportes Eléctricos del Distrito Federal, el cual opera en el Sur de la Ciudad de México prestado un servicio de transporte no contaminante a la población de las Delegaciones Coyoacán, Tlalpan y Xochimilco, brindando su servicio a través de 16 estaciones y 2 terminales, mediante 20 trenes dobles acoplados con doble cabina de mando con capacidad máxima de 374 pasajeros por unidad. [14]

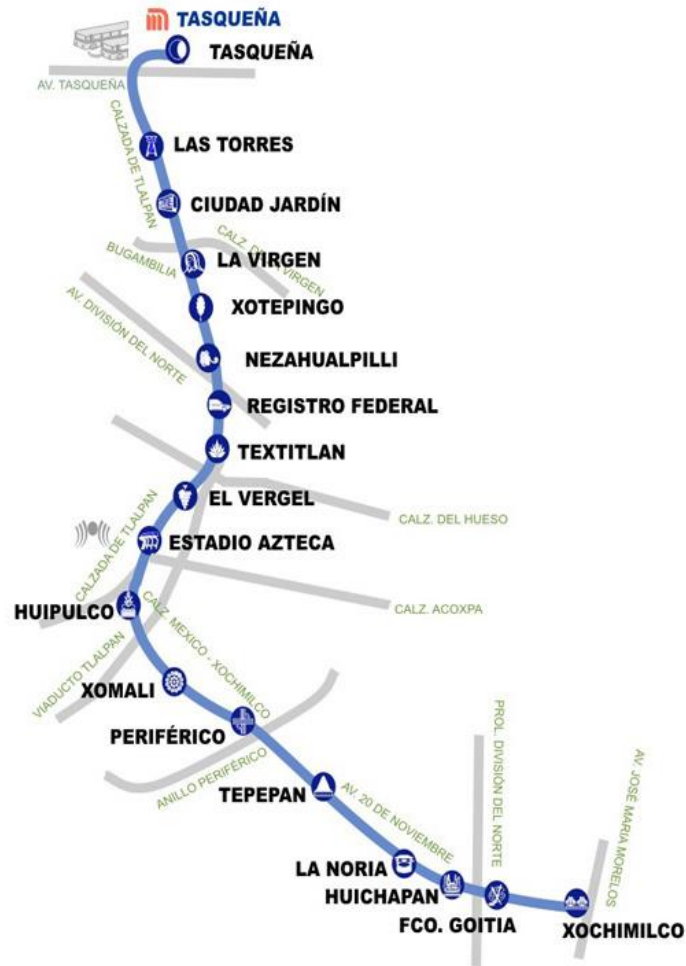


Fig. 1-5 Mapa del tren Ligero de la ciudad de México

Tren Suburbano: El Tren Suburbano es el medio de transporte entre el Estado de México y el Distrito Federal, es el primero en utilizar el transporte ferroviario masivo para pasajeros en la República Mexicana, y representa un esfuerzo conjunto encabezado por el Gobierno Federal, a través de la Secretaría de Comunicaciones y Transportes, junto con CAF México, y con la colaboración de los gobiernos del Distrito Federal y del Estado de México.[15]



Fig. 1-6 Mapa del tren Suburbano México

Metrorrey: El Sistema de Transporte Colectivo Metrorrey y coloquialmente como Metro, es un sistema de trenes ligeros eléctricos presente en la ciudad de Monterrey, en el estado de Nuevo León, México. Cruza de oriente a poniente la ciudad de Monterrey, así como parte del municipio de Guadalupe. De norte a sur une actualmente al municipio de Escobedo (donde se sitúa la estación terminal Sendero) con la Macro plaza (Donde se sitúa la estación terminal Zaragoza) en el municipio de Monterrey. El crecimiento del sistema fue detenido por poco más de 11 años, por lo que ha sido insuficiente para las necesidades de una ciudad en constante crecimiento. Fue hasta mediados del año 2005 que se inició la construcción de la ampliación para la línea 2. [16]



Fig. 1-7 Mapa del Metrorrey

SITEUR: Sistema de transporte eléctrico urbano Guadalajara es un sistema de trenes ligeros eléctricos que prestan servicio a los municipios de la zona metropolitana de Guadalajara con dos líneas. La Línea 1, que corre de norte a sur cuenta con 19 estaciones mientras que en el recorrido de la Línea 2, que corre de centro a este, hay 10 estaciones. [17]



Fig. 1-8 Mapa SITEUR

Como mencionamos anteriormente, si recopilamos la información necesaria acerca de las líneas anteriores, podemos construir un modelo que nos pueda obtener la ruta más corta, siendo algunas de ellas mucho más fácil de encontrar pues en muchas de las redes anteriores existen muy pocas líneas aún. También debemos considerar que existen proyectos de expansión en varias redes de transporte (incluyendo al STC Metro DF) asimismo como construcción de nuevas. Algunos de los proyectos que existen en este momento son:

- Construcción de tren transurbano México Toluca
- Construcción de Tren de alta velocidad México Querétaro
- Construcción de tren transpeninsular Yucatán-QuintanaRoo
- Ampliación de líneas del STC
- Ampliación de línea 1 y construcción de línea 3 del SITEUR
- Construcción de línea 3 del Metrorrey
- Construcción de línea 6 y 7 del Metrobús

Se decidió enfocar este proyecto sólo a la red del Metro de la ciudad de México debido a que es la más afluencia de pasajeros y líneas tiene, por lo tanto es la que mayores problemas presenta. Sin embargo, el sistema puede crecer e implementarse para muchas más redes de transporte.

Resumen de funcionamiento de sistema

En los posteriores capítulos se hará el análisis de dos factores cuantificables de los que depende el tiempo de trayecto: la distancia entre estaciones y la afluencia de pasajeros. Pero como mencionamos anteriormente, el sistema tendrá un módulo de creación de reportes los cuales podrán advertir a los usuarios sobre alguna anomalía en determinada estación, lo que le ayudaría a planificar mejor su viaje. Habrá varios casos de uso para la aplicación, los cuales serán explicados en posteriores capítulos, pero el principal se podría resumir de esta manera:

1. Cálculo de la ruta más corta y sus mejores alternativas (si existieran)
2. Sobre las rutas obtenidas, calcular la afluencia esperada de pasajeros en cada una de ellas
3. Verificar si en alguna de las rutas hay alguna anomalía reportada por otros usuarios, los reportes tendrán un medidor de confianza, la fecha y la hora.

Si bien la “ruta óptima” puede ser bastante difícil de obtener debido al número de factores que influyen en ella, la información que dará el sistema ayudará al usuario a tomar la decisión que él considere mejor para su trayecto.

Capítulo 2: Algoritmos de enrutamiento

El primer aspecto cuantitativo que se abordará en esta investigación es el parámetro de la distancia; para esto nos apoyaremos en el uso de algoritmos que determinen la ruta más corta en la que debe transitar un flujo de datos sobre una trayectoria representada por un grafo, pero en este caso, cambiaremos ese flujo de datos por el pasajero en sí. Antes que nada debemos

familiarizarnos con el concepto de órdenes de complejidad, ya que para implementar un algoritmo es deseable saber cuál será su tiempo de ejecución.

Se dice que $O(f(n))$ define un “orden de complejidad”, escogeremos como representante de este orden a la función $f(n)$ más sencilla del mismo. [18]

- $O(1)$: Complejidad constante. Cuando las instrucciones se ejecutan una vez.
- $O(\log n)$: Complejidad logarítmica. Esta suele aparecer en determinados algoritmos con iteración o recursión no estructural, ejemplo la búsqueda binaria.
- $O(n)$: Complejidad lineal. Es una complejidad buena y también muy usual. Aparece en la evaluación de bucles simples siempre que la complejidad de las instrucciones interiores sea constante.
- $O(n \log n)$: Complejidad cuasi-lineal. Se encuentra en algoritmos de tipo divide y vencerás como por ejemplo en el método de ordenación quicksort y se considera una buena complejidad. Si n se duplica, el tiempo de ejecución es ligeramente mayor del doble.
- $O(n^2)$: Complejidad cuadrática. Aparece en bucles o ciclos doblemente anidados. Si n se duplica, el tiempo de ejecución aumenta cuatro veces.
- $O(n^3)$: Complejidad cúbica. Suele darse en bucles con triple anidación. Si n se duplica, el tiempo de ejecución se multiplica por ocho. Para un valor grande de n empieza a crecer dramáticamente.
- $O(n^a)$: Complejidad polinómica ($a > 3$). Si ‘ a ’ crece, la complejidad del programa es bastante mala.
- $O(2^n)$: Complejidad exponencial. No suelen ser muy útiles en la práctica por el elevadísimo tiempo de ejecución. Se dan en subprogramas recursivos que contengan dos o más llamadas internas.

En la siguiente tabla se muestra un ejemplo del tiempo de ejecución de cada orden de complejidad haciendo variar el valor de n .

| n \ f(n) | 10 | 20 | 30 | 40 | 50 | 70 | 100 |
|---------------------|-----------|-----------|-------------------------|-------------------------|-------------------------|------------------------|-----------------------|
| n | 0.00001 s | 0.00002 s | 0.00003 s | 0.00004 s | 0.00005 s | 0.00007 s | 0.0001 s |
| n log(n) | 0.00003 s | 0.00008 s | 0.00014 s | 0.00021 s | 0.00028 s | 0.00049 s | 0.0006 s |
| n ² | 0.0001 s | 0.0004 s | 0.0009 s | 0.0016 s | 0.0025 s | 0.0049 s | 0.01 s |
| n ³ | 0.001 s | 0.008 s | 0.027 s | 0.064 s | 0.125 s | 0.343 s | 1 s |
| n ⁴ | 0.01 s | 0.16 s | 0.81 s | 2.56 s | 6.25 s | 24 s | 1.6 min |
| n ⁵ | 0.1 s | 3.19 s | 24.3 s | 1.7 s | 5.2 min | 28 min | 2.7 horas |
| n ^{Log(n)} | 0.002 s | 4.1 s | 17.7 s | 5 min 35 s | 1 h 4min | 2.3 días | 224 días |
| 2 ⁿ | 0.001 s | 1.04 s | 17 min | 12 días | 35.6 a. | 37 Ma | 2.6 MEU |
| 3 ⁿ | 0.059 s | 58 min | 6.5 a. | 385495 a. | 22735 Ma | 52000 Ma | 10 ¹⁸ MEU |
| n! | 3.6 s | 77054 a. | 564 MEU | 1.6 10 ¹⁸ ME | 6 10 ³⁴ | 2.4 10 ⁷⁰ M | 2 |
| n ⁿ | 2.7 horas | 219 EU | 4.2 10 ¹⁴ ME | 2.4 10 ²⁸ ME | 1.8 10 ⁶⁷ ME | 3 10 ¹¹¹ M | 2 10 ¹⁸² M |

s: segundos min: minutos h: horas a: años
 Ma: millones de años
 EU: edad del universo MEU: millones de veces la edad del universo

Fig. 2-1 Tiempo en segundos que tardan en realizarse $f(n)$ operaciones, haciendo un millón por segundo

Es importante tomar en cuenta esta información, ya que al implementar nuestro algoritmo el tiempo de ejecución debe ser razonable, sin mencionar que los dispositivos móviles tienen un poder de procesamiento moderado. [19]

Ejemplos de algoritmos de enrutamiento

Ahora que hemos visto las bases para medir la complejidad de un algoritmo, procedemos a elegir uno de ellos, claro está que debe ser un algoritmo buscador de rutas, tales como los algoritmos de enrutamiento. Los algoritmos de enrutamiento son utilizados comúnmente en computación, pero se pueden aplicar a diversos campos, en este caso lo aplicaremos a ingeniería de transporte. Algunos tipos de algoritmos de enrutamiento son:

- Enrutamiento por la ruta más corta.
- Inundación.
- Enrutamiento por vector de distancia.
- Enrutamiento por estado del enlace.

- Enrutamiento jerárquico.
- Enrutamiento por difusión.
- Enrutamiento por multidifusión.
- Enrutamiento para hosts móviles.
- Enrutamiento en redes ad hoc.
- Búsqueda en nodos de redes de igual a igual.

Un algoritmo de enrutamiento debe tener en cuenta cinco características generales: debe ser óptimo, sencillo, robusto, de rápida convergencia y flexible. [20] Cada protocolo define y sigue en forma estricta su algoritmo y su métrica para cálculos de rutas, sin embargo, el tipo de enrutamiento que mejor se adapta a nuestra problemática es el enrutamiento por la ruta más corta, ya que los demás están enfocados a las redes de computadoras.

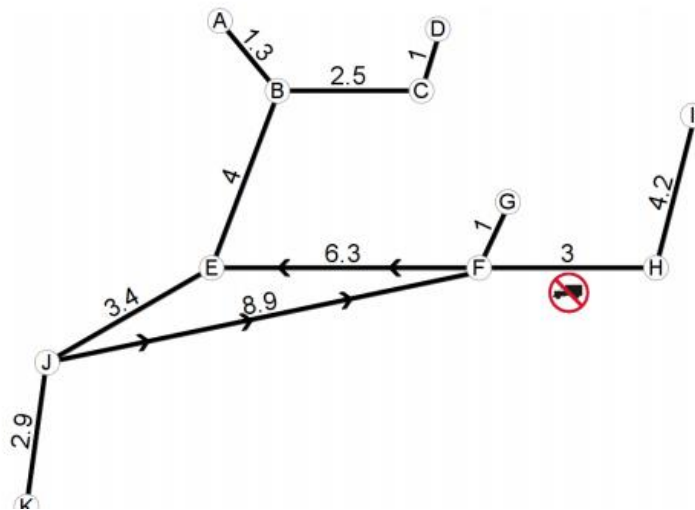
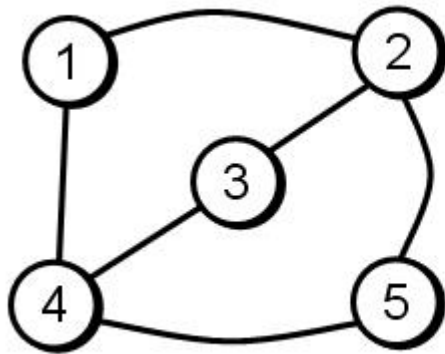


Fig.2-2 Grafo de un modelo de carreteras

Para hablar de este tipo de algoritmos, se debe tener claro qué es un grafo y cómo se representa en un ordenador. Un grafo en el ámbito de las ciencias de la computación es una estructura de datos, en concreto un tipo abstracto de datos (TDA), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos. El concepto de grafo TDA descende directamente del concepto matemático de grafo.

Informalmente se define como $G = (V, E)$, siendo los elementos de V los vértices, y los elementos de E las aristas (edges en inglés). Formalmente, un grafo G , se define como un par ordenado, $G = (V, E)$, donde V es un conjunto finito y E es un conjunto que consta de dos elementos de V .

Existen diferentes implementaciones del tipo grafo: con una matriz de adyacencias (forma acotada) y con listas y multilistas de adyacencia (no acotadas). Matriz de adyacencias: se asocia cada fila y cada columna a cada nodo del grafo, siendo los elementos de la matriz la relación entre los mismos, tomando los valores de 1 si existe la arista y 0 en caso contrario.



| M | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 0 |

Fig. 2-3 Ejemplo de grafo con matriz de adyacencia

Una vez dicho esto, existen algoritmos que se dedican a buscar el camino más corto en un espacio, entre dos puntos dado, la elección de un camino u otro conlleva un valor mayor o menor, es decir no es lo mismo ir del nodo 1 al nodo 3 con un valor de 5 unidades(metros), que ir del nodo 1 pasando por el 2 y llegando al 3 teniendo 4 unidades(metros), es preferible coger el que tenga menos coste, para que sea el valor más corto. Es decir que el problema consiste en encontrar un camino entre dos vértices (o nodos) de tal manera que la suma de los pesos de las aristas que lo constituyen es mínima. Un ejemplo es encontrar el camino más rápido para ir de una ciudad a otra en un mapa. En este caso, los vértices representan las ciudades, y las aristas las

e-mail: ouox_xouo@iive.com

carreteras que las unen, cuya ponderación viene dada por el tiempo que se emplea en atravesarlas. Para solucionar este problema, haremos mención de algunos de los algoritmos más utilizados:

Algoritmo Dijkstra: El algoritmo de Dijkstra consiste en ir explorando todos los caminos más cortos que parten del vértice origen y que llevan a todos los demás vértices; cuando se obtiene el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene. Su grado de complejidad es el siguiente:

$$O(|V|^2 + |E|) = O(|V|^2)$$

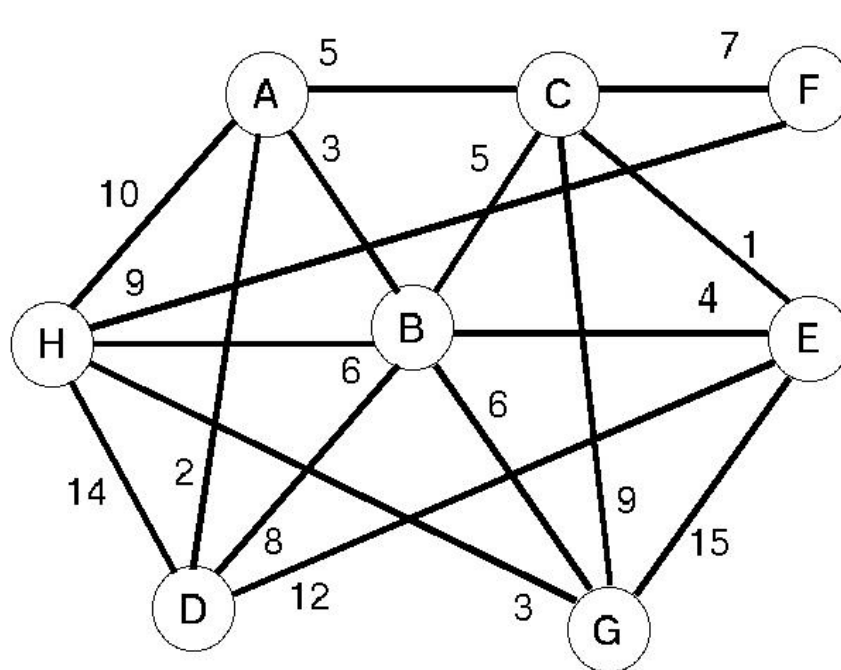


Fig. 2-4 Tipo de grafo en el que se puede aplicar el algoritmo Dijkstra

Algoritmo Prim: El algoritmo de Prim Es bastante parecido al algoritmo de Dijkstra; éste encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible. Si el grafo no es conexo, entonces el algoritmo encontrará el árbol mínimo para uno de los componentes conexos que forman dicho grafo no conexo; su grado de complejidad es de:

$O(E \log V)$

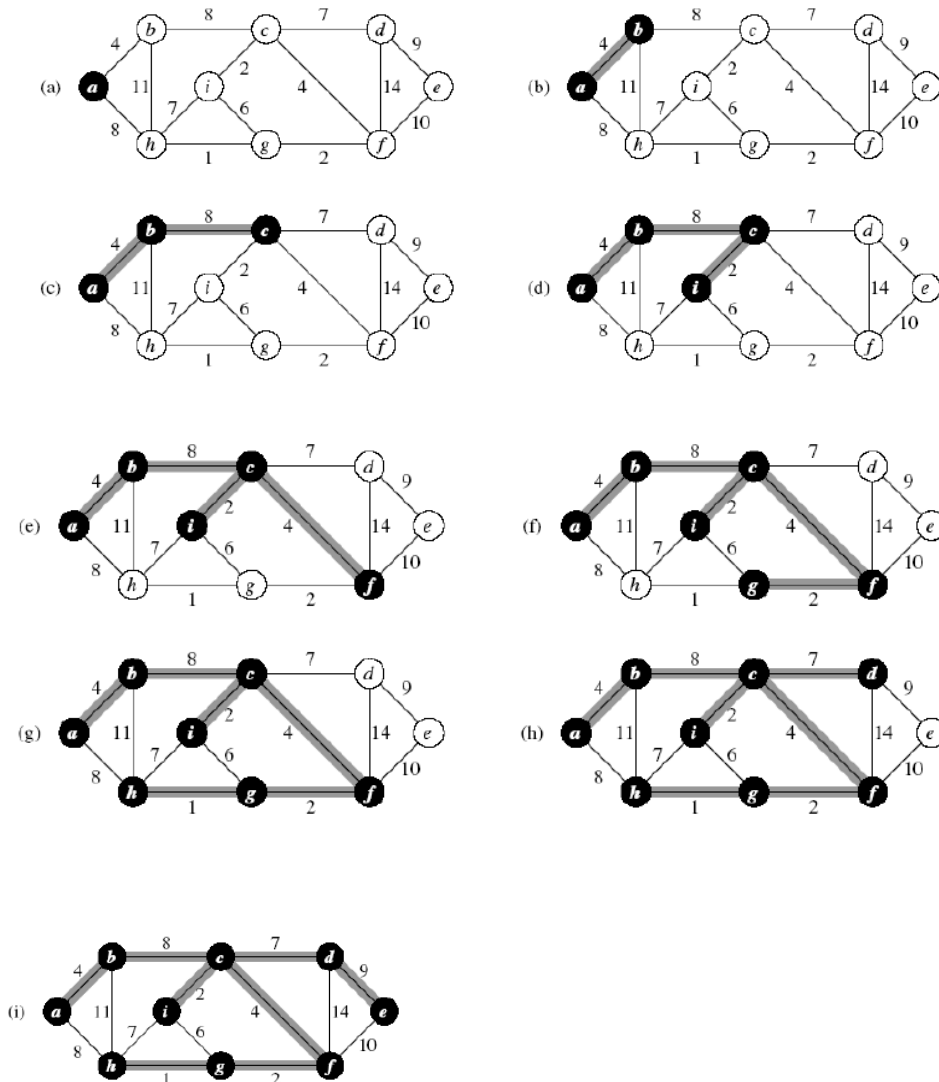


Fig. 2-5 Ejemplo algoritmo Prim

Algoritmo A*: El algoritmo A* evalúa los nodos combinando $g(n)$, el coste para alcanzar el nodo, y $h(n)$, el costo de ir al nodo objetivo:

$$F(n) = g(n) + h(n)$$

Ya que la $g(n)$ nos da el coste del camino desde el nodo inicio al nodo n , y la $h(n)$ el coste estimado del camino más barato desde n al objetivo, tenemos:

$F(n)$ = coste más barato estimado de la solución a través de n .

Así, si tratamos de encontrar la solución más barata, es razonable intentar primero el nodo con el valor más bajo de $g(n) + h(n)$. Resulta que esta estrategia es más que razonable: con tal de que la función heurística $h(n)$ satisfaga ciertas condiciones, la búsqueda A^* es tanto completa como óptima. [21]

El algoritmo A^* es utilizado en videojuegos, y existen diferentes métodos para optimizar su funcionamiento. Es aplicado para problemas inclusive más complicados que el que se está planteando en el presente documento. Asimismo se puede aplicar a ingeniería de transporte, abarcando la planificación de viaje para diferentes rutas, no solamente del sistema de transporte del metro.

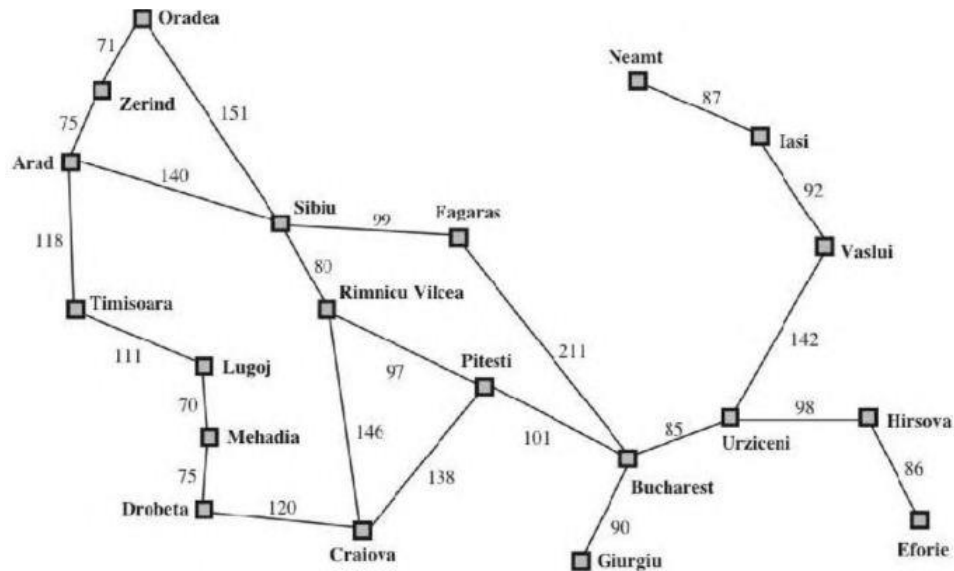


Fig. 2-6 Grafo de un modelo ferroviario

Algoritmo a implementar: búsqueda heurística a lo ancho priorizando menor costo

El algoritmo que se empleará será basado en los algoritmos anteriormente mencionados, aplicando una búsqueda heurística a lo ancho, tomando como prioridad el nodo con el menor costo. No se siguen los algoritmos anteriores al pie de la letra por las siguientes razones:

No sólo necesitamos encontrar la ruta más corta, sino también las mejores alternativas (si las hubiera).

Se toma en cuenta el tiempo de transbordo, (en promedio 5 minutos).

Al encontrar las rutas más cortas se hará un estimado de afluencia de gente.

Después de hacer el análisis, los pasos para seguir el algoritmo de enrutamiento a utilizar son plasmados en el siguiente diagrama de flujo:

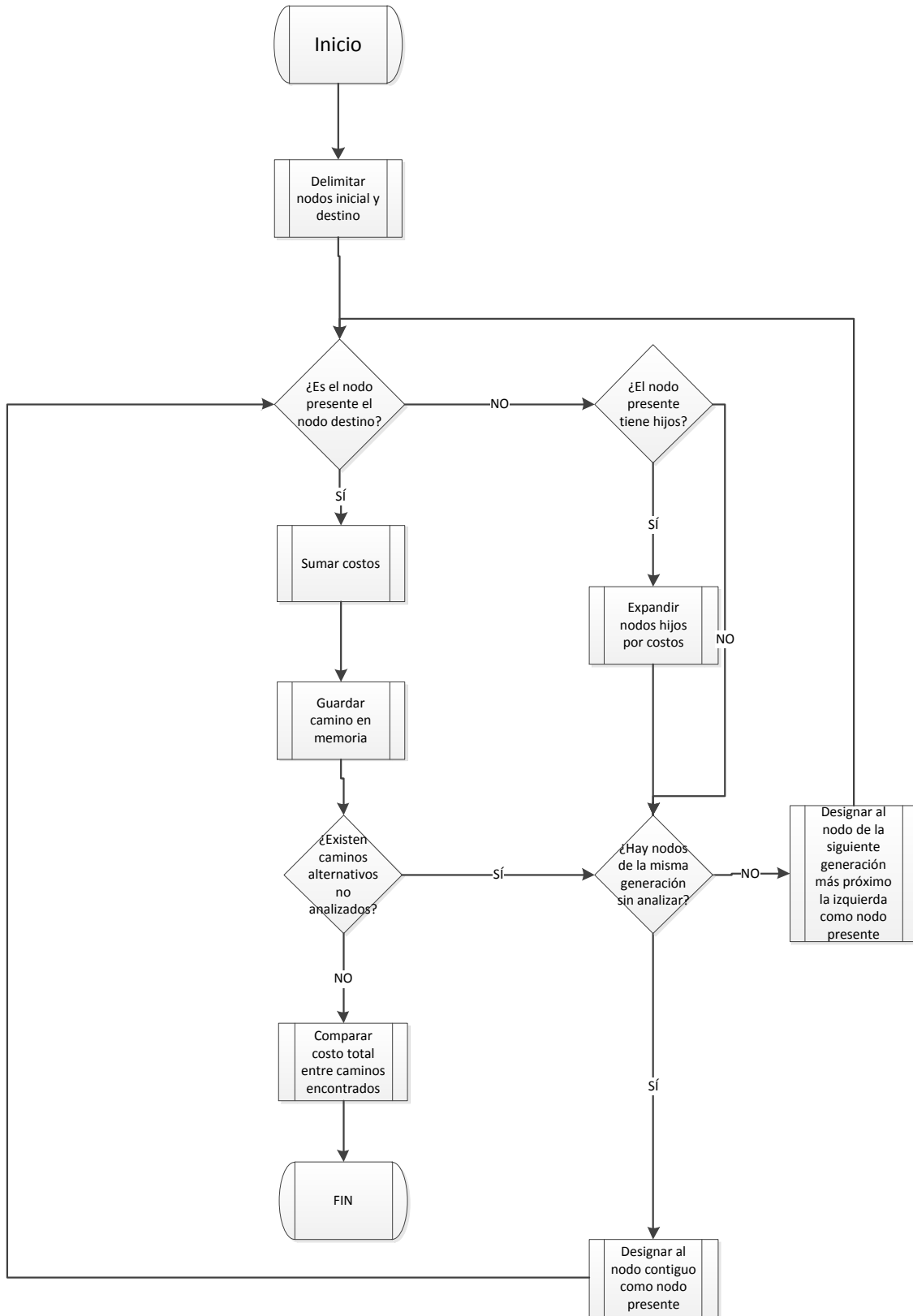


Fig. 2-7 Diagrama de flujo del algoritmo de enrutamiento a utilizar en el proyecto

Es un algoritmo que puede llegar a extenderse bastante, pero el tiempo de ejecución es bastante razonable y un poco mejor que los algoritmos mencionados anteriormente. Al tener un orden de búsqueda a lo ancho, su grado de complejidad será de:

$$O(|V|+|E|)$$

Ejemplo práctico de algoritmo a utilizar

El algoritmo que utilizaremos toma en cuenta el valor de las transiciones entre los nodos, lo cual tomamos de Dijkstra y Prim, pero el orden de apertura de nodos sigue una heurística a lo ancho utilizada en inteligencia artificial; al analizar los nodos irá descartando aquellos que ya han sido analizados e irá introduciendo a otros con la estructura de una cola.

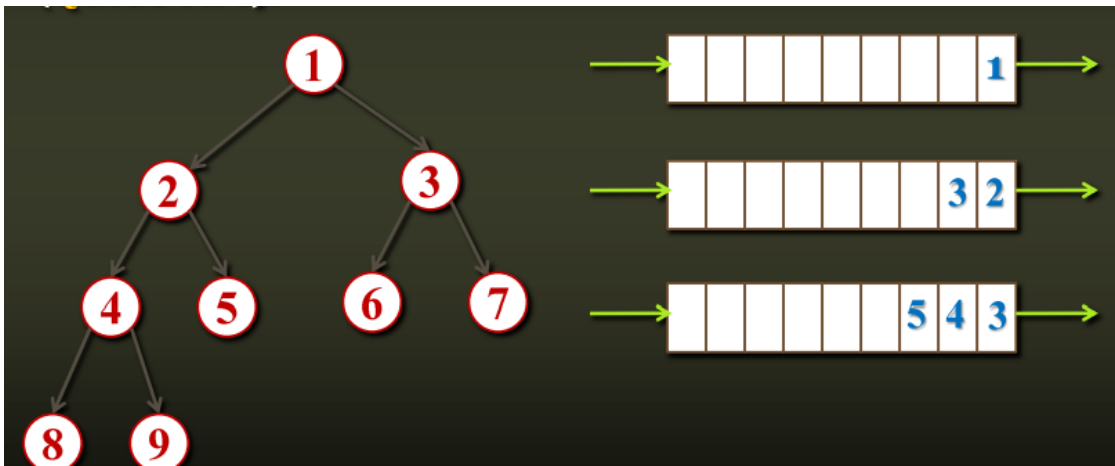


Fig.2-8 Ejemplo de back tracking en búsqueda a lo ancho

Ahora implementemos el algoritmo en una trayectoria cualquiera, esto para demostrar que este algoritmo puede funcionar para cualquier otro grafo del mismo tipo y no sólo para el de nuestro problema.

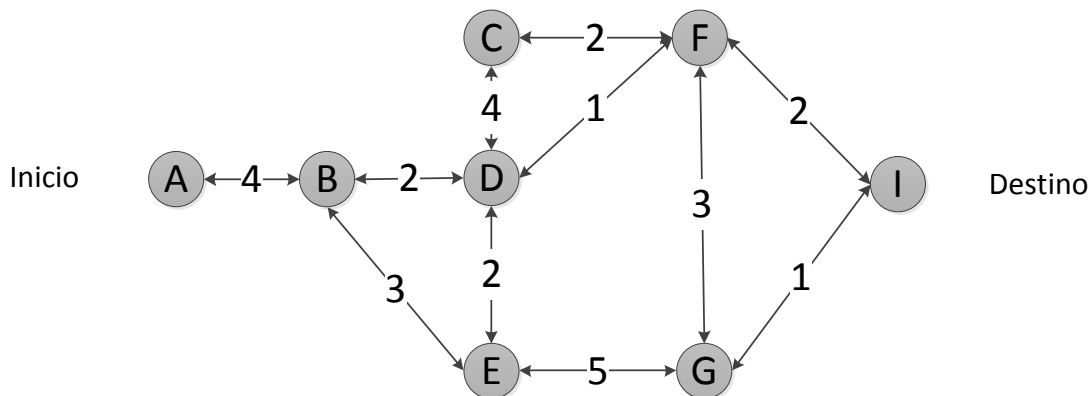


Fig.2-9 Grafo a resolver por método propuesto

Desarrollando el algoritmo, tomando como límite 2 rutas alternativas, obtenemos el siguiente árbol inverso:

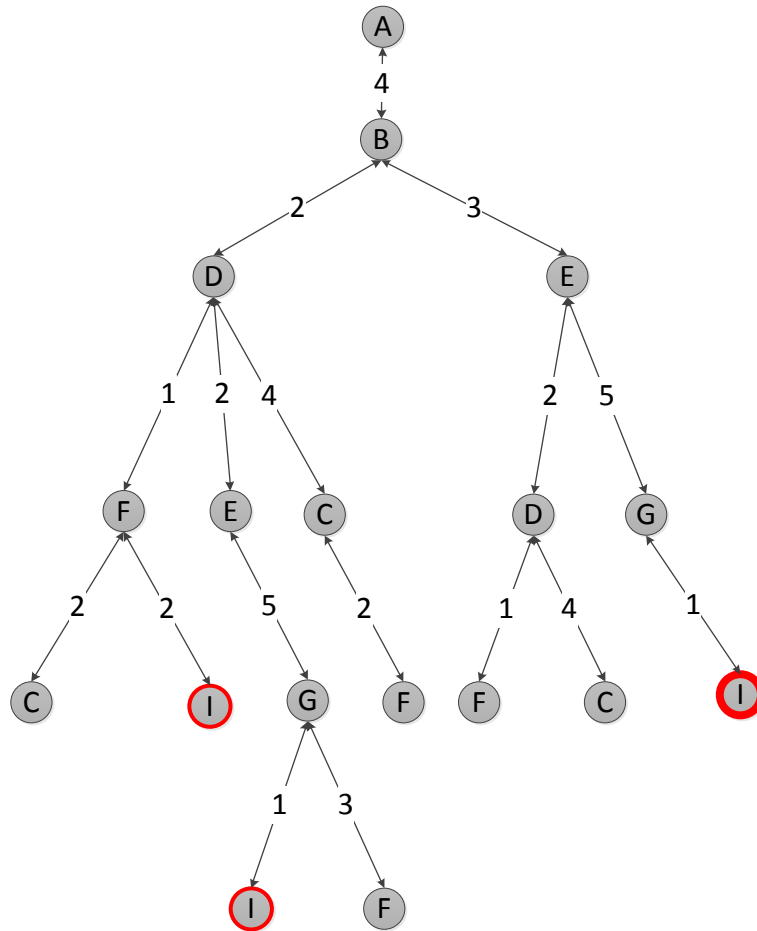


Fig.2-10 Árbol inverso generado por algoritmo

Ahora extraemos la información necesaria del árbol para obtener el costo que requerirá cada camino, el cual se ejemplifica en la siguiente tabla:

| No. Camino | Trayectoria | Costo |
|------------|-------------|-------|
| 1 | A-B-D-F-I | 9 |
| 2 | A-B-D-E-G-I | 14 |
| 3 | A-B-E-G-I | 13 |

Tabla 2-1 Extracción de datos de costo en cada ruta calculada por algoritmo

En este caso, la ruta con menor costo es la primera encontrada, esto debido a la naturaleza del algoritmo implementado. Debemos tomar en cuenta que podríamos obtener más caminos, pero para limitar el consumo de recursos hemos detenido la búsqueda al encontrar 3 rutas. Al probar el algoritmo, es hora de aplicarlo a los datos sobre las distancias entre estaciones que hemos recabado previamente.

Construcción de grafo del STC Metro DF

Los trenes del STC pueden alcanzar una velocidad de hasta 80km/h, pero su velocidad comercial es de sólo 36km/h [12]. La página oficial del STC del DF no nos proporciona el tiempo de trayecto entre estaciones, pero sí nos proporciona la distancia entre estación y estación. Podemos calcular el tiempo promedio de viaje sabiendo la velocidad promedio y la distancia que se quiere recorrer:

$$t=d/V$$

Sin embargo, debemos tomar en cuenta que 36 km/hr no es la velocidad promedio, ya que debemos de tomar en cuenta que el tren se debe detener por completo en cada estación y volver a arrancar. Al hacer varias pruebas en varios trayectos en distintos días, se hizo la relación de los tiempos entre estaciones; la velocidad promedio que mejor se acerca a los resultados obtenidos es la de 25km/hr.

Después de hacer los cálculos, obtenemos un tiempo estimado en el cual se va de estación a estación. La siguiente tabla muestra los resultados obtenidos de una de las líneas más populares: la línea 2.

| Entre estación | Longitud interceptación (metros) | Longitud Entre estación(km) | Tiempo promedio trayecto (Minutos) |
|----------------------------|----------------------------------|------------------------------|------------------------------------|
| Cuatro caminos – Panteones | 1639 | 1,639 | 3,9336 |
| Panteones – Tacuba | 1416 | 1,416 | 3,3984 |
| Tacuba – Cuitláhuac | 637 | 0,637 | 1,5288 |
| Cuitláhuac – Popotla | 620 | 0,62 | 1,488 |
| Popotla – | 462 | 0,462 | 1,1088 |

| | | | |
|--------------------------------|-----|-------|--------|
| Colegio Militar | | | |
| Colegio Militar – Normal | 516 | 0,516 | 1,2384 |
| Normal – San Cosme | 657 | 0,657 | 1,5768 |
| San Cosme – Revolución | 537 | 0,537 | 1,2888 |
| Revolución – Hidalgo | 587 | 0,587 | 1,4088 |
| Hidalgo – Bellas Artes | 447 | 0,447 | 1,0728 |
| Bellas Artes – Allende | 387 | 0,387 | 0,9288 |
| Allende – Zócalo | 602 | 0,602 | 1,4448 |
| Zócalo – Pino Suárez | 745 | 0,745 | 1,788 |
| Pino Suárez – San Antonio Abad | 817 | 0,817 | 1,9608 |
| San Antonio Abad – Chabacano | 642 | 0,642 | 1,5408 |
| Chabacano – Viaducto | 774 | 0,774 | 1,8576 |
| Viaducto – Xola | 490 | 0,49 | 1,176 |
| Xola – Villa de Cortés | 698 | 0,698 | 1,6752 |
| Villa de Cortés – Nativitas | 750 | 0,75 | 1,8 |
| Nativitas – Portales | 924 | 0,924 | 2,2176 |

| | | | |
|-----------------------------|--------------|--------|---------|
| Portales Ermita | 748 | 0,748 | 1,7952 |
| Ermita General Anaya | 838 | 0,838 | 2,0112 |
| General Anaya - Taxqueña | 1330 | 1,33 | 3,192 |
| TOTALES | 17263 | 17,263 | 41,4312 |

Tabla 2-2 Tempo promedio de recorrido de estación a estación de la línea 2

Cualquiera que viaje regularmente en el DF debe estar familiarizado con el trazado de la red del STC. Hasta el momento existen 12 líneas, las cuales se extienden por gran parte de la ciudad, sin embargo en este trabajo vamos a incluir solamente 11 líneas, ya que una de ellas no está en funcionamiento.



Fig. 2-11 Mapa de la red del STC

Debemos considerar que todo camino de punto a punto puede ser representado por un grafo y como vimos en el marco general del documento, este sistema podría aplicarse para redes de transporte similares. El grafo está basado en el mapa de metro y con los datos recabados en el punto anterior, con los cuales obtenemos lo siguiente:

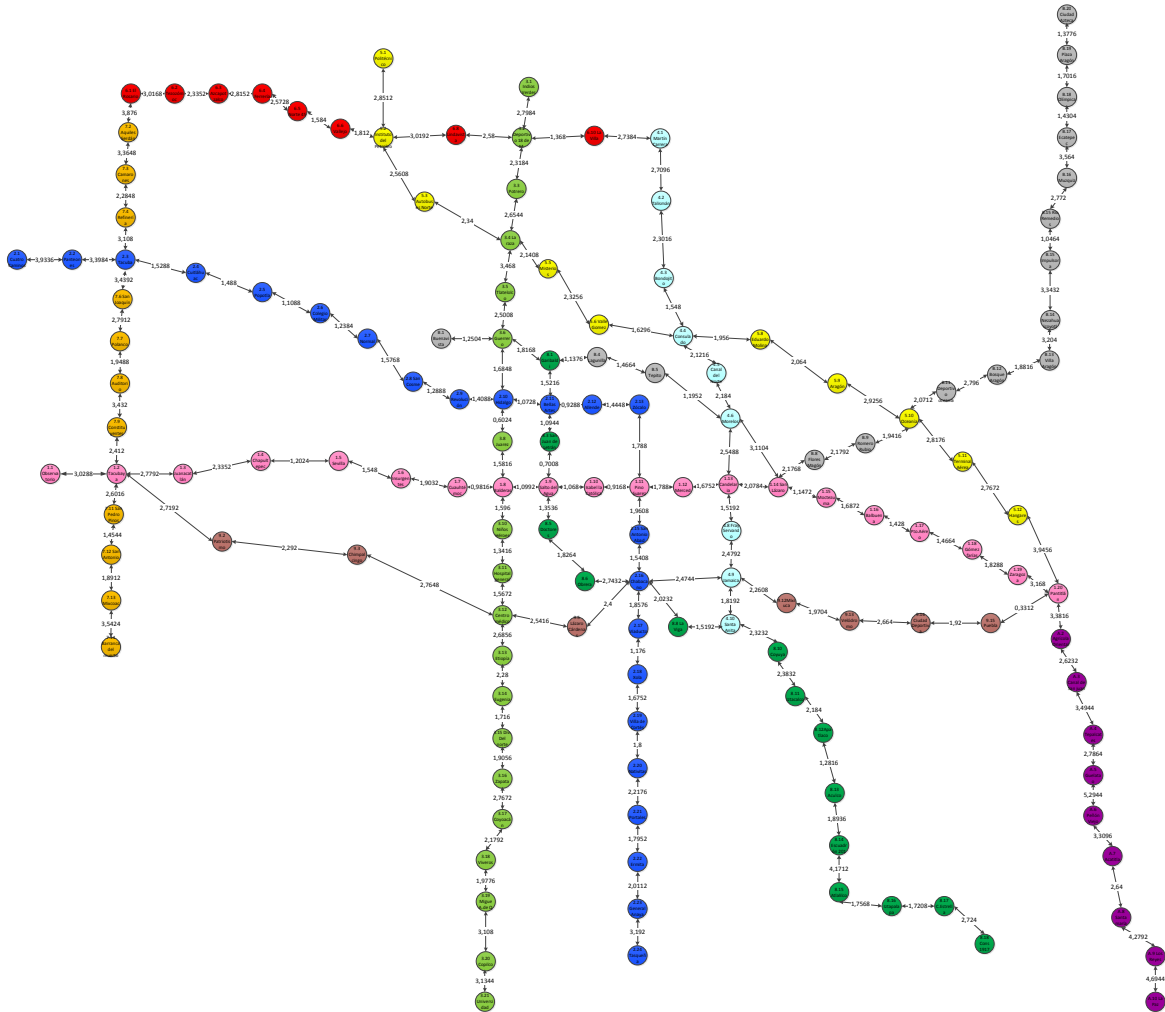


Fig. 2-12 Grafo construido de la red del STC

Si observamos a detalle el grafo, podemos darnos cuenta de que los valores de las transiciones se representan en minutos, y en la mayoría de los casos, hay diferencia de milésimas de segundo entre ellos. En muchos planificadores de trayectos estos tiempos se contemplan por promedios, pero suelen ser inexactos ya que al acumular varios segundos en varias estaciones el tiempo del viaje puede incrementar bastante, sin tomar en cuenta los trasbordos.

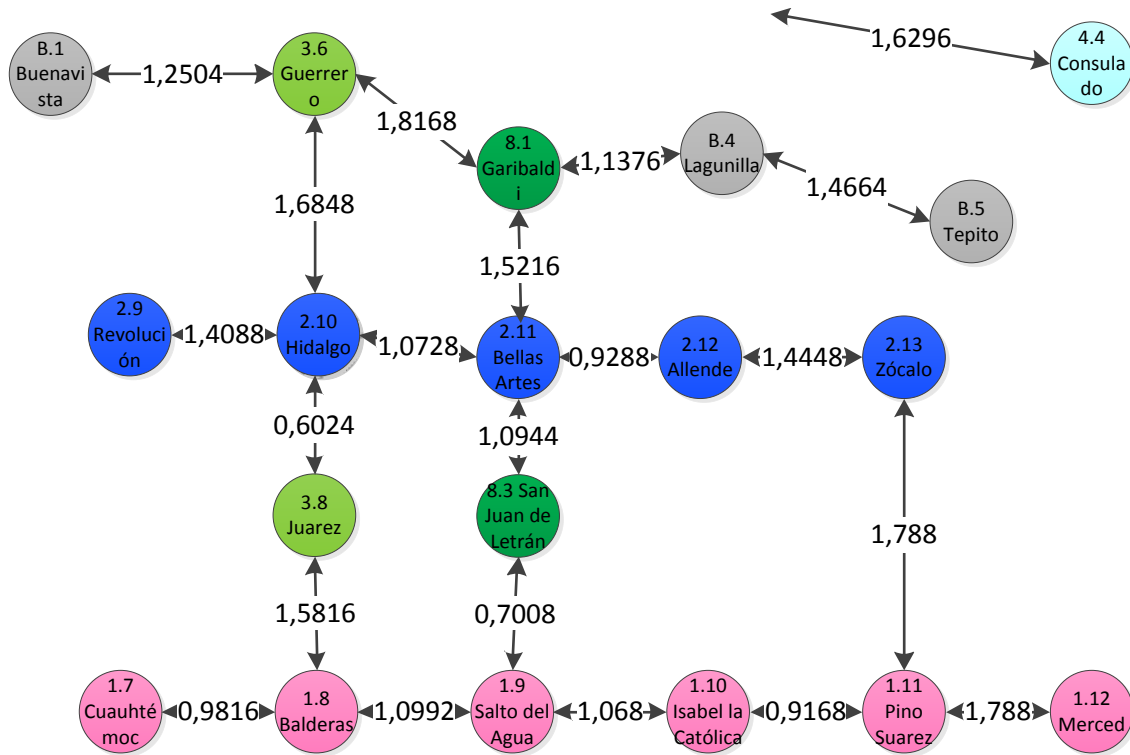


Fig. 2-13 Fragmento del grafo construido del STC

Implementación de algoritmo sobre el grafo construido

En este capítulo hemos hecho el análisis de diferentes algoritmos que podíamos emplear para la atacar el problema que tiene objeto esta investigación; con esto hemos probado un algoritmo basado en los ya analizados y hemos probado que es totalmente viable implementarlo en nuestro problema.

Ahora que hemos construido el grafo de la red del STC Metro DF, es hora de implementarle el algoritmo de búsqueda que hemos escogido. Digamos que queremos ir de la estación Balderas a la estación Garibaldi; a simple vista podemos apreciar que existen varios caminos que podemos tomar.

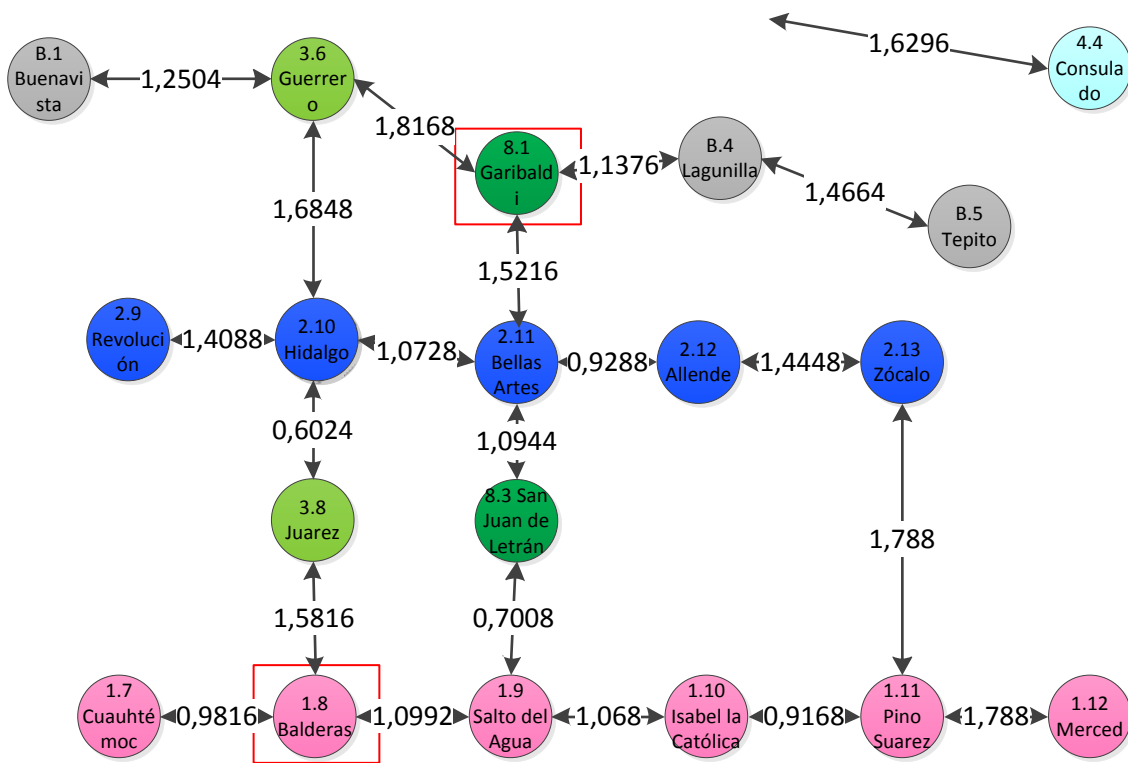


Fig. 2-14 Ejemplificación de nodo inicial y nodo destino dentro del grafo del STC

Un inconveniente con el algoritmo es, que al buscar alternativas, puede encontrar rutas poco viables, entonces una vez que se llegue al nodo destino, si no se encuentra una alternativa después de cierto número de iteraciones, el algoritmo terminará su proceso, de igual manera se terminará el proceso al encontrar 3 caminos diferentes. Es importante mencionar que debido a la metodología que utilizamos en el sistema, el algoritmo puede ser modificado a modo de mejorarlo o inclusive cambiarlo, ya que sistemas como este evolucionan con la implementación y ayuda de la comunidad. Al seguir el algoritmo sobre el anterior grafo, se generará el siguiente árbol inverso:

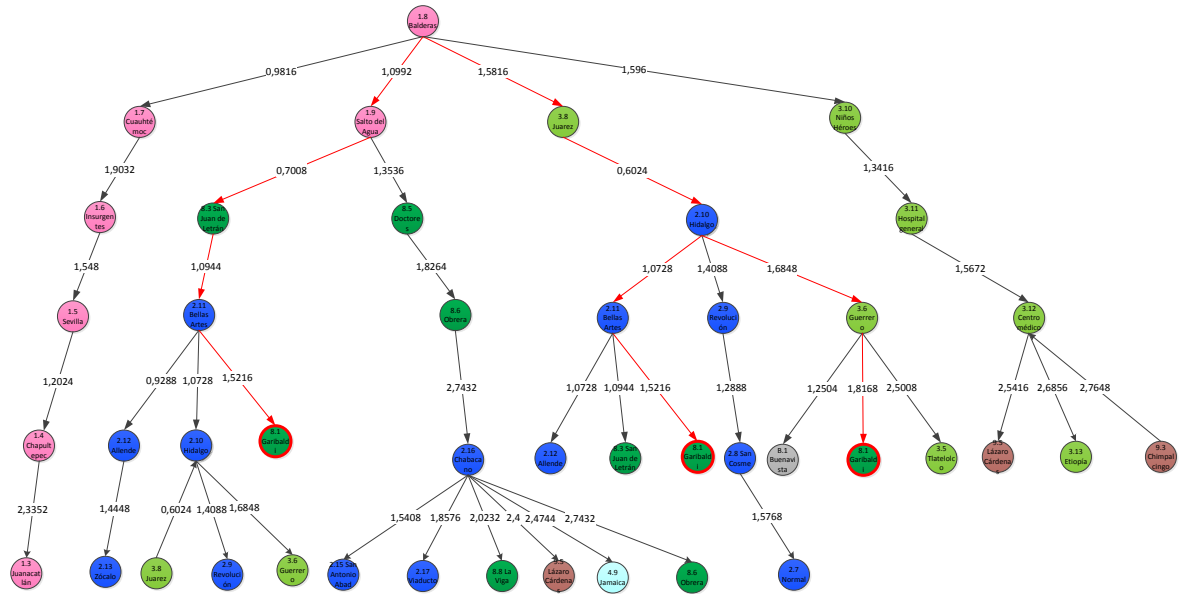


Fig. 2-15 Árbol inverso generado por algoritmo

Al hacer el análisis del árbol podemos ver que se crean 44 nodos y se analizan 28 de estos para llegar a nuestro objetivo. Cabe señalar que al seguir el algoritmo propuesto, detenemos el proceso cuando encontramos 3 diferentes caminos; los cuales son representados por el siguiente árbol simplificado:

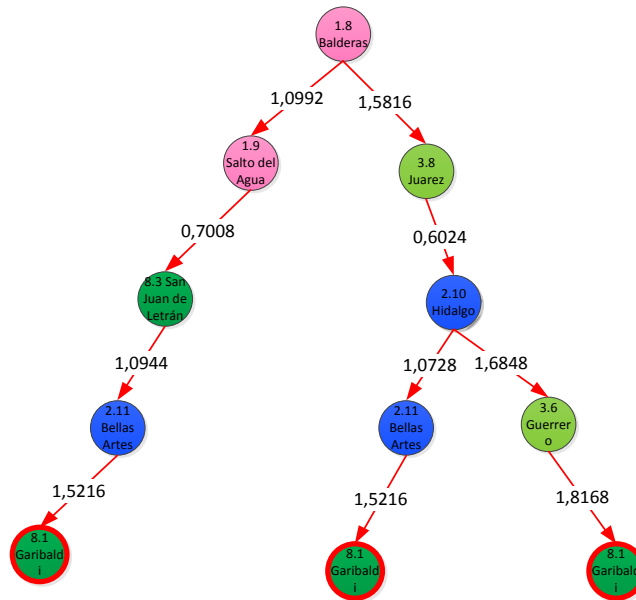


Fig. 2-16 Árbol inverso mostrando las rutas encontradas

Ahora que hemos obtenido 3 diferentes caminos, debemos comparar sus tiempos de recorrido para saber cuál de todos es el más conveniente a tomar.

| No. Camino | Trayectoria | No.trasbordes | Tiempo recorrido(Minutos) |
|------------|-----------------------|---------------|---------------------------|
| 1 | 1.8-1.9-8.3-2.11-8.1 | 1 | 9,416 |
| 2 | 1.8-3.8-2.10-2.11-8.1 | 2 | 14,7784 |
| 3 | 1.8-3.8-2.10-3.6-8.1 | 1 | 10,6856 |

Tabla 2-3 Comparación de tiempos de las rutas encontradas por algoritmo

Como podemos ver, la ruta más corta es el primer camino encontrado por el algoritmo, teniendo como siguiente mejor alternativa el tercer camino encontrado. El segundo camino no es conveniente debido a que se debe trasbordar dos veces, pero en casos de emergencia (cierre de estaciones) puede ser una buena opción. Ahora que hemos explicado el cómo se implementará el algoritmo dentro de la lógica del sistema, es hora de abordar otro tema: predicción de afluencia de pasajeros.

Capítulo 3: Predicción a corto plazo de afluencia de gente

Se busca desarrollar un subsistema de pronóstico de afluencia de pasajeros para el STC Metro que se convierta en una pieza clave de la planeación de los recorridos que realiza cada pasajero sobre la red de transporte ya mencionada. Para eso, debemos seguir el siguiente proceso:

- Estudiar los patrones de comportamiento de la afluencia de pasajeros,
- Construir modelos de series de tiempo para la proyección a corto plazo de la afluencia de pasajeros.
- Encontrar la relación existente entre los distintos patrones de afluencia, con el fin de determinar si esta relación aporta a la proyección de corto plazo.
- Se comparará y elegirá el modelo más adecuado para la proyección de afluencia de pasajeros.

Técnicas de series de tiempo

Los datos con los que se cuenta para esta investigación son series de tiempo, y el área de conocimiento es la ingeniería de transporte, así que es preciso utilizar un método que ya ha sido implementado anteriormente en éste campo para atacar problemáticas similares. Por ende, los modelos a aplicar en primer lugar son los de series de tiempo.

Existen varios métodos de predicción, siendo los más importantes y utilizados:

Promedios Móviles

Suavización Exponencial.

Estos pueden utilizarse en los siguientes contextos: cuando hay información disponible de la variables que se pronostica, la información puede ser cuantificada, si se considera razonable que el patrón de comportamiento del pasado continuará en el futuro, si se cuenta con una base de datos histórica y se desea pronosticar una variable considerando su comportamiento pasado; entonces, es posible utilizar el Método de Promedios Móviles o el Método de Suavización Exponencial, conocidos también como Métodos de Series de Tiempo. [22]

Debido a su eficiencia y simplicidad, el método que se empleará para el diseño de este proyecto será el de promedios móviles.

Método de promedios móviles

La utilización de esta técnica supone que la serie de tiempo es estable, es decir, los datos que la componen se generan sin variaciones importantes entre un dato y otro (error aleatorio=0), lo cual significa que el comportamiento de los datos, aunque muestre un crecimiento o un decrecimiento, lo hará con una tendencia constante.

Al usar el Método de Promedios Móviles se supone que todas las observaciones de la serie de tiempo son igualmente importantes para la estimación del parámetro a pronosticar (en este caso la afluencia de pasajeros). De esta manera, se utiliza como pronóstico para el siguiente periodo el promedio de los n valores pertenecientes a los datos más recientes de la serie de tiempo.

Utilizando una expresión matemática, se obtiene:

$$\text{Promedio Movil} = \frac{\sum(n \text{ valores más recientes de la serie de tiempo})}{n}$$

En la anterior ecuación, el término n indica que, conforme se tiene una nueva observación de la serie de tiempo, se reemplaza la más antigua de la ecuación y se calcula un nuevo promedio. El resultado es un desplazamiento del promedio (un periodo en el futuro), y en la medida que se obtienen nuevos datos, se sustituyen en la fórmula y generan una modificación del valor promedio.

No existe una regla específica que indique cómo seleccionar la base del promedio móvil. Si la variable a pronosticar no presenta variaciones considerables, esto es, si su comportamiento es relativamente estable en el tiempo, se recomienda que el valor de n sea grande. Por el contrario, es aconsejable un valor pequeño de n si la variable muestra patrones cambiantes. El método de promedios móviles es muy útil cuando se tiene información no desagregada. [23]

Implementación de método

El Metro puede calcular la afluencia de gente gracias a unos contadores que se encuentran en los torniquetes que se utilizan para acceder a las estaciones.

Cabe recalcar que el concepto ‘afluencia de pasajeros’ no es lo mismo que

‘demanda de pasajeros’. La diferencia es que la primera es la cantidad de personas que pasa por torniquete, es decir, introduciendo su boleto o pasando su tarjeta; en cambio, la demanda de pasajeros es la cantidad de personas que se encuentran en el andén de Metro, es decir, las personas que pasan por torniquete, más las personas que realizan trasbordos y personas que pasan sin pagar.

e-mail: o0ox_xo0o@live.com

La presente investigación está destinada a pronosticar en corto plazo la afluencia de pasajeros, variable que corresponde a la parte cuantificable de la demanda del STC, ya que los infiltrados y la cantidad de personas que efectúan transbordos no pueden ser medidos con exactitud.

Ahora que tenemos clara la diferencia de ambos conceptos, mostramos los datos obtenidos de la página oficial del STC, los cuales muestran la afluencia de pasajeros por estación de la mayoría de sus estaciones. [24]

Los datos proporcionados por la página oficial del STC muestran valores que suelen variar muy poco, como podemos observar en la siguiente tabla (sólo línea 1).

| Línea | Estación | Afluencia Ene-Mar 2013 | Afluencia Abr-Jun 2013 | Afluencia Jul-Sep 2013 | Afluencia Oct-Dic 2013 | Afluencia Ene-Mar 2014 | Afluencia Abr-Jun 2014 | Afluencia Jul-Sep 2014 |
|-------|--------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| 1 | Pantitlán | 4219304 | 4511551 | 4748221 | 4584581 | 4216211 | 4146125 | 4390184 |
| | Zaragoza | 5549677 | 5812581 | 5604018 | 5390802 | 4840555 | 4766643 | 5013449 |
| | Gómez Farías | 3514376 | 3575359 | 4067735 | 4104656 | 3714663 | 3760404 | 3989984 |
| | Blvd. Puerto Aéreo | 2530941 | 2846322 | 2839344 | 2812173 | 2601428 | 2641443 | 2835394 |
| | Balbuena | 1227622 | 1567543 | 1525886 | 1417888 | 1298745 | 1285789 | 1533932 |
| | Moctezuma | 2597976 | 3289697 | 3065346 | 2517840 | 2266625 | 2258747 | 2762144 |
| | San Lázaro | 2549466 | 2560177 | 2797378 | 2830279 | 2685049 | 2775007 | 2942614 |
| | Candelaria | 2011345 | 2054153 | 2187728 | 2443376 | 2129870 | 2204894 | 2388809 |
| | Merced | 4565356 | 4514473 | 5123510 | 5670969 | 4707784 | 4659428 | 4960292 |
| | Pino Suárez | 2552033 | 2771463 | 3549563 | 3731922 | 2844191 | 2770403 | 3321258 |
| | Isabel Católica | 2059675 | 2255952 | 2310191 | 2296026 | 2291067 | 2097711 | 2130834 |
| | Salto del Agua | 2256075 | 2364994 | 2551175 | 2378691 | 2325596 | 2216593 | 2330713 |
| | Balderas | 1900915 | 2059906 | 2136237 | 2107679 | 1990448 | 1984680 | 2121089 |

| | | | | | | | |
|--------------|---------|---------|---------|---------|---------|---------|---------|
| Cuauhtémoc | 1753946 | 1922201 | 1984909 | 1927699 | 1625514 | 1866218 | 2202333 |
| Insurgentes | 5194437 | 5280619 | 5679201 | 5779488 | 5375246 | 5497469 | 5864024 |
| Sevilla | 2724272 | 3026575 | 3009647 | 2899699 | 2706925 | 2933280 | 3090901 |
| Chapultepec | 4449352 | 5278461 | 5524628 | 5263419 | 4470309 | 5084347 | 5431693 |
| Juanacatlán | 1146191 | 1311605 | 1247149 | 1157321 | 1102631 | 1107371 | 1327607 |
| Tacubaya | 3110389 | 3328969 | 3092704 | 3292681 | 2910218 | 3173938 | 3408639 |
| Observatorio | 6481933 | 6679727 | 6918456 | 7025602 | 6672564 | 6629998 | 7009408 |

Tabla 3-1 Afluencia de pasajeros en Jul-Sep 2014 de la línea 1 del STC Metro DF

Para este estudio se utilizaron ventanas de tiempo de 2, 3, 4,5 y 6 periodos anteriores, los cuales son comparados con un periodo específico para obtener el margen de error y seleccionar la ventana de tiempo apropiada para predecir un valor futuro. En este caso, se aplicará el método con los 6 periodos anteriores a Julio-Septiembre 2014 para después comparar el valor resultante con el de ese periodo. Al hacer los debidos cálculos sobre la afluencia total en cada periodo, obtenemos los siguientes resultados:

| Ventana de tiempo | Promedio móvil | Margen de error |
|-------------------|----------------|-----------------|
| 2 | 376799245 | 5,92 |
| 3 | 386685525 | 3,21 |
| 4 | 391008922 | 2,07 |

| | | |
|---|-----------|------|
| 5 | 397272788 | 0,46 |
| 6 | 394207444 | 1,25 |

Tabla 3-2 Entrenamiento de método de promedios móviles

Al analizar la anterior tabla, nos damos cuenta que la ventana de tiempo que tiene un margen de error menor es la que tiene valor de 5. Debemos tomar en cuenta que estos valores deben ir actualizándose cada nuevo periodo.

Ahora que hemos obtenido la ventana de tiempo adecuada, podemos dar un pronóstico a un nuevo periodo de tiempo. Debemos aplicar la técnica del promedio móvil a los últimos 5 periodos publicados hasta el momento por la página oficial del STC [12].

Al aplicar el método obtenemos un valor 392634451 pasajeros esperados en el próximo periodo. Debemos recalcar que este valor irá cambiando en cada periodo.

Obtención de la ruta con menor afluencia

Al saber la afluencia de cada estación podemos implementar exactamente el mismo algoritmo visto en el capítulo 1, sin embargo, eso repercutiría en un consumo excesivo de recursos. El sistema aplicará el algoritmo de búsqueda una sola vez, y al encontrar la ruta más corta y algunas alternativas, nos indicará cuál de estas tendrá la menor afluencia, así, el usuario podrá elegir cuál de ellas prefiere tomar. Se debe tomar en cuenta también, que demasiada cantidad de gente podría retrasar al usuario en su trayecto, así que en horas pico, sería mejor tomar una ruta alterna evitando tomar estaciones con más afluencia de pasajeros.

Al retomar el ejemplo del punto 2.3, se hace el cálculo de la afluencia de gente en los tres caminos encontrados. Al aplicar la técnica de los promedios móviles con ventana de tiempo de 5 periodos sobre las estaciones por las que se viaja en cada trayecto, obtenemos el siguiente pronóstico:

| Camino | Trayectoria | No.trasbordes | Tiempo de recorrido(Minutos) | Afluencia pronosticada (personas por periodo) |
|--------|----------------------|---------------|------------------------------|---|
| 1 | 1.8-1.9-8.3-2.11-8.1 | 1 | 9,416 | 10966354,8 |

| | | | | |
|---|-------------------------------|---|---------|-----------|
| 2 | 1.8-3.8- 2.10-2.11- 8.1 | 2 | 14,7784 | 8864764,8 |
| 3 | 1.8-3.8- 2.10-3.6- 8.1 | 1 | 10,6856 | 7389876 |

Tabla 3-3 Predicción de afluencia de pasajeros en periodo siguiente de las rutas obtenidas por algoritmo de enrutamiento del capítulo 2

Como podemos ver, la ruta más corta es la que tiene mayor afluencia de pasajeros, y la segunda más corta es la que tiene menor afluencia; la opción 2 sigue siendo la menos conveniente al ser la ruta más larga con una afluencia media. Hemos terminado la parte del marco teórico, ahora tenemos todas las herramientas teóricas que necesitamos para la implementación, el siguiente paso es diseñar el sistema.

Capítulo 4: Diseño de sistema

A lo largo de este documento hemos estudiado los antecedentes, las técnicas y algoritmos necesarios para desarrollar este proyecto, sin embargo, en este punto no podemos empezar a desarrollar el software ya que necesitamos un análisis y diseño para poder implementarlo finalmente sobre las plataformas de desarrollo necesarias para crear nuestro sistema.

Tecnologías principales para desarrollo de sistema

Para el desarrollo de este proyecto, utilizaremos varias tecnologías, algunas de ellas bastante conocidas; el objetivo principal es poner a trabajar todas estas para que en conjunto podamos tener una aplicación móvil que se alimente de la información de un servidor y este mismo le de respuestas a sus peticiones. A continuación, enlistamos las herramientas que utilizaremos en este para el desarrollo de este proyecto:

Android SDK: es el conjunto de herramientas de desarrollo necesario para poder desarrollar software en la plataforma de Android [25]. Se eligió esta plataforma porque es la más utilizada por los usuarios de Smartphone, además es bastante versátil.

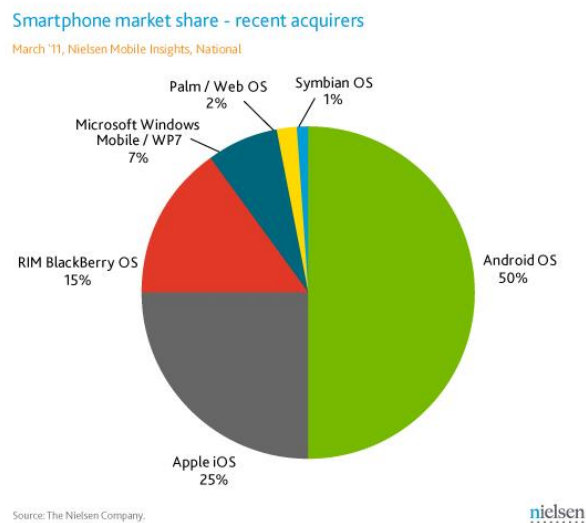


Fig. 4-1 Comparativa de uso de sistemas operativos de usuarios en Smartphones

Apache Tomcat: Es una implementación de software abierto para servlets en java y jsp. Al carecer de un dominio o un servidor dedicado, se emulará el funcionamiento del mismo con esta herramienta. [26]

Google Maps API: Es una interfaz de programación que nos proporciona varias herramientas y servicios para la geo localización, creación y personalización de mapas construidas a base de imágenes satelitales. Se escogió esta plataforma al ser la que cuenta más soporte. [27]



Fig. 4-2 Ejemplo de mapa en Google Maps

Mysql 5.0: Es un gestor de base de datos que nos proporciona las herramientas necesarias para la creación y mantenimiento de bases de datos. Al tener que guardar ciertos datos de usuario y rutas, es necesario en gestor mediana mente robusto como este.

Java: Lenguaje de programación basado en C++ que elimina varias características de reserva de memoria para su mejor uso. Se utilizará para la implementación de los algoritmos utilizados en el sistema.

Fundamentos generales

Los usuarios son parte fundamental de nuestro proyecto, ya que una parte innovadora del sistema es el reporte de incidentes o anomalías en las líneas del STC. Al tener un número de usuarios que contribuyan con los reportes, el funcionamiento de este sistema será mucho más eficiente.

La aplicación va dirigida para todo usuario que tenga un Smartphone con sistema operativo Android con acceso a internet preferentemente y utilice el STC del DF.

La aplicación funcionará únicamente online, ya que existe información que va a ser procesada o capturada por el servidor para poder enviarla al cliente, la cual consultará el usuario para una mejor experiencia de uso. Las razones principales son las siguientes:

e-mail: o0ox_xo0o@live.com

El usuario no podrá acceder si no cuenta con internet

El usuario no podrá hacer reportes si no cuenta con internet

El usuario no podrá consultar acerca de la lluvia si no cuenta con internet

Si el usuario no actualiza la aplicación en un plazo de 3 meses, los pronósticos pueden ser menos precisos, ya que cada 3 meses la información de afluencia de gente va cambiando.

Las restricciones vistas anteriormente son bastante obvias, sin embargo se debe tomar en cuenta que el usuario pueda desconocer que sin acceso internet, muchas opciones de la aplicación no funcionarán. Asimismo, existen varios factores que podrían hacer que el sistema no funcione correctamente:

En muchas estaciones del metro la señal 3G o 4G no funciona apropiadamente

Si el servidor recibe muchas peticiones al mismo tiempo, puede tardar más de lo normal en procesar las mismas e incluso experimentar fallos.

El sistema al ser un prototipo no cuenta con servidores suficientemente potentes, pero como toda aplicación, puede ser expandido y mejorado.

Actores

El sistema contará con dos actores: el usuario y el administrador: el primero accederá al mismo por medio de una aplicación móvil siendo esta misma un cliente que solicite información al servidor; asimismo, el cliente podrá enviar información al servidor sobre las anomalías que el usuario reporte; el segundo podrá actualizar información sobre los datos que se vayan actualizando en el sistema subiendo archivos al servidor mediante una aplicación web, en donde podrá monitorear a los usuarios y los reportes que generan.

| | |
|--------------------|---|
| ACT-01 | Usuario |
| Descripción | Este actor representa al usuario de la aplicación móvil |

| | |
|--------------------|--|
| ACT-02 | Administrador |
| Descripción | Este actor representa al administrador del sistema |

Tabla 4-1 Actores de sistema

Objetivos del sistema

| | |
|-------------|---|
| OBJ-01 | Encontrar la ruta más corta y algunas alternativas |
| Descripción | El sistema deberá aplicar un algoritmo de búsqueda para encontrar la ruta más sus alternativas más viables (si hubiera) basándose en la distancia recorrida en cada una |
| Estabilidad | alta |

| | |
|-------------|--|
| OBJ-02 | Determinar la afluencia esperada en cada una de las rutas encontradas por algoritmo de búsqueda |
| Descripción | El sistema deberá comparar las rutas encontradas por el algoritmo de búsqueda y verificar la afluencia de pasajeros esperada |
| Estabilidad | alta |

| | |
|-------------|---|
| OBJ-03 | Determinar hora Pico |
| Descripción | El sistema deberá verificar la hora en el que el usuario hace la petición e indicarle si es o no hora pico. |
| Estabilidad | alta |

| | |
|-------------|---|
| OBJ-04 | Gestionar información de reportes |
| Descripción | El sistema deberá gestionar la información de los reportes que generen los usuarios sobre las anomalías en alguna estación del metro, haciendo públicos estos a cualquier usuario que ingrese al sistema. |
| Estabilidad | alta |

| | |
|--------|----------------------------------|
| OBJ-05 | Gestionar información de usuario |
|--------|----------------------------------|

| | |
|-------------|---|
| Descripción | El sistema deberá gestionar la información de datos de usuario para que este pueda acceder al sistema de creación de reportes |
| Estabilidad | alta |

| | |
|-------------|---|
| OBJ-06 | Gestionar información de rutas |
| Descripción | El sistema deberá gestionar la información de rutas más populares en la comunidad |
| Estabilidad | alta |

Tabla 4-2 Objetivos del sistema

Requisitos de almacenamiento de información:

Esta sección contiene la lista de requisitos de almacenamiento de información que se han identificado, en las siguientes tablas especificaremos toda la información que debemos almacenar en nuestro sistema.

| | |
|----------------------|--|
| RI-01 | Información sobre afluencia predicha en periodo actual |
| Objetivos asociados | OBJ-02 |
| Requisitos asociados | Subir archivo a servidor |
| Descripción | El administrador deberá subir un archivo en cada periodo, actualizando la información sobre la afluencia de pasajeros. |
| Datos específicos | Nombre de estación Línea Afluencia pronosticada |
| Estabilidad | alta |
| Comentarios | ninguno |

| | |
|---------------------|----------------------------|
| RI-02 | Información sobre usuarios |
| Objetivos asociados | OBJ-05 |

| | |
|----------------------|---|
| Requisitos asociados | Alta usuario Baja usuario Editar usuario Login Usuario Consultar usuarios |
| Descripción | El sistema deberá almacenar la información correspondiente a los usuarios que deseen hacer reportes sobre anomalías en las estaciones del metro |
| Datos específicos | Id de usuario Nombre y apellidos de usuario Fecha de nacimiento Sexo Fecha de alta como usuario Correo User Password |
| Estabilidad | alta |
| Comentarios | ninguno |

| | |
|----------------------|--|
| RI-03 | Información sobre reportes |
| Objetivos asociados | OBJ-04 |
| Requisitos asociados | Creación de reporte Baja de reporte Calificación de reporte Consulta de reporte |

| | |
|-------------------|---|
| | |
| Descripción | El sistema deberá almacenar la información correspondiente a los reportes que crean los usuarios, y monitorear su reputación. |
| Datos específicos | Id Reporte Tipo de anomalía Estación de anomalía Id usuario que reporta Puntos positivos Puntos negativos |
| Estabilidad | alta |
| Comentarios | Esta tarea está sujeta a si el usuario cuenta con conexión a internet o no |

| | |
|----------------------------|---|
| RI-04 | Información sobre rutas populares |
| Objetivos asociados | OBJ-06 |
| Requisitos asociados | Encontrar ruta Registrar ruta Consultar ruta Almacenar ruta en caché |
| Descripción | El sistema deberá almacenar la información correspondiente a las rutas almacenadas por otros usuarios previamente, con el fin de no consumir demasiados recursos y almacenarlos en el caché del usuario que consulta. |
| Datos específicos | Id Ruta Estación inicial Estación destino Trayecto Id usuario que calcula primero dicha ruta |

| | |
|-------------|--|
| Estabilidad | alta |
| Comentarios | Esta tarea está sujeta a si el usuario cuenta con conexión a internet o no |

| | |
|----------------------------|--|
| RI-05 | Información sobre estaciones |
| Objetivos asociados | OBJ-06 |
| Requisitos asociados | Registrar ruta Registrar Reporte |
| Descripción | El sistema deberá asociar los reportes y las rutas almacenados con información sobre las estaciones. |
| Datos específicos | Id Ruta Nombre estación Líneas Es_cruce |
| Estabilidad | alta |
| Comentarios | Esta tarea está sujeta a si el usuario cuenta con conexión a internet o no |

Tabla 4-2 Requisitos de almacenamiento de información

Al analizar los objetivos y los requisitos de almacenamiento de información, podemos darnos cuenta que se generan varios procesos diferentes que se encargan de la gestión de da diferentes tipos de dato, cada uno de esos procesos serán ejecutados por diferentes módulos o subsistemas que trabajarán en conjunto para un funcionamiento óptimo del sistema, dichos subsistemas se ejemplifican a continuación:



Fig. 4-3 Diagrama de subsistemas

Diagramas de casos de uso

En esta sección hemos incluido los diagramas de casos de uso de nuestro sistema; cabe mencionar que un sistema de este tipo puede generar muchos más casos de uso de los que se presentan, pero varios de ellos serían bastante parecidos entre ellos, por lo que sólo incluimos los más importantes:

El primer caso de uso es el más simple, o mejor dicho, el que menos procesos implica, se trata de cuando el usuario entra a la aplicación sólo para interactuar con el mapa e indicar los puntos de inicio y destino para que el sistema calcule la ruta más corta.

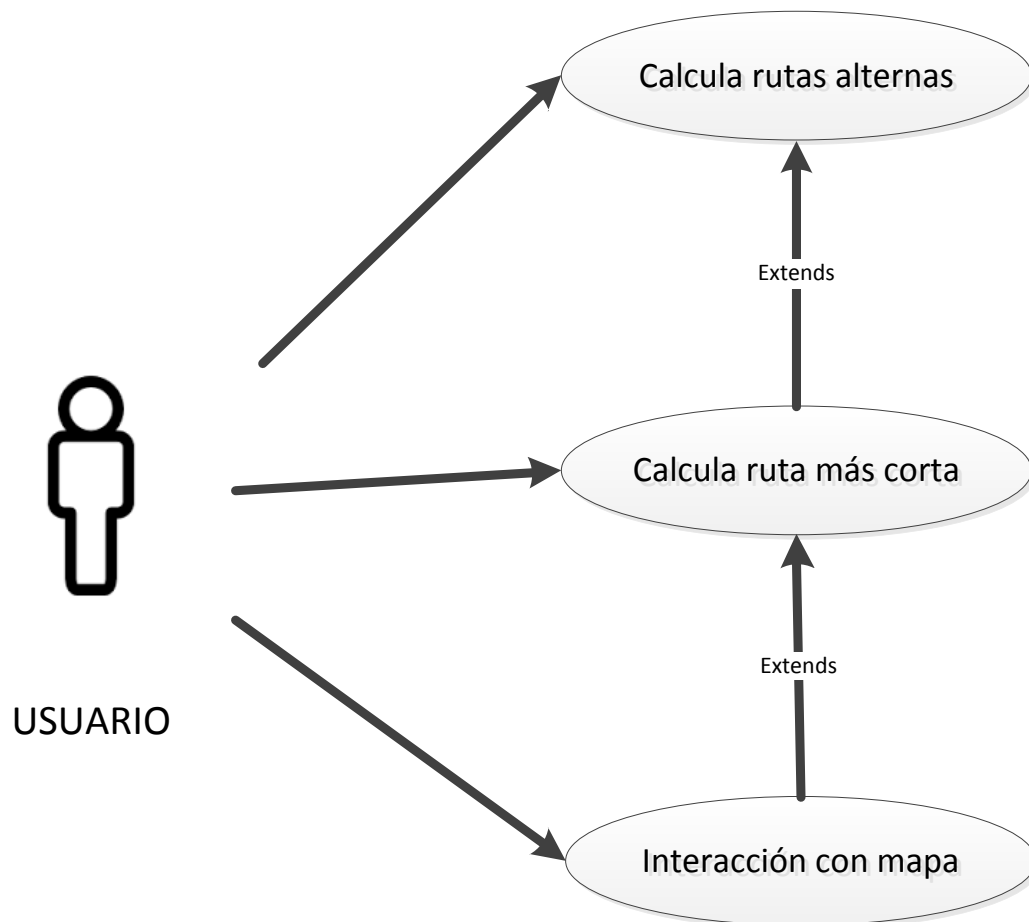


Fig. 4-4 Diagrama de casos de uso del subsistema cálculo de ruta más corta

El segundo caso de uso es cuando el usuario también quiere saber la afluencia de pasajeros predicha en una ruta que haya puesto a calcular al sistema previamente. Para actualizar la información deberá iniciar sesión, lo cual le dará el permiso de descargar el archivo de actualización que el administrador del sistema irá cambiando periódicamente.

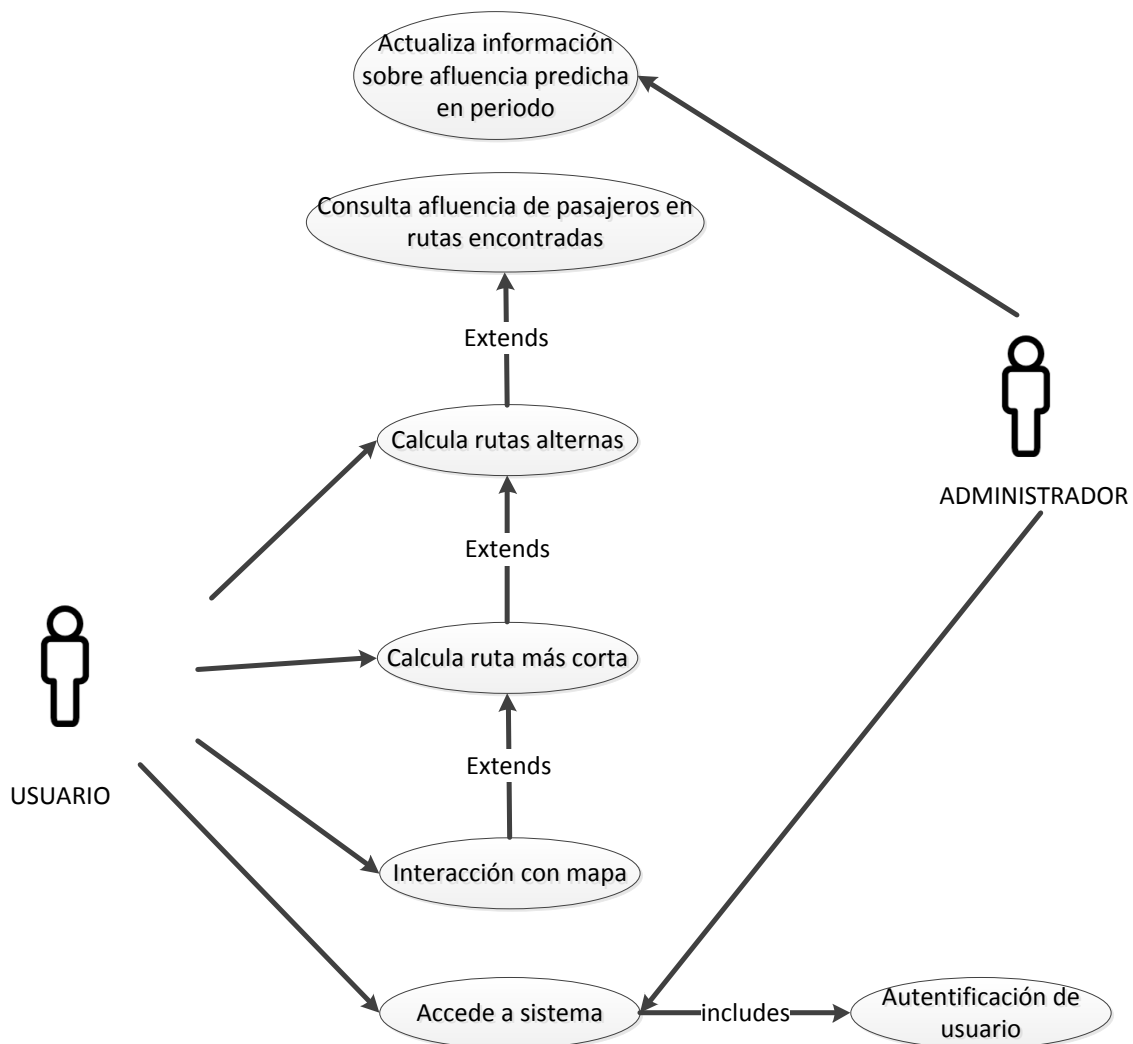


Fig. 4-5 Diagrama de casos de uso de subsistema comparación de afluencia predicha en rutas encontradas

El siguiente caso de uso sería el de gestión de reportes, el cual sería creado por el usuario en caso de encontrar anomalías en su trayecto. Estos reportes serían consultados por los demás usuarios y en ciertos casos serían calificados por los mismos. En caso de un reporte tenga puntos negativos, el administrador puede bloquear el reporte e inclusive al usuario que lo crea.

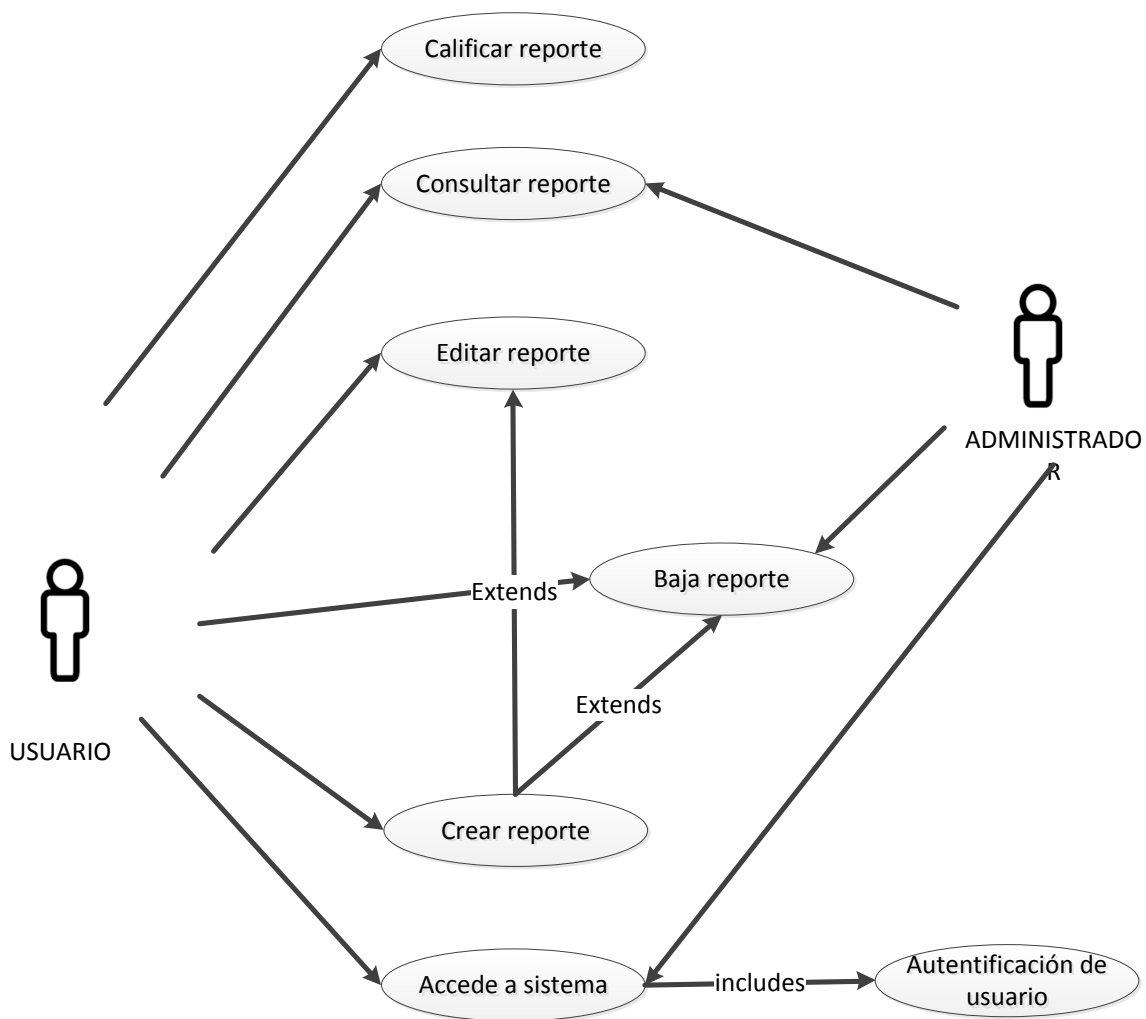


Fig. 4-6 Diagrama de casos de uso de subsistema gestión de reportes

El siguiente caso de uso es el de gestión de usuarios, en el cual el mismo usuario puede darse de alta en el sistema, editar su perfil e inclusive darse de baja. El administrador tendrá derecho de consultar cierta información de cualquier usuario en el sistema.

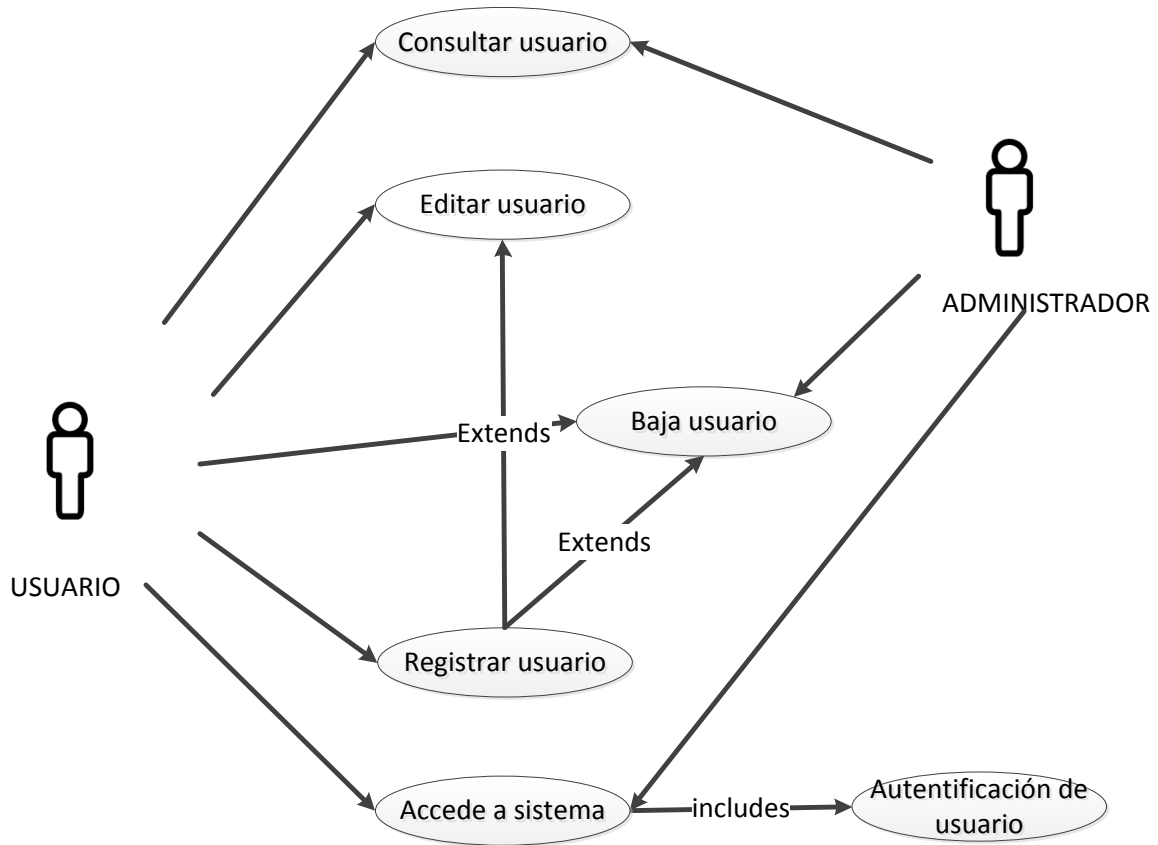


Fig. 4-7 Diagrama de caos de uso de gestión de usuarios

El último caso de uso que presentamos en este documento es el de gestión de rutas populares, en el cual el sistema irá almacenando en caché las rutas más usadas por el usuario, y de ser posible, subirlas al servidor. Esto con el propósito de ahorrar recursos y no calcular una misma ruta más de una vez.

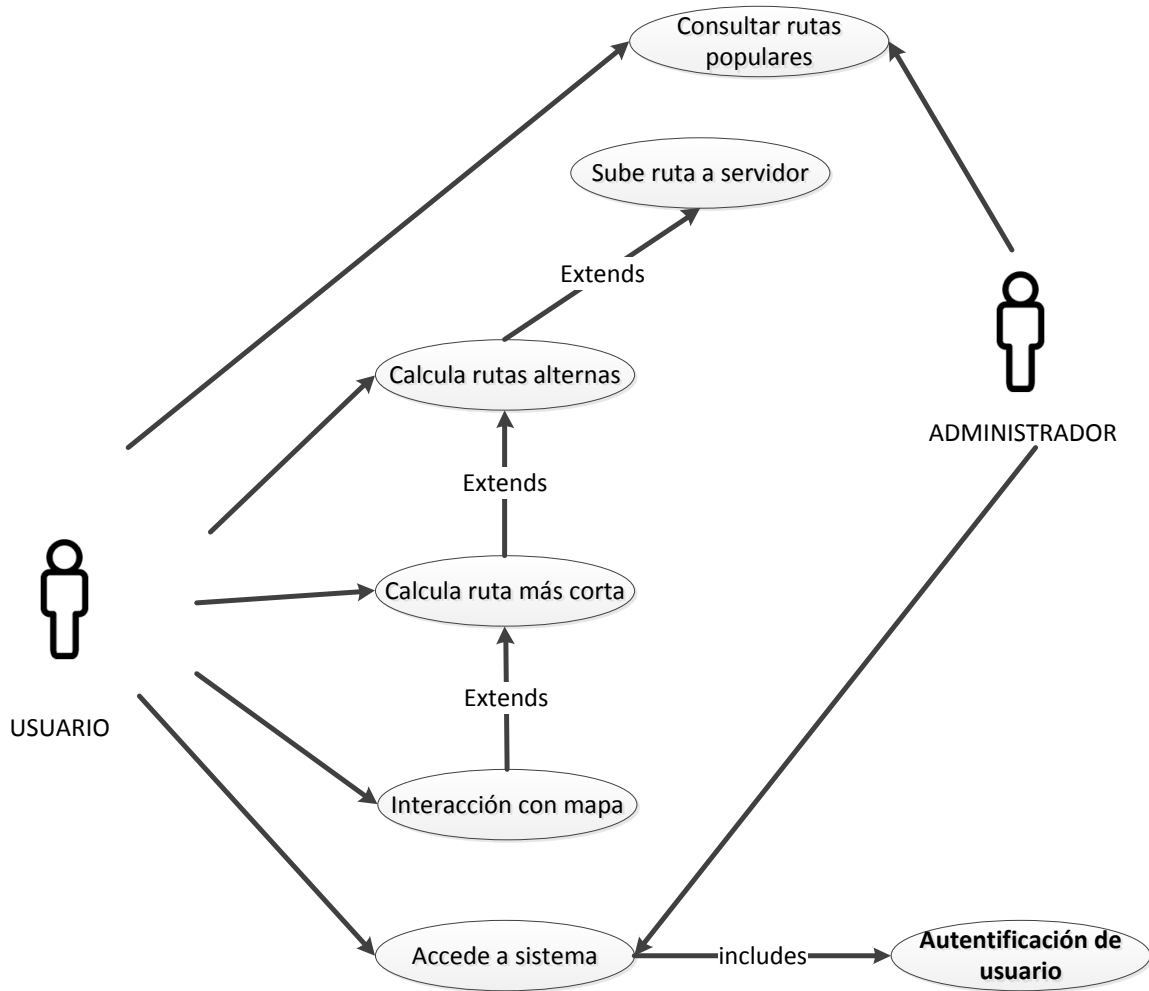


Fig.4-8 Diagrama de casos de uso de gestión de rutas populares

Diagrama de clases UML

Al ser un sistema desarrollado en java es necesario explicar la interacción entre las clases que componen el sistema, para eso nos apoyamos del diagrama UML mostrado a continuación:

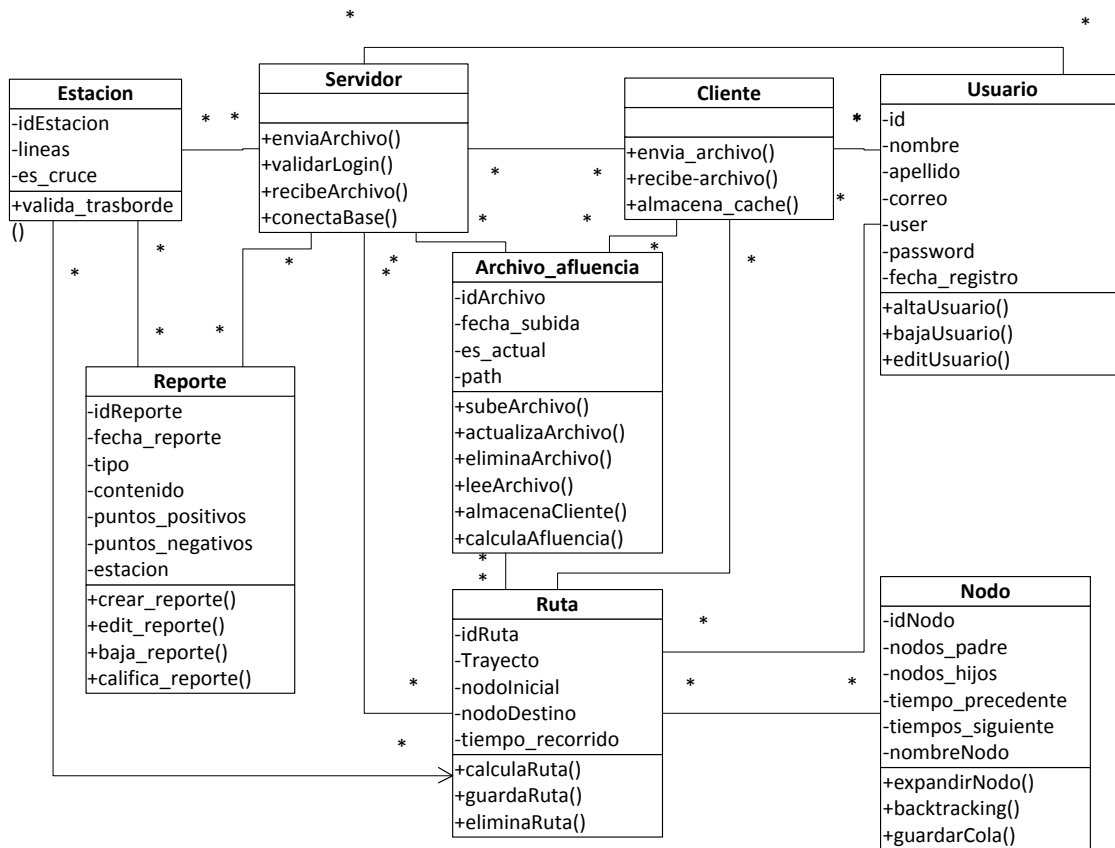


Fig. 4-9 Diagrama UML del sistema

Arquitectura del sistema

Como se ha mencionado anteriormente, el sistema se compone de 2 entidades: servidor y cliente.

Las tareas principales de cada entidad son las siguientes:

Cliente

- Interactuar con el usuario a través de una interfaz gráfica, en donde el usuario indicará la estación de inicio y la estación destino, creará reportes, etc.
- Calcular la ruta más corta y sus alternativas (si existieran)
- Calculará la afluencia de las rutas obtenidas en el punto anterior (necesita actualizar los datos periódicamente)
- Enviar al servidor reportes sobre anomalías

e-mail: o0ox_xo0o@live.com

- Capturar datos de registro de usuario y enviar dichos datos al servidor
- Capturar datos de acceso, recibir respuesta de validación de servidor
- Consultar anomalías reportadas por otros usuarios conectándose con el servidor
- Consultar clima del día del trayecto planificado conectándose al servidor

Servidor

- Recibir datos sobre rutas que el cliente calcule
- Acceso a base de datos
- Enviar al cliente actualizaciones sobre afluencia de gente cada periodo
- Procesar anomalías reportadas por los clientes y dejarlas listas para consulta
- Validar acceso, registrar datos de usuario a base de datos
- Consumir datos sobre clima por medio de API de clima y enviar los datos procesados al cliente.

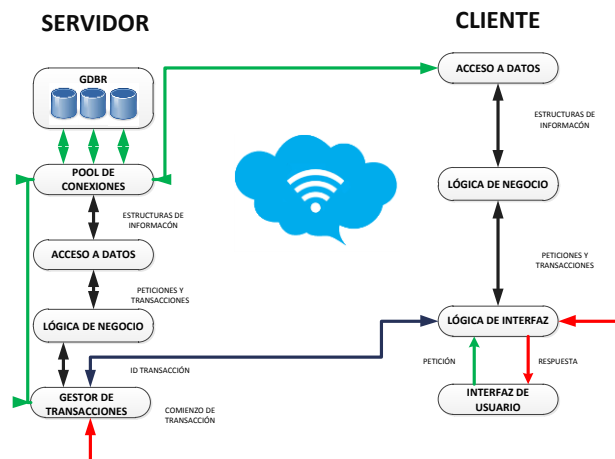


Fig.4-10 Arquitectura general cliente-servidor

Tomando en base la arquitectura cliente servidor, construimos la arquitectura del sistema que estamos diseñando, tomando en cuenta los módulos más importantes y cómo se comunican entre sí.

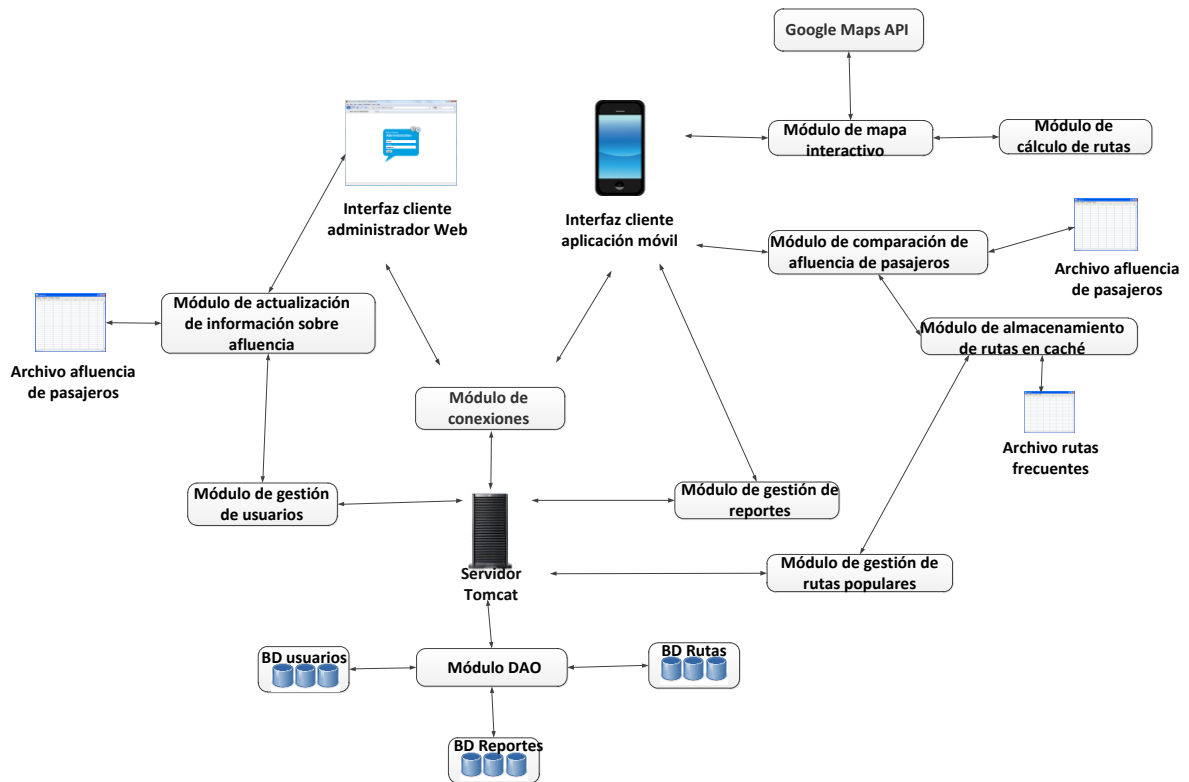


Fig. 4-11 Arquitectura del sistema

Modelo relacional de la base de datos

Debido al tiempo de ejecución que conlleva el algoritmo de búsqueda y a la cantidad de datos a consultar, algunos módulos no trabajarán con bases de datos, lo cual no es un gran inconveniente pues los datos a procesar son datos públicos. Los módulos que trabajará con base de datos será el módulo de creación de reportes, almacenamiento de rutas, y el acceso a sistema. Estos módulos serán procesados por la información que reciba el servidor de los clientes.

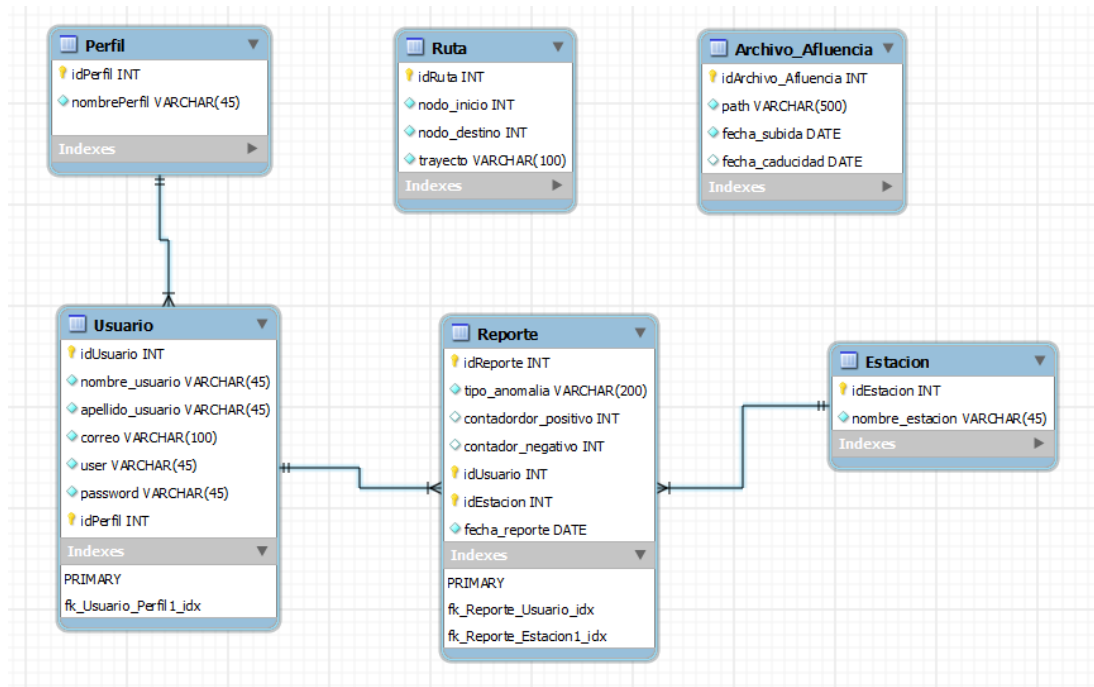
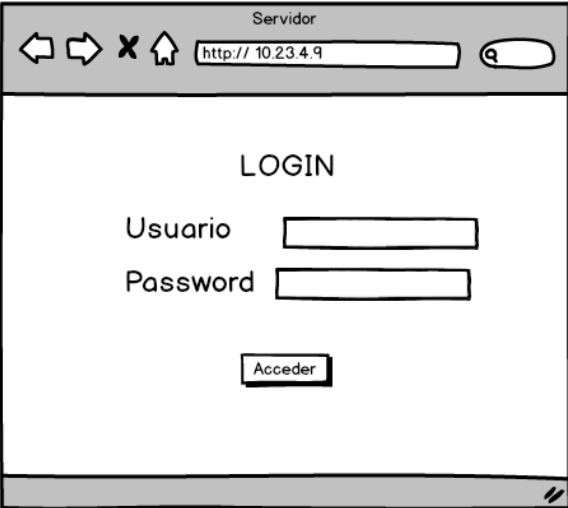


Fig. 4-12 Modelo relacional de la base de datos

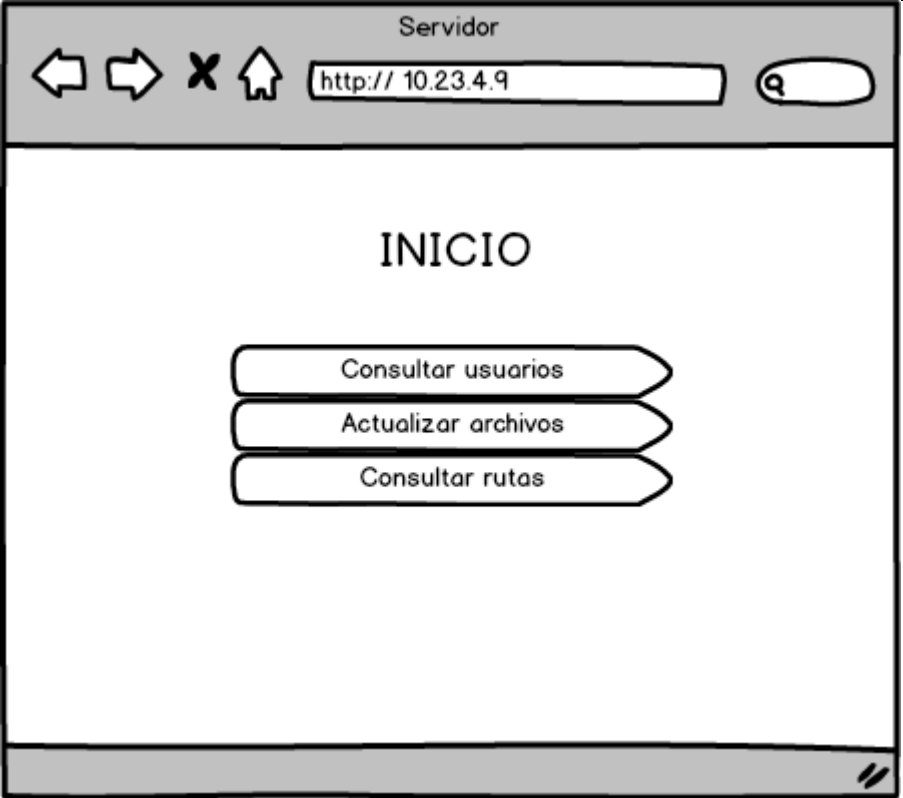
Pantallas de diseño

El sistema contará con diversas pantallas que interactuarán con el usuario, para que este pueda introducir o consultar datos, interactuar con el mapa de metro o simplemente desee interactuar con el sistema. Las principales pantallas se describen en las siguientes tablas:

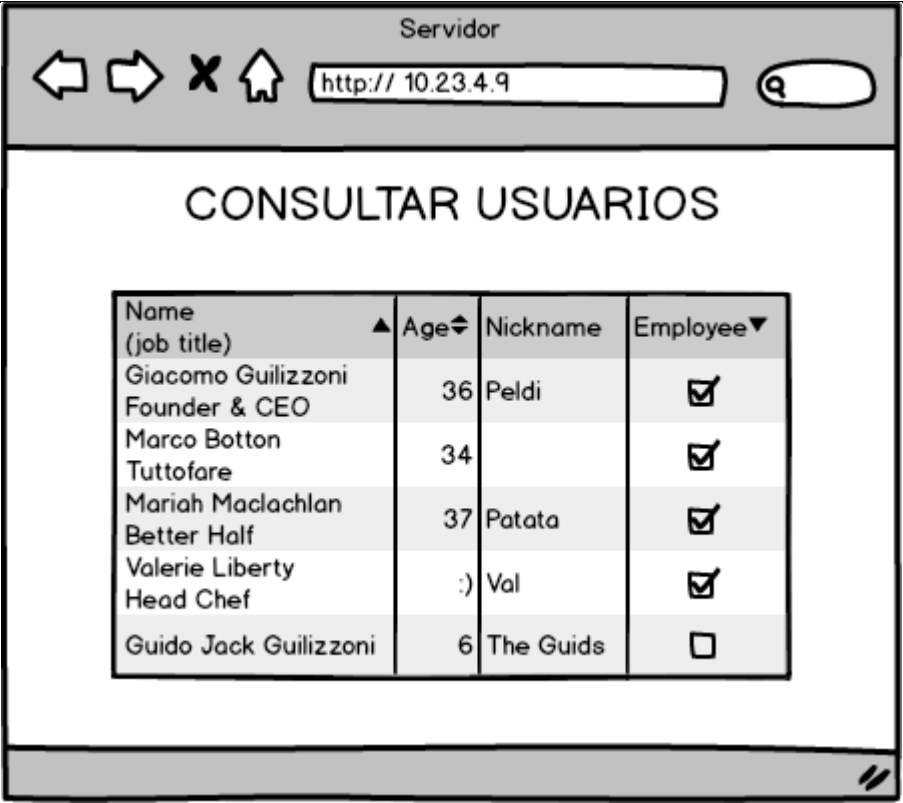
| | |
|------------------|--|
| Nombre | Login (Administrador). |
| Objetivo | Permitir al administrador tener acceso al sistema. |
| Pantalla. | |

| | |
|-------------------------------|---|
| |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Cuadro de texto para introducir nombre de usuario • Cuadro de texto para introducir password. |
| Flujos. | <ol style="list-style-type: none"> 1. Usuario. Introduce nombre de usuario 2. Usuario. Introduce su password 3. Sistema. Valida datos introducidos por usuario, si son correctos permitirá al usuario entrar al sistema, de lo contrario regresamos al flujo 1 |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla |
| Observaciones. | Pantalla solo visible para el usuario Administrador. |

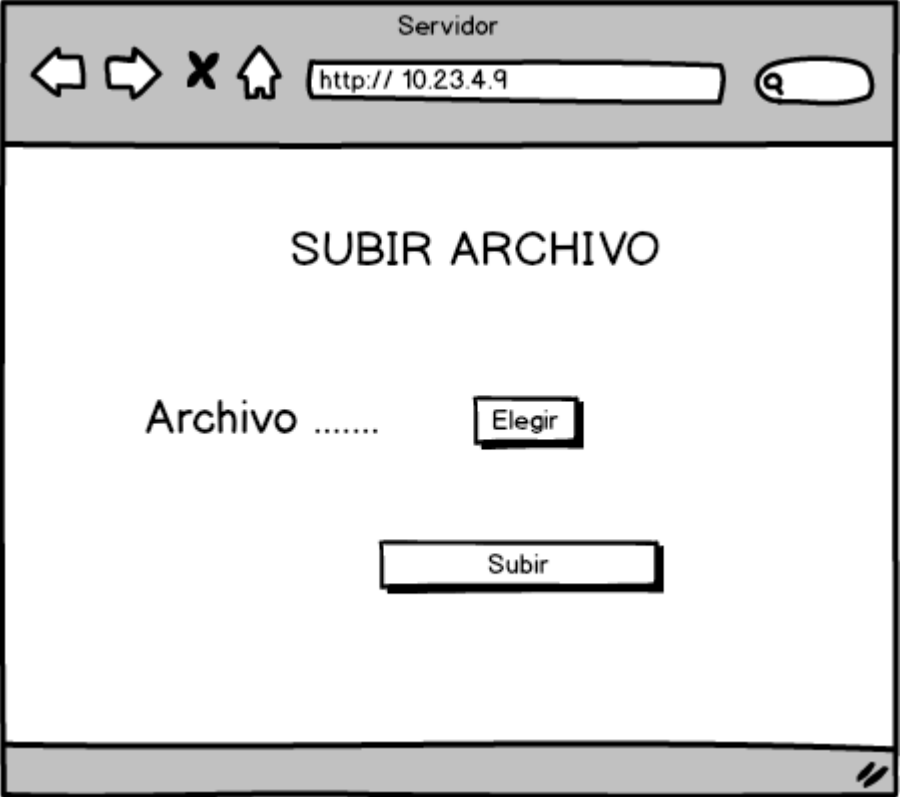
| | |
|-----------------|--|
| Nombre | Inicio (Usuario Administrador). |
| Objetivo | Permitir al administrador tener acceso al sistema. |

| | |
|-------------------------------|--|
| Pantalla. |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Menú vertical para elegir tarea • Botón Logout |
| Flujos. | <ol style="list-style-type: none"> 1. Usuario. Presiona botón con opción de la tarea que desea realizar 2. Sistema. Muestra pantalla de la tarea indicada por el usuario |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla |
| Observaciones. | Pantalla solo visible para el usuario Administrador. |


| | |
|-----------------|---|
| Nombre | Consultar usuarios (Administrador) |
| Objetivo | Permite al usuario consultar la información de los usuarios registrados en el sistema |


| Pantalla. |  <table border="1" data-bbox="634 537 1308 919"> <thead> <tr> <th>Name (job title)</th> <th>Age</th> <th>Nickname</th> <th>Employee</th> </tr> </thead> <tbody> <tr> <td>Giacomo Guilizzoni Founder & CEO</td> <td>36</td> <td>Peldi</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Marco Botton Tuttofare</td> <td>34</td> <td></td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Mariah Maclachlan Better Half</td> <td>37</td> <td>Patata</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Valerie Liberty Head Chef</td> <td>:)</td> <td>Val</td> <td><input checked="" type="checkbox"/></td> </tr> <tr> <td>Guido Jack Guilizzoni</td> <td>6</td> <td>The Guids</td> <td><input type="checkbox"/></td> </tr> </tbody> </table> | Name (job title) | Age | Nickname | Employee | Giacomo Guilizzoni Founder & CEO | 36 | Peldi | <input checked="" type="checkbox"/> | Marco Botton Tuttofare | 34 | | <input checked="" type="checkbox"/> | Mariah Maclachlan Better Half | 37 | Patata | <input checked="" type="checkbox"/> | Valerie Liberty Head Chef | :) | Val | <input checked="" type="checkbox"/> | Guido Jack Guilizzoni | 6 | The Guids | <input type="checkbox"/> |
|-------------------------------------|--|---------------------|-------------------------------------|----------|----------|-------------------------------------|----|-------|-------------------------------------|---------------------------|----|--|-------------------------------------|----------------------------------|----|--------|-------------------------------------|------------------------------|----|-----|-------------------------------------|-----------------------|---|-----------|--------------------------|
| Name (job title) | Age | Nickname | Employee | | | | | | | | | | | | | | | | | | | | | | |
| Giacomo Guilizzoni Founder & CEO | 36 | Peldi | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Marco Botton Tuttofare | 34 | | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Mariah Maclachlan Better Half | 37 | Patata | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Valerie Liberty Head Chef | :) | Val | <input checked="" type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Guido Jack Guilizzoni | 6 | The Guids | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Tabla con información de usuarios • Botón de baja de usuario. | | | | | | | | | | | | | | | | | | | | | | | | |
| Flujos. | <ol style="list-style-type: none"> 1. Sistema. Muestra información sobre usuarios registrados 2. Usuario. Consulta información 3. Usuario. Si el administrador desea eliminar a algún usuario sólo da clic sobre el botón eliminar 4. Sistema. Ejecuta las tareas que le indique el administrador en torno a la gestión de usuarios | | | | | | | | | | | | | | | | | | | | | | | | |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla | | | | | | | | | | | | | | | | | | | | | | | | |
| Observaciones. | Pantalla solo visible para el usuario Administrador. | | | | | | | | | | | | | | | | | | | | | | | | |

| | |
|-----------------|--|
| Nombre | Actualizar archivo de afluencia (Administrador). |
| Objetivo | Permitir al administrador subir un archivo con información necesaria sobre la afluencia de pasajeros |

| | |
|-------------------------------|--|
| Pantalla. |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Encabezado • Botón para elegir archivo • Botón para subir archivo a servidor |
| Flujos. | <ol style="list-style-type: none"> 1. Usuario. Elige archivo a subir 2. Usuario. Da clic sobre el botón de subida para ejecutar dicha tarea 3. Sistema.: Valida si el formato del archivo es correcto 4. Sistema.: Determina al archivo más nuevo como el archivo a usar en predicción |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla |
| Observaciones. | Pantalla solo visible para el usuario Administrador. |

| | |
|-----------------|--|
| Nombre | Login (Usuario). |
| Objetivo | Permitir al usuario tener acceso al sistema. |

| | |
|-------------------------------|---|
| Pantalla. |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Cuadro de texto para introducir nombre de usuario • Cuadro de texto para introducir password. |
| Flujos. | <ol style="list-style-type: none"> 4. Usuario. Introduce nombre de usuario 5. Usuario. Introduce su password 6. Sistema. Valida datos introducidos por usuario, si son correctos permitirá al usuario entrar al sistema, de lo contrario regresamos al flujo 1 |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla |
| Observaciones. | Pantalla solo visible para el usuario normal sin privilegios. |

| | |
|-------------------------------|---|
| Nombre | Selección de ruta (Usuario). |
| Objetivo | Permitir al usuario seleccionar la estación de inicio a la estación destino |
| Pantalla. |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Mapa de Google Maps • Mapa del metro con elementos seleccionables en cada estación |
| Flujos. | <ol style="list-style-type: none"> 7. Usuario. Interactúa con el mapa y elige la estación de inicio 8. Usuario. Interactúa con mapa y elige la estación destino 9. Sistema. Muestra las rutas más cortas y su afluencia de pasajeros |
| Condiciones iniciales. | No es necesario contar con conexión a internet para usar esta pantalla, pero el archivo de afluencia de pasajeros debe actualizarse cada periodo. |
| Observaciones. | Pantalla solo visible para el usuario sin privilegios |

| | |
|-----------------|--|
| Nombre | Crear reporte (Usuario). |
| Objetivo | Permitir al usuario crear un reporte sobre alguna anomalía en determinada estación |


| | |
|-------------------------------|---|
| Pantalla. |  |
| Descripción. | <p>La pantalla de inicio cuenta con los siguientes componentes:</p> <ul style="list-style-type: none"> • Combo box para elegir tipo de anomalía de usuario • Cuadro de texto para introducir comentario |
| Flujos. | <ol style="list-style-type: none"> 10. Usuario. Elige la estación del mapa 11. Usuario. Elige opción 'Crear reporte' 12. Usuario. Elige tipo de anomalía 13. Usuario: Escribe comentario acerca del reporte 14. Sistema. Valida datos introducidos por usuario 15. Sistema. Registra reporte en base de datos |
| Condiciones iniciales. | Se debe contar con conexión a internet para usar esta pantalla |
| Observaciones. | Pantalla solo visible para el usuario sin privilegios. |

Tabla. 4-4 Pantallas de sistema

Capítulo 5: Demostración de prototipo

Como se planteó en un principio, el objetivo de este documento era desarrollar un prototipo de una aplicación móvil que facilitara al usuario sus recorridos en el STC. Todo esto fundamentado con la investigación previa, y el marco teórico ya expuesto anteriormente, seguido de la implementación y diseño del sistema.

El prototipo expuesto en este capítulo, como ya se ha mencionado, se desarrolló sobre Android usando principalmente Java en la codificación y Apache Tomcat para correr el cliente web; sin embargo cabe mencionar que no son las únicas tecnologías que se implementaron. A continuación se mencionan algunas de ellas:

-Phongap

Se usó principalmente para crear módulos web y traducirlos a una aplicación móvil capaz de consumir servicios externos.

-JSON

Se utilizó principalmente para manejar las peticiones de los servicios generados por la App móvil y devolvérselos en un formato legible para la misma.

-JavaScript

Se utilizó tanto en el cliente Web como para procesar la información generada por los JSON.

-Apache Server

Los servicios generados por PHP corren sobre este servidor de aplicaciones. Cabe mencionar que no se debe confundir Apache server y Apache Tomcat Server. En este Caso se utiliza el primero para levantar servicios y el segundo para correr la aplicación Web.

-Geometry library (Google Maps)

Se utiliza para calcular las distancias esféricas (ya que la Tierra no es plana) y así poder dar un resultado mejor en cuanto al cálculo de la distancia del recorrido.

-JSP

Utilizadas principalmente para la aplicación web. Bastante manejable, ya que se puede utilizar el lenguaje java dentro de ellas.

-PHP

Se utiliza para levantar los servicios; en este caso, PHP se es el que se comunica directamente con la base de datos, codifica el resultado a JSON y lo envía a la App móvil.

Sin más preámbulos, se procede a explicar el funcionamiento de los principales casos de uso del prototipo, las mismas fueron tomadas directamente de un dispositivo con Android 4.2:

Pantalla de Login

En la captura de pantalla mostrada debajo, podemos observar un login de sistema, en el cual el usuario accederá a la aplicación. Si no se cuenta con una cuenta, se puede crear una nueva.



Figura 5-1 Pantalla de Login

Pantalla principal

Si accedemos exitosamente, el usuario podrá ver del lado derecho a un 70% del ancho de pantalla el mapa interactivo del STC, con sus líneas representadas por sus respectivos colores. A la izquierda, se puede apreciar el logotipo de la aplicación seguido el botón Reset (con el cual se reiniciará mapa) y el panel con los últimos 5 reportes levantados por los usuarios, asimismo por un botón con el cual el actor puede crear uno nuevo.

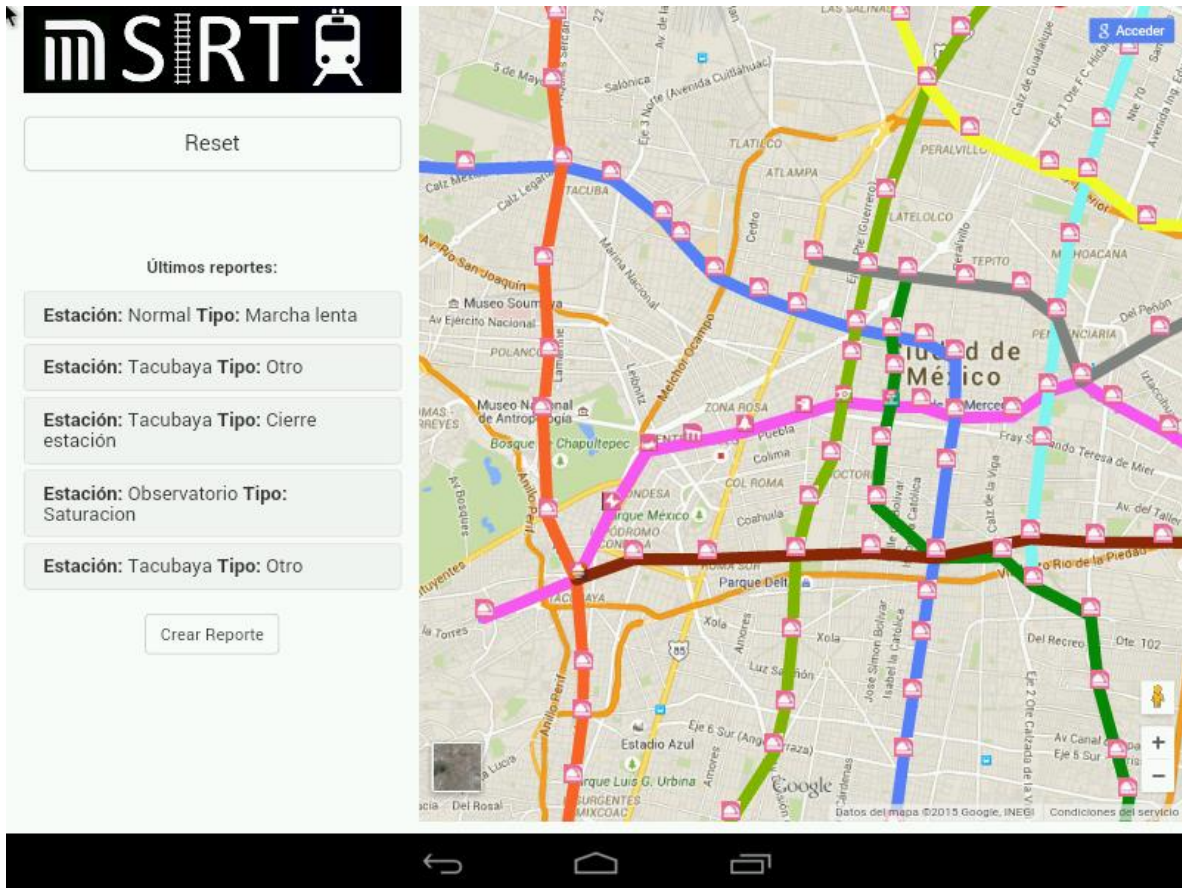


Figura 5-2 Pantalla principal de sistema

Consultar información de una estación en específico:

Si el actor quiere consultar la información de determinada estación, basta con dar un solo 'tap' al marcador ubicado en ella. Se desplegará una ventana de información con el nombre, id, y logotipos de la estación y líneas donde se encuentra. En el panel izquierdo de nuevo se mostrará el nombre de la estación y su afluencia promedio predicha en el periodo actual.

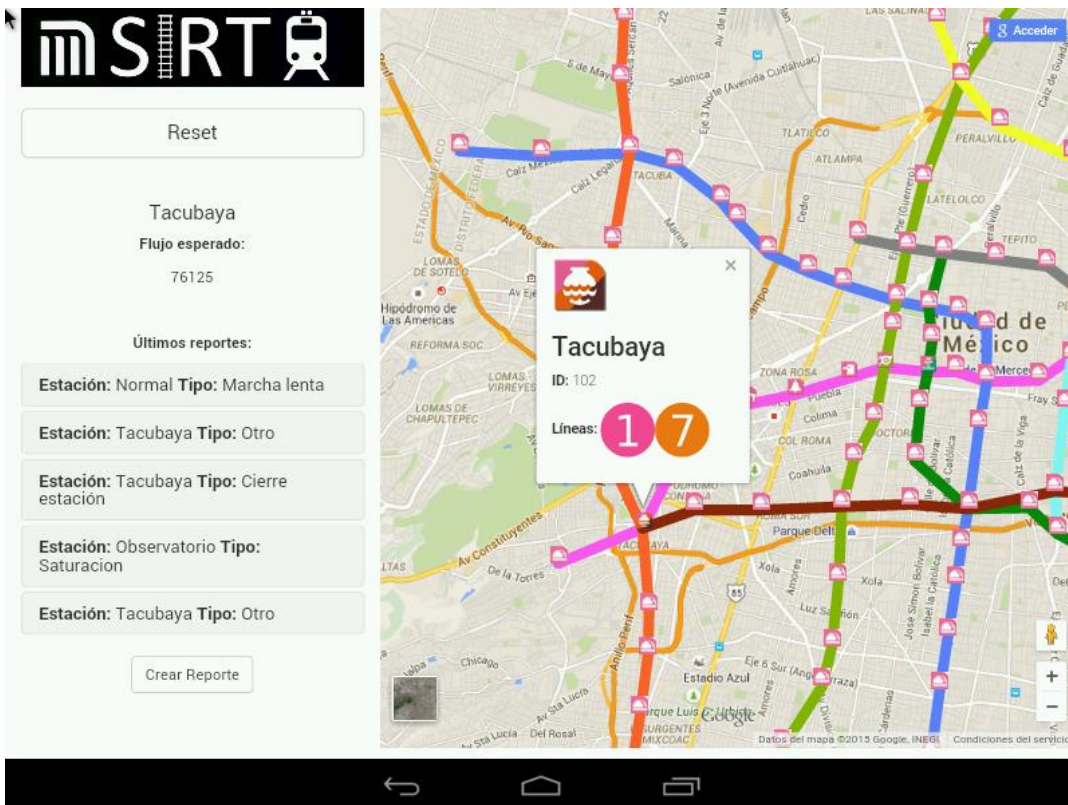
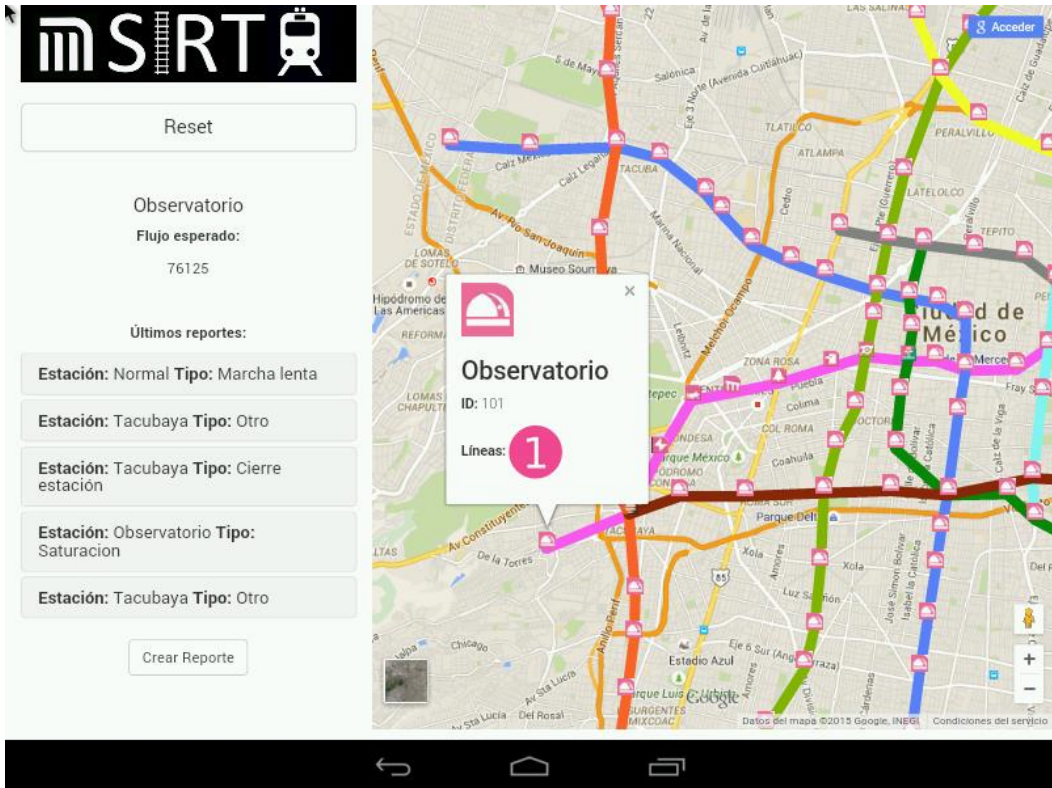


Figura 5-3 Consulta de información de estaciones

Consultar últimos reportes levantados

Para consultar los últimos reportes levantados en el sistema, basta sólo con seleccionar el menú del panel izquierdo debajo de la etiqueta “Últimos reportes”. Se debe seleccionar el incidente que sea del interés del usuario, el cual se guiará por el encabezado del menú desplegable, el cual contiene el nombre de la estación de donde se levantó el reporte y el tipo de incidente. En seguida, al seleccionar el encabezado, se desplegarán los detalles del mismo, los cuales incluyen: descripción, fecha y hora en la que se levantó.

The screenshot displays the SIRT mobile application interface. On the left, there is a sidebar menu with the following elements:

- Reset** button.
- Station name: **Tacubaya**.
- Flujo esperado:** 76125.
- Últimos reportes:**
 - Estación: Normal Tipo: Marcha lenta** (2015-06-01 a las 21:57:35) - Description: **El tren está detenido desde hace media hora, no sabemos que hacer.**
 - Estación: Tacubaya Tipo: Otro**
 - Estación: Tacubaya Tipo: Cierre estación**
 - Estación: Observatorio Tipo: Saturacion**
 - Estación: Tacubaya Tipo: Otro**

On the right, a map of Mexico City shows the SIRT network with various colored lines and station markers. The Tacubaya station is highlighted in blue. The map includes labels for various districts and landmarks, such as Chapultepec, Polanco, and the Museo Nacional de Antropología.

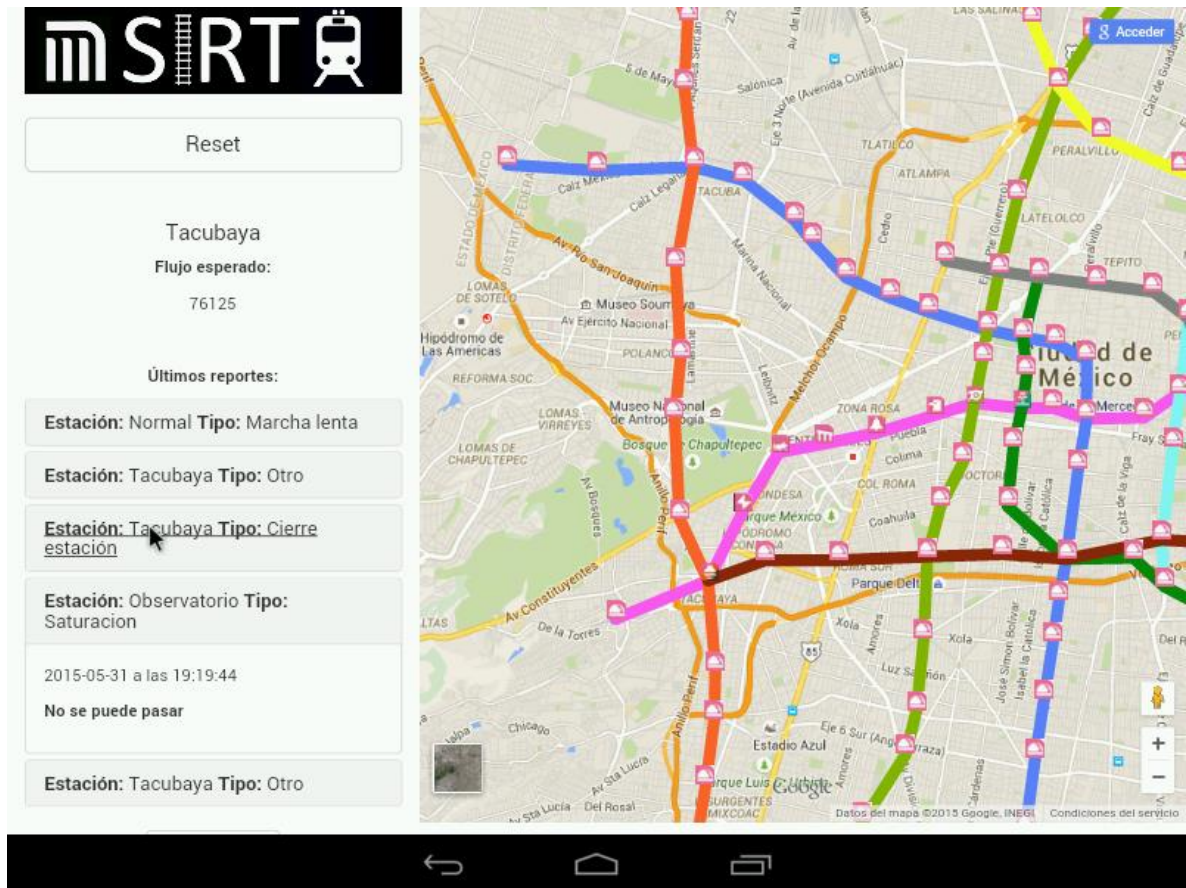


Figura 5-4 Consulta de reportes sobre anomalías en estaciones

Interactuar con mapa

La aplicación corre sobre Google Maps, sin embargo el usuario tiene la libertad de disfrutar de los beneficios que ya vienen incluidos en la misma. Poder navegar por el mapa interactivo, acceder a Google Maps y consultar datos de geo localización son algunas de ellas.

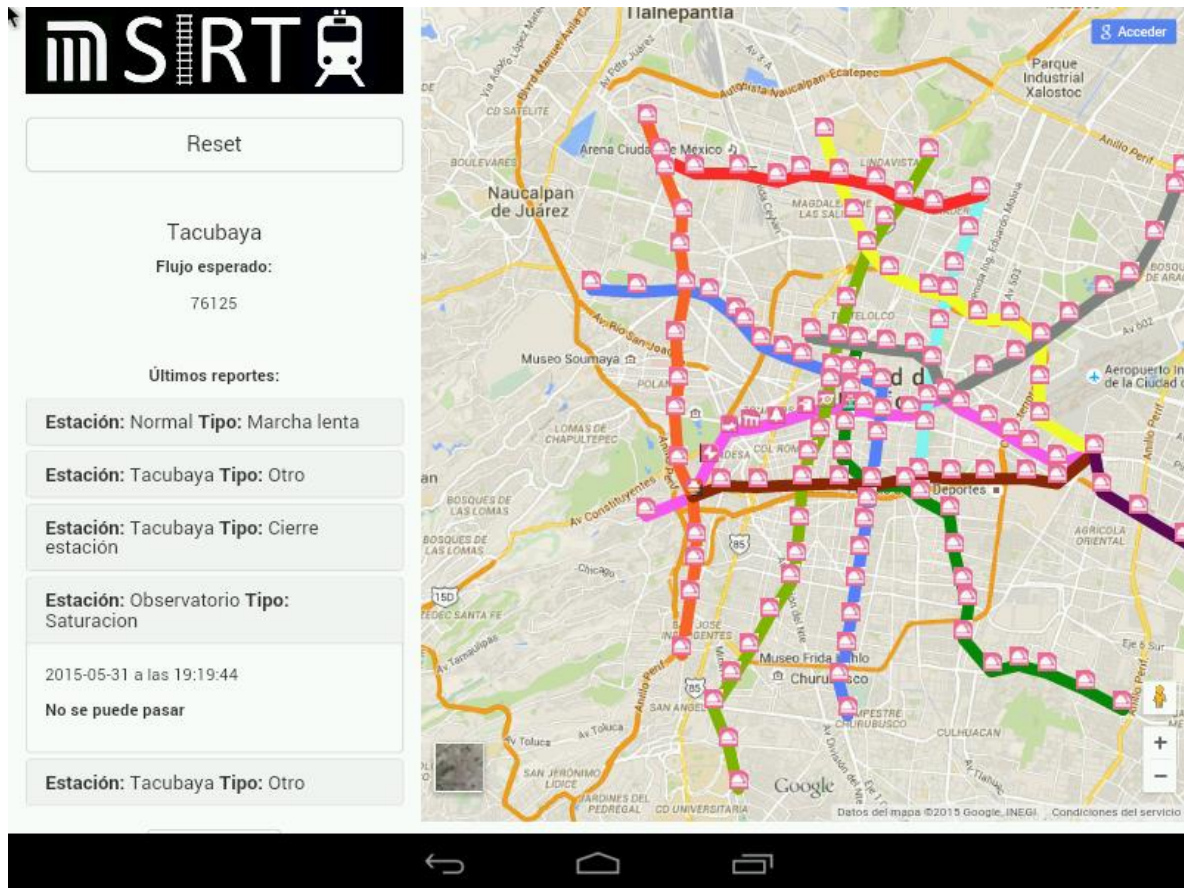


Figura 5-5 Interacción con mapa fuera del flujo

Calcular ruta o rutas más cortas

El algoritmo está diseñado para encontrar la ruta más corta y en ciertos casos, una ruta alterna casi igual de corta, si esta existiese. Para calcularlas basta con que el usuario presione ‘doble tap’ sobre el marcador de la estación de la que quiera partir, el marcador cambiará a color verde:

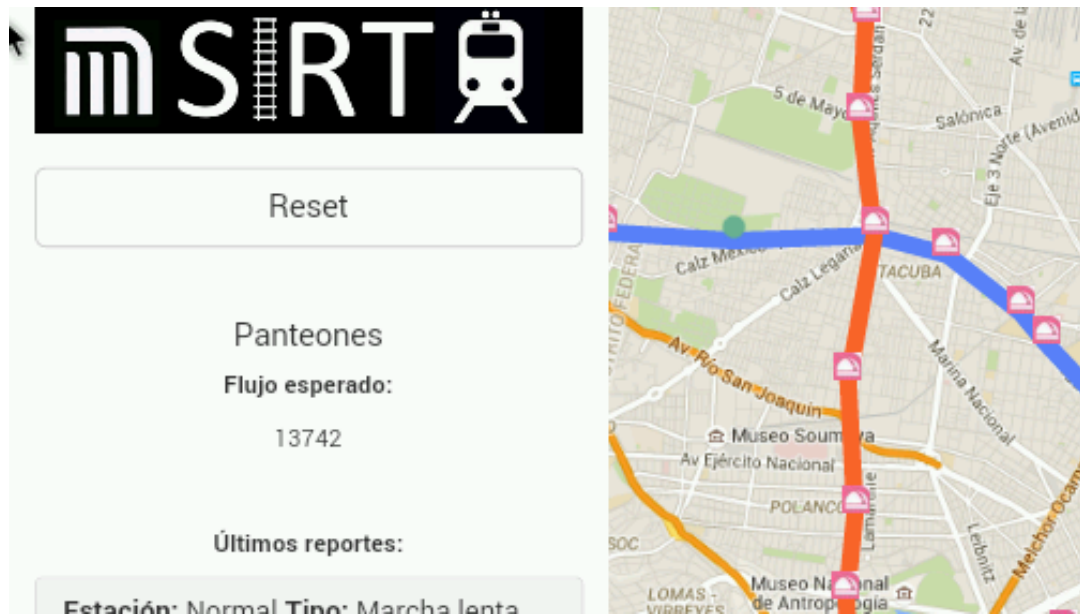


Figura 5-6 Selección de punto inicial

En seguida, si el usuario ya identificó su destino, debe presionar de nuevo con ‘doble tap’ sobre el marcador de la estación a la que quiera llegar, el marcador cambiará a rojo y la App procede a hacer el cálculo de ruta:

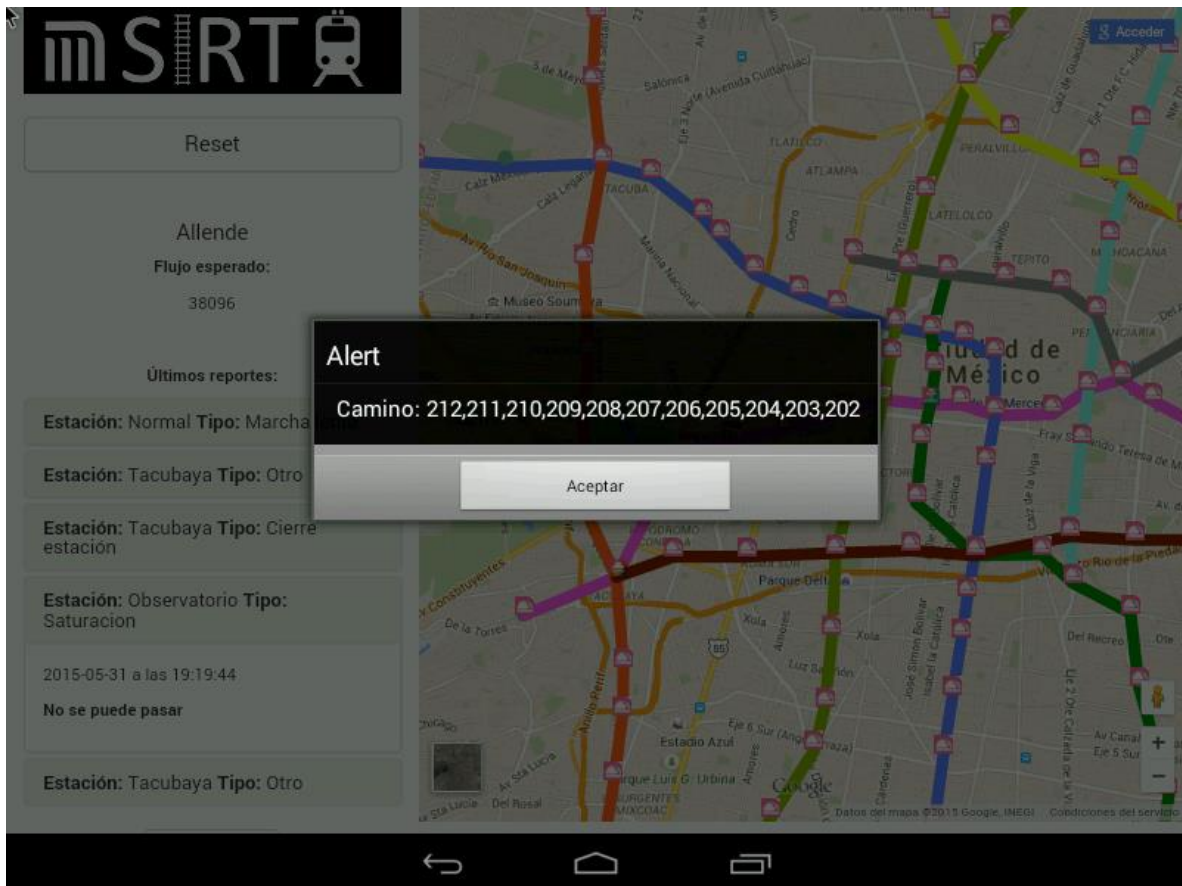


Figura 5-7 Despliegue de id estación

Si sólo se encuentra una ruta, se desplegará un mensaje con información sobre esa ruta; dicha información es: distancia, tiempo y aflujo de personas.

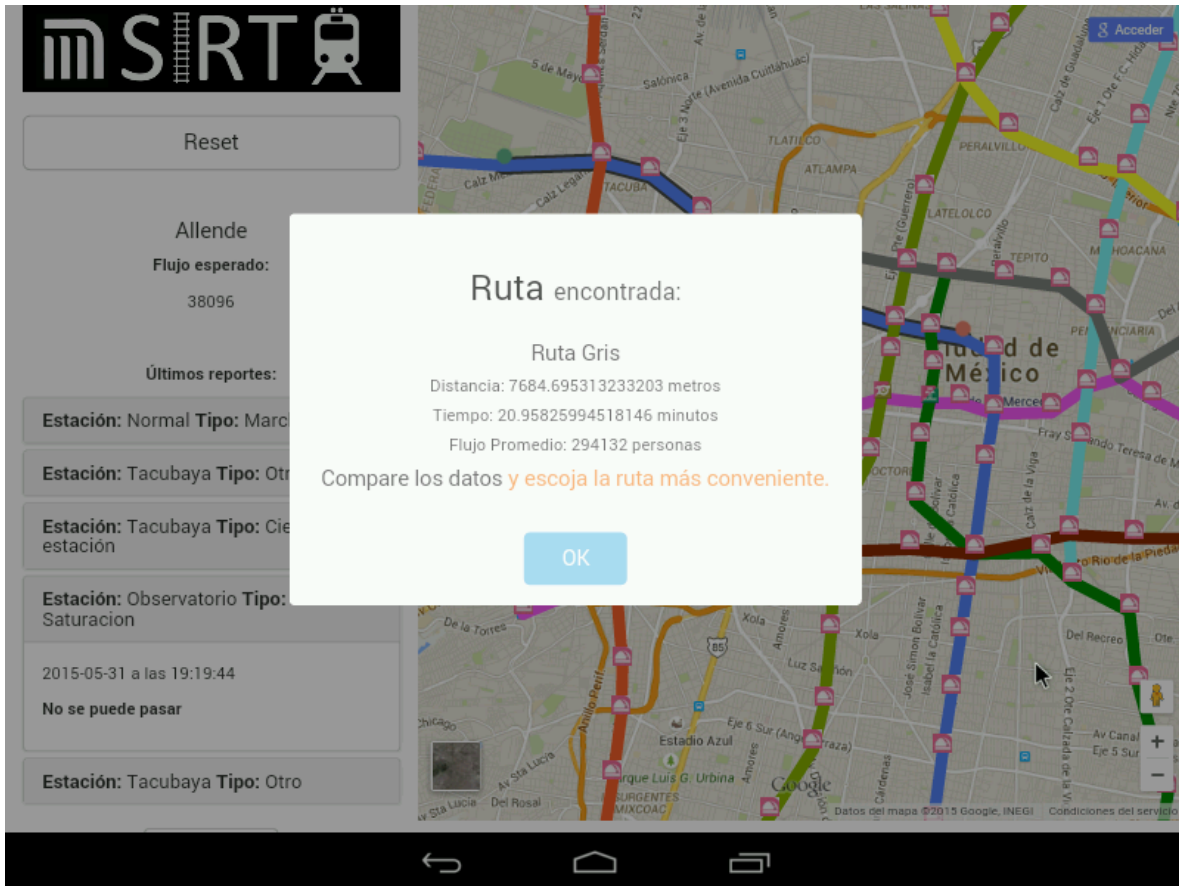


Figura 5-7 Despliegue de id estación

Y se será representada por una poli línea gris:

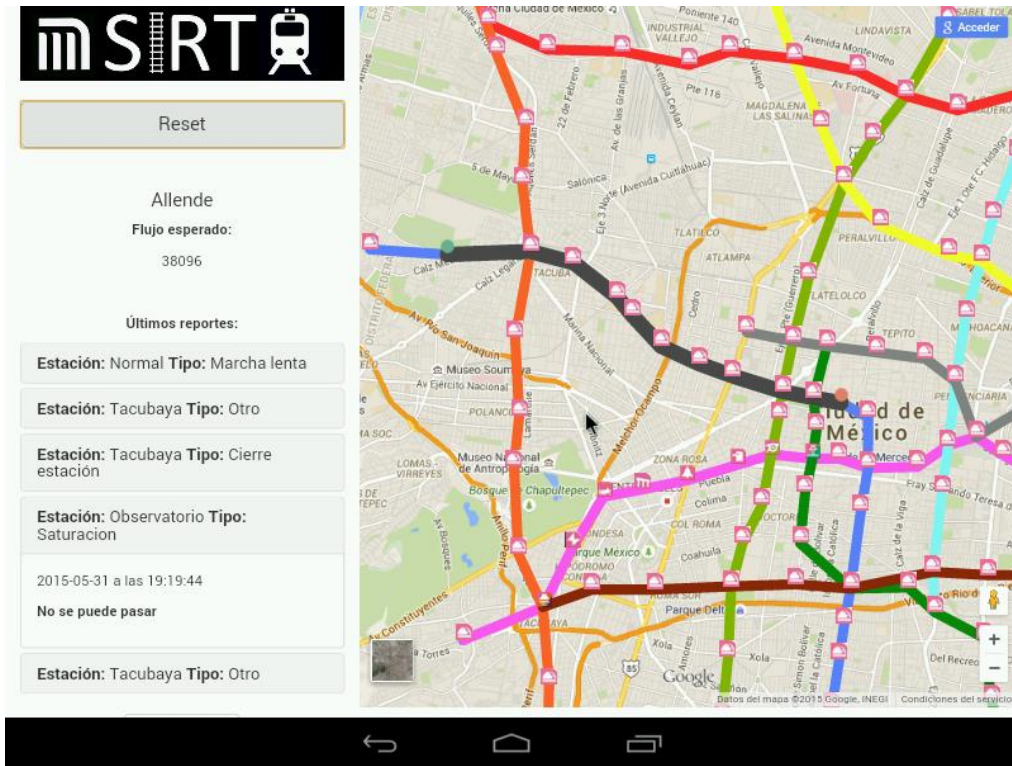
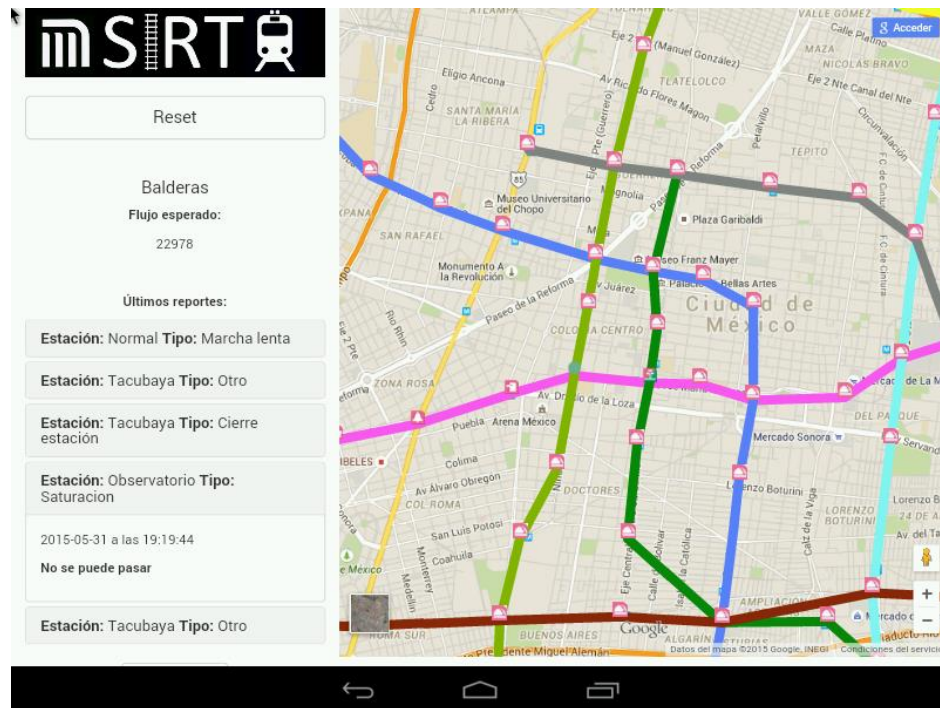


Figura 5-8 Despliegue de ruta óptima

Si el sistema encuentra una ruta alterna, se desplegará una ventana con la información con ambas rutas. Esto para que el usuario compare entre ellas.



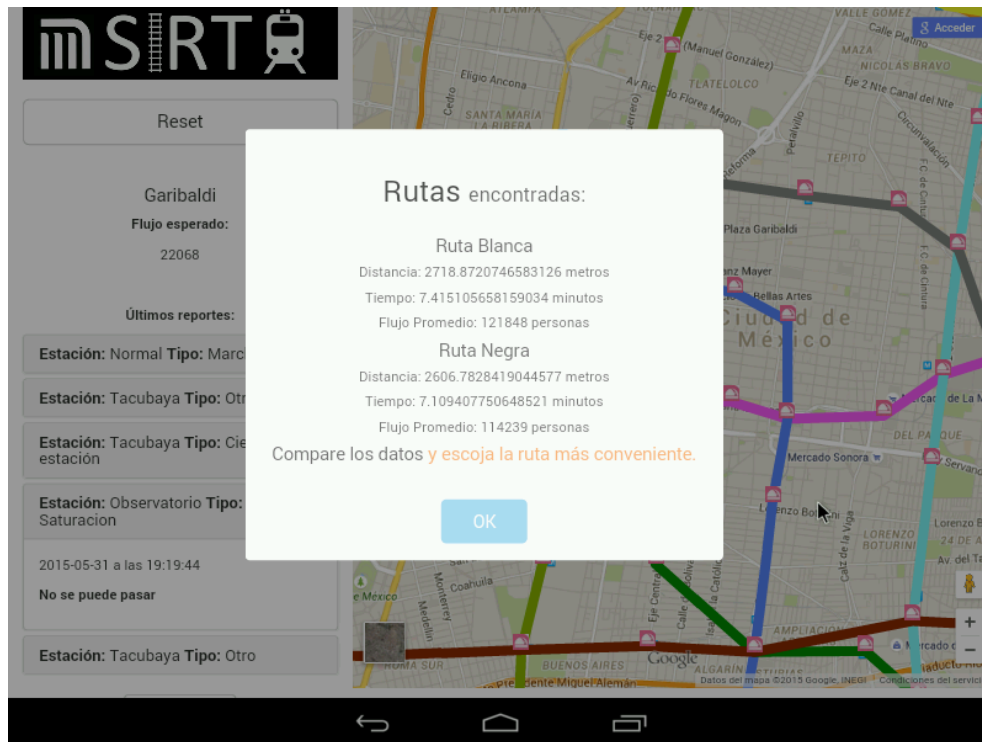
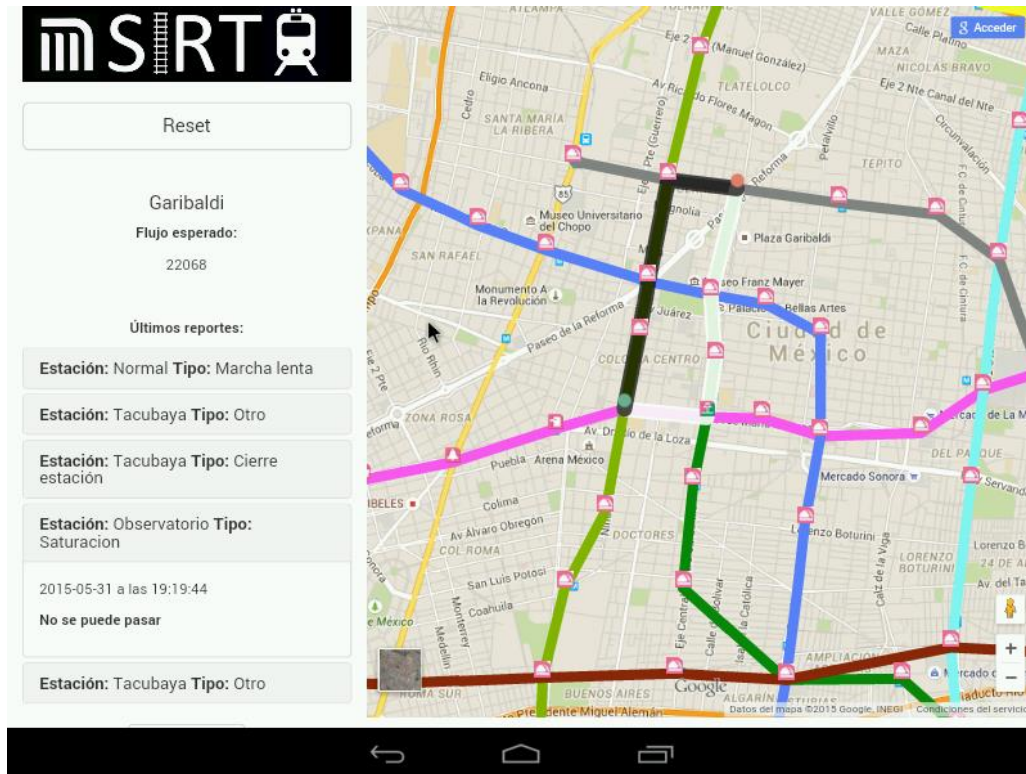
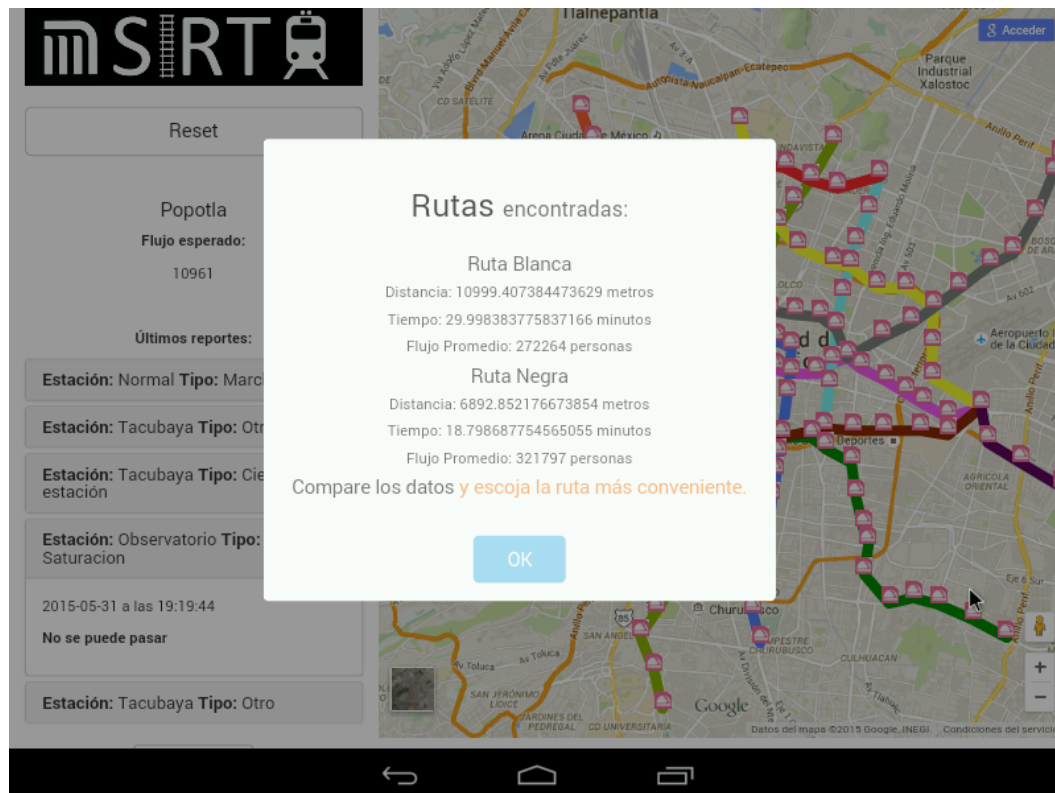


Figura 5-9 Despliegue de información ruta óptima y ruta alterna

Y cada ruta es representada con los colores blanco y negro respectivamente. Si ambas rutas comparten tramos, estos se verán grises.



Mismo caso, con un viaje diferente:



SRT

Reset

Popotla
Flujo esperado:
10961

Últimos reportes:

Estación: Normal Tipo: Marcha lenta

Estación: Tacubaya Tipo: Otro

Estación: Tacubaya Tipo: Cierre estación

Estación: Observatorio Tipo: Saturación

2015-05-31 a las 19:19:44

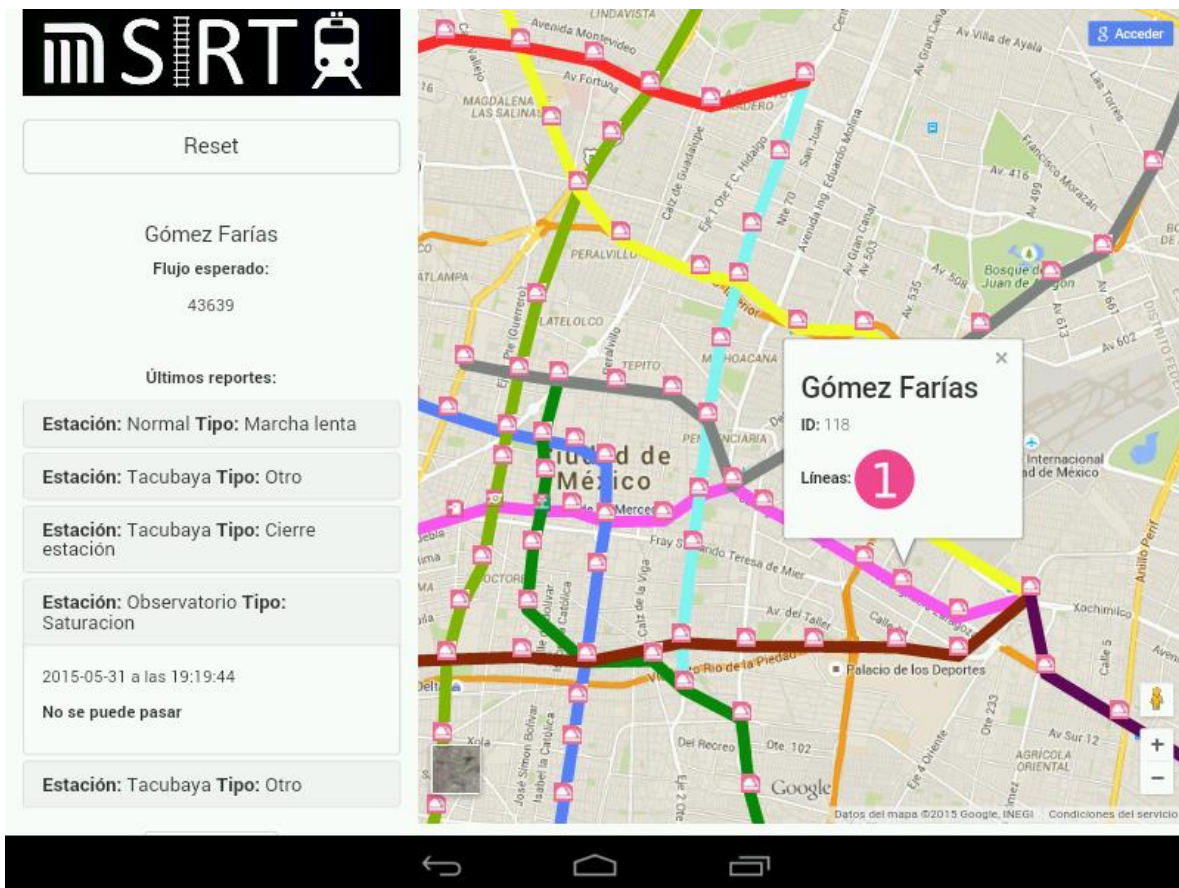
No se puede pasar

Estación: Tacubaya Tipo: Otro

Mapa de Ciudad de México con líneas de transporte y reportes de incidentes.

Levantar reporte de incidencia en estación

Para levantar un reporte en determinada estación, se debe primero seleccionar una de ellas, de lo contrario no se podrá realizar dicho flujo exitosamente:



Damos ‘tap’ en el botón “Crear reporte”, lo cual nos desplegará el siguiente formulario:

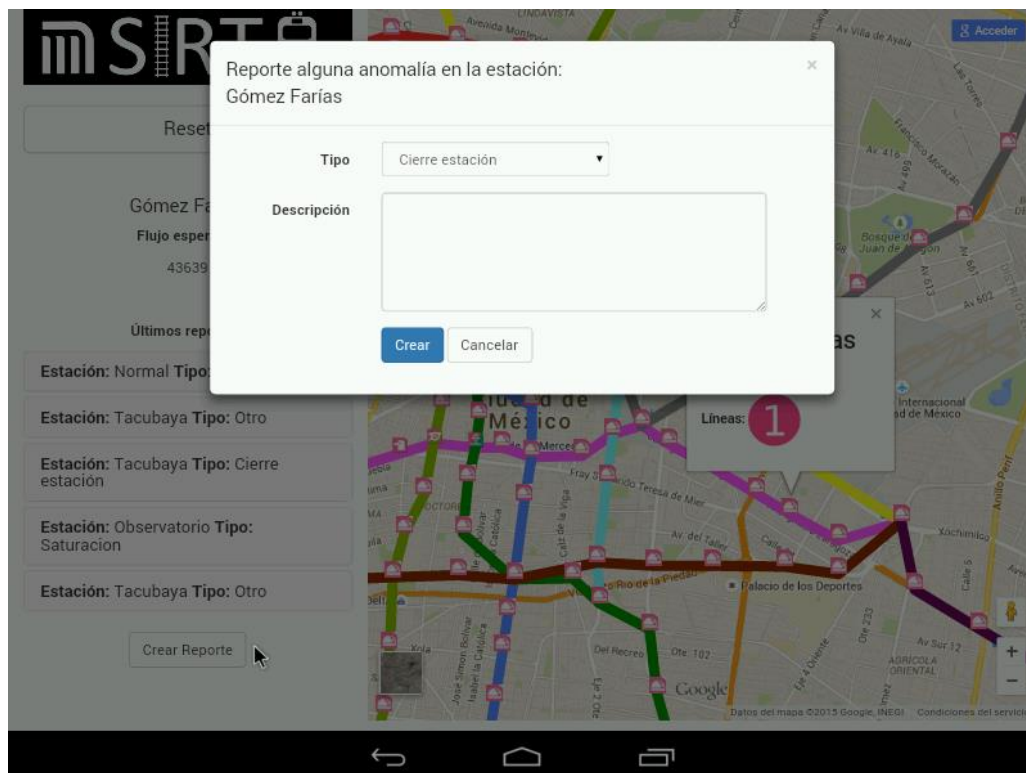
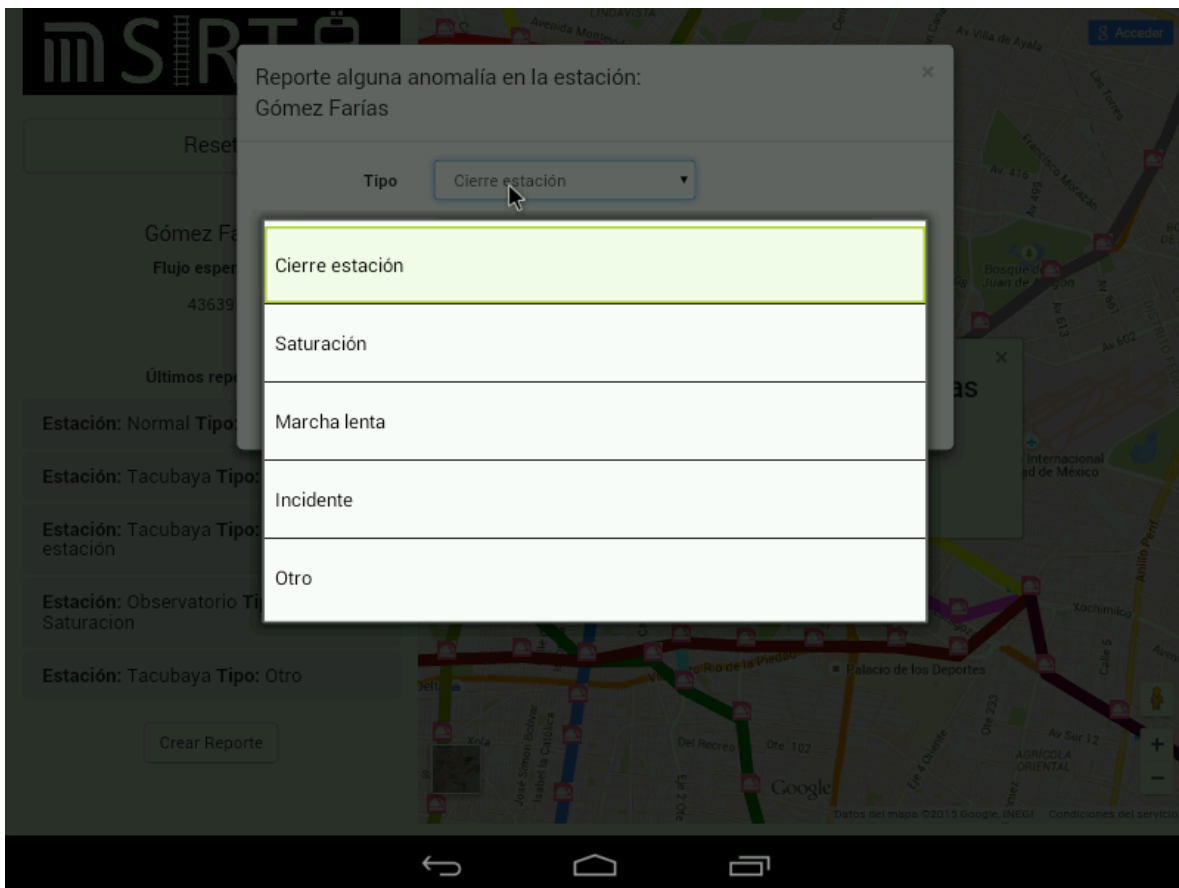
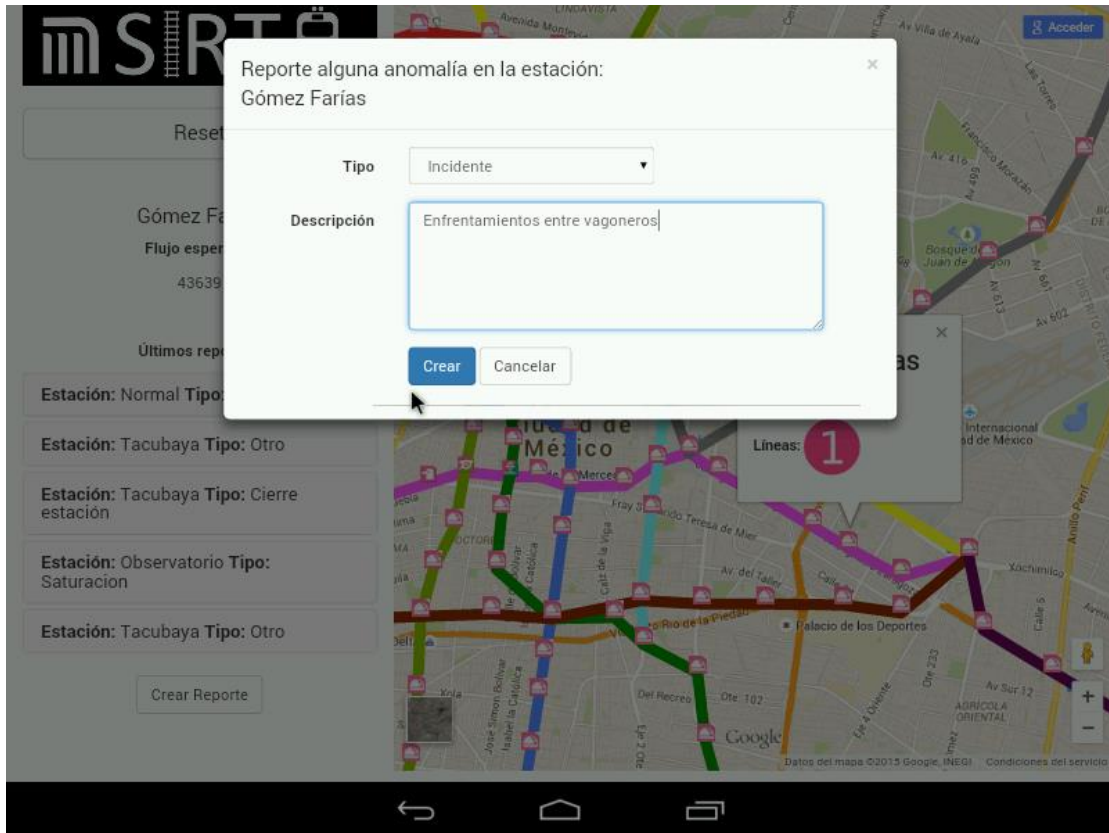


Figura 5-10 Despliegue de formulario de reportes

El usuario debe introducir el tipo de anomalía; el sistema contempla 4 tipos de incidente principales: ‘Cierre estación’, ‘Saturación’, ‘Incidente’ y ‘Otro’, los cuales pueden ser clasificados por sus respectivos campos. También se debe introducir una descripción de la anomalía para describir mejor el acontecimiento anormal en la estación.



Se introducen datos:



Si la conexión e inserción es exitosa, el reporte se debe ver reflejado inmediatamente en nuestro menú de anomalías recientes. Este reporte ahora puede ser visto por todos los demás usuarios que utilicen la App.

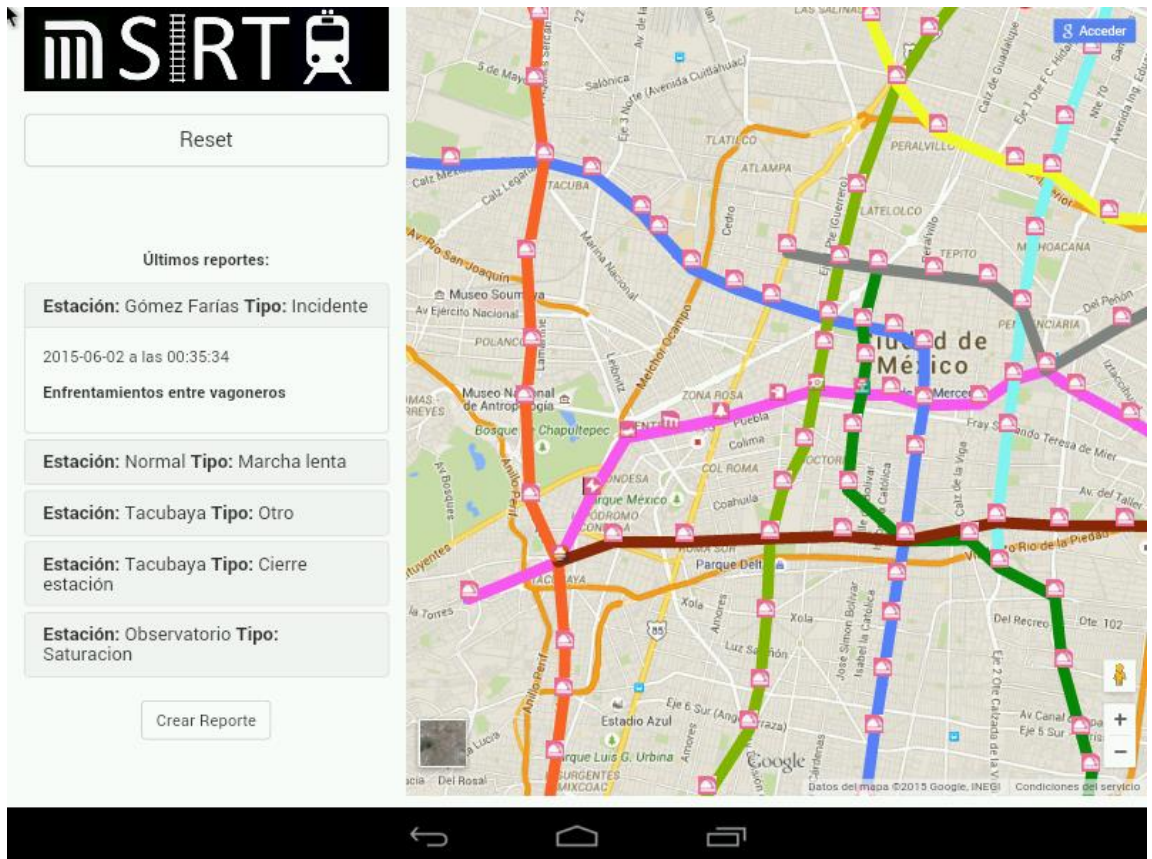


Figura 5-11 Despliegue de reporte recién creado

Como se ha dejado claro, existen más casos de uso, pero sólo se han presentado los más importantes y representativos del sistema.

Conclusiones y Comentarios Finales

En el desarrollo de este prototipo se ha podido observar cómo campos de la computación, en este caso, de la Inteligencia Artificial, muchas veces se malinterpretan, y se piensa que no pueden resolver problemas reales y más que eso, de la vida de un ciudadano promedio. El problema es que anteriormente, la aplicación de dichas ramas no llegaba fácilmente a las manos de un consumidor común, alguien que no fuera un computólogo. Esta situación ha cambiado con la introducción del Smartphone, el cual nos provee de muchas herramientas para facilitar nuestra vida, en este caso, obtener información para planificar un viaje en el STC.

En el caso de los desarrolladores, es importante que exploren las tecnologías actuales, ya que cada vez van penetrando más y más en el mundo y es importante que se enfoquen en desarrollar sistemas que tengan una utilidad más allá del entretenimiento, lo que actualmente predomina en este mercado.

El sistema aún es prototipo, y se necesitaría expandir y optimizar en algunos módulos; sin embargo, es totalmente factible lanzarlo al mercado, ya que en el escenario actual, el software va mejorando con la retroalimentación de sus usuarios. No obstante, el problema principal es la red de telefonía celular del país, aunque dicho impedimento no aplique en todas las estaciones del STC. La aplicación sería gratis, ya que en México no se tiene la cultura de comprar apps, al menos no por ahora; además, el STC está dirigido a la clase popular, por lo que el prototipo debe también estar dirigido para la clase popular.

Se puede concluir que el objetivo del documento se cumple satisfactoriamente, ya que se logra el objetivo planteado de desarrollar un sistema informativo para rutas de transporte sobre un dispositivo móvil. Si bien, el proyecto aún tiene mucho que mejorar, la base está muy bien implementada. Muchas gracias estimado lector, por darse el tiempo de leer este trabajo de investigación, esperando le sea útil de alguna manera.

Trabajo a Futuro

A continuación se exponen algunas propuestas de trabajo a futuro, con los fundamentos que da este documento:

- Mejorar algoritmo de búsqueda : lograr más rutas alternas, optimizar recorridos
- Desarrollar proyecto sobre más plataformas: desarrollar sobre iOS o Windows Phone
- Mejorar gestión de usuarios y reportes: desarrollar una plataforma que gestione información de usuarios que utilicen aplicación de forma masiva
- Implementar algoritmos de predicción a largo plazo: en este caso, sólo se implementaron algoritmos de predicción a corto plazo
- Contemplar más factores que influyen en los recorridos: un ejemplo es la lluvia

Referencias

- [1] SÁNCHEZ MACÍAS. José “ONU: Ciudad de México es la 4ta más grande del mundo”. {En línea}. {10 julio de 2014} disponible en: (www.sandiegored.com/noticias/55165/ONU-Ciudad-de-Mexico-es-la-4ta-mas-grande-del-mundo/).
- [2] “Waze Social GPS & Traffic”. {El línea} disponible en (<https://play.google.com/store/apps/details?id=com.waze>)
- [3] “Metro y Metrobús de México.” {El línea} disponible en <https://play.google.com/store/apps/details?id=com.metromexico>
- [4] “Metro DF {El línea} disponible en <https://play.google.com/store/apps/details?id=com.crayonlion.metro>
- [5] SOMMERVILLE. Ian ”Ingeniería de Software, Madrid, 7a edición, Prentice Hall Ed, 2005, pp 63-75
- [6] JIMÉNEZ HERNÁNDEZ. Eréndira Miriam, ORANTES JIMÉNEZ. Sandra Dinora “Metodologías híbridas para el desarrollo de software en México, una opción factible”. {en línea}. {1 enero de 2012} disponible en (<http://www.revista.unam.mx/vol.13/num1/art16/#up>)
- [7] UNAM, “Ingeniería de transporte” {en línea} disponible en (<http://www.iingen.unam.mx/esmx/Investigacion/Areas/Paginas/Transporte.aspx>)
- [8] Institute of Transportation Engineers, “Transportation engineers ” {en línea } disponible en (ITE <http://www.ite.org/>)
- [9] “Problema de transporte o distribución” {en línea} disponible en: (

<http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigaci%C3%B3n-de-operaciones/problema-del-transporte-o-distribuci%C3%B3n/>

- [10] EXCÉLSIOR {en línea} “línea 12 del metro: el problema es de diseño, no de la obra” {10 enero 2015} disponible en (www.excelsior.com.mx/comunidad/2014/03/31/951245)
- [11] EL UNIVERSAL, “Escenas impactantes en el Metro” {en línea} disponible en (<http://www.eluniversaldf.mx/home/nota17457.html>)
- [12] “Página oficial del metro” {en línea} disponible en (<http://www.metro.df.gob.mx>)
- [13] “¿Qué es el metrobús?” {en línea} disponible en (<http://www.metrobus.df.gob.mx>)
- [14] “Red del tren Ligero” {en línea} disponible en (<http://www.ste.df.gob.mx/servicios/trenligero.html>)
- [15] “Ferrocarriles Suburbanos” {en línea} disponible en (<http://www.fsuburbanos.com/secciones/operacion/recorrido.php>)
- [16] “Sistema de Transporte Colectivo Metrorrey?” {en línea} disponible en (<http://www.movimet.com/2012/12/metrorrey-sistema-de-transporte-metro-en-monterrey/>)
- [17] “SITEUR” {en línea} disponible en (<http://www.siteur.gob.mx/>)
- [18] TORRES. Carmen, “Diseño y análisis de algoritmos”, Parainfo, {1992}, pág. 123
- [19] “Algoritmos y su complejidad” {en línea} disponible en (<http://robotica.uv.es/pub/Libro/PDFs/CAPI7.pdf>)
- [20] PIJOAN. Ander, BORGES. Cruz, PENYA. Yoseba, ALONSO-VICARIO. Ainhoa “Uso d e-mail: o0ox_xo0o@live.com

algoritmos de enrutamiento para el cálculo de indicadores de sostenibilidad”.{en línea}.disponible en: (<http://dugi-doc.udg.edu/bitstream/handle/10256/9041/UsoAlgoritmos-Article.pdf?sequence=2>)

[21] “Tipos de algoritmos de enrutamiento” {en línea} disponible en: (<https://www.tlm.unavarra.es/mod/resource/view.php?id=5872>)

[22] R. Pindyck and D. Rubinfeld, Econometria: Modelos y pronósticos. México D.F.: Mc. Graw Hill, 2001.

[23] D. Larose, Discovering Knowledge in Data. New Jersey, EEUU: Wiley, 2005.

[24] “Afluencia de pasajeros estación por estación” {en línea} disponible en (<http://www.metro.df.gob.mx/operacion2/afluencia.html>)

[25] “Android SDT ” {en línea} disponible en (developer.android.com/sdk/)

[26] “Apache Tomcat” {en línea} disponible en (<http://tomcat.apache.org/>)

[27] “Google Developers: Google Maps API” {en línea} disponible en (<https://developers.google.com/maps/?hl=es>)