



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS” ZACATENCO

**“DESARROLLO DE UNA APLICACIÓN WEB PARA EL USO
Y DIFUSIÓN DE LOS DATOS CONTENIDOS EN EL
ESTÁNDAR GTFS ESTÁTICO DE LOS DISTINTOS
TRANSPORTES PÚBLICOS DE LA CIUDAD DE MÉXICO”**

PROYECTO DE INVESTIGACIÓN

PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMUNICACIONES Y ELECTRÓNICA

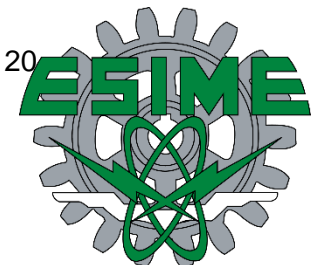
PRESENTAN:

**JOSE PABLO BUENDIA VALVERDE
PAOLA ESPINOSA OREGEL
RODRIGO RUIZ MEDINA**

ASESORES:

**ING. MIGUEL SÁNCHEZ MERAZ
ING. MARTHA CECILIA GALAZ LARIOS**

CIUDAD DE MÉXICO, SEPTIEMBRE 2022



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”

TEMA DE TESIS

QUE PARA OBTENER EL TÍTULO DE INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
POR LA OPCIÓN DE TITULACIÓN PROYECTO DE INVESTIGACION SIP 20210272
DEBERA (N) DESARROLLAR C. JOSE PABLO BUENDIA VALVERDE
C. PAOLA ESPINOSA OREGEL
C. RODRIGO RUIZ MEDINA

**“DESARROLLO DE UNA APLICACIÓN WEB PARA EL USO Y DIFUSIÓN DE LOS DATOS
CONTENIDOS EN EL ESTÁNDAR GTFS ESTÁTICO DE LOS DISTINTOS TRANSPORTES PÚBLICOS DE
LA CIUDAD DE MÉXICO”**

DESARROLLAR UNA APLICACIÓN WEB QUE CONTRIBUYA A MEJORAR LA EFICIENCIA DEL USO DEL
TRANSPORTE PÚBLICO DE LA CIUDAD DE MÉXICO OFRECIENDO INFORMACIÓN VERAZ Y OPORTUNA A
LOS USUARIOS PARA DESPLAZARSE A SU DESTINO

- ❖ FUNDAMENTOS TEÓRICOS
- ❖ MÉTODOS PARA EL DESRROLLO DE LA APLICACIÓN WEB.
- ❖ METODOLOGÍA

CIUDAD DE MÉXICO, A 11 DE SEPTIEMBRE DE 2022.

ASESORES



ING. MIGUEL SÁNCHEZ MERAZ



ING. MARTHA CECILIA GALAZ LARIOS



M. EN C. ITZALÁ RABADÁN MALDA
JEFA DE LA CARRERA DE INGENIERÍA
EN COMUNICACIONES Y ELECTRÓNICA

Instituto Politécnico Nacional

Presente

Bajo protesta de decir verdad los que suscriben **JOSE PABLO BUENDIA VALVERDE, PAOLA ESPINOSA OREGEL Y RODRIGO RUIZ MEDINA**, manifiestan ser autores y titulares de los derechos morales y patrimoniales de la obra titulada "**DESARROLLO DE UNA APLICACIÓN WEB PARA EL USO Y DIFUSIÓN DE LOS DATOS CONTENIDOS EN EL ESTÁNDAR GTFS ESTÁTICO DE LOS DISTINTOS TRANSPORTES PÚBLICOS DE LA CIUDAD DE MÉXICO**", en adelante "**La Tesis**" y de la cual se adjunta copia, en un impreso y un cd, por lo que por medio del presente y con fundamento en el artículo 27 fracción II, inciso b) de la Ley Federal del Derecho de Autor, otorgamos al **INSTITUTO POLITÉCNICO NACIONAL**, en adelante **EI IPN**, autorización no exclusiva para comunicar y exhibir públicamente total o parcialmente en medios digitales o en cualquier otro medio; **para apoyar futuros trabajos relacionados con el tema de "La Tesis"** por un periodo de **5 años** contado a partir de la fecha de la presente autorización, dicho periodo se renovará automáticamente en caso de no dar aviso expreso a **EI IPN** de su terminación.

En virtud de lo anterior, **EI IPN** deberá reconocer en todo momento nuestra calidad de autores de "**La Tesis**".

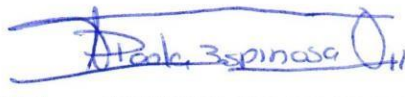
Adicionalmente, y en nuestra calidad de autores y titulares de los derechos morales y patrimoniales de "**La Tesis**", manifestamos que la misma es original y que la presente autorización no contraviene ninguna otorgada por los suscritos respecto de "**La Tesis**", por lo que deslindamos de toda responsabilidad a **EI IPN** en caso de que el contenido de "**La Tesis**" o la autorización concedida afecte o viole derechos autorales, industriales, secretos industriales, convenios o contratos de confidencialidad o en general cualquier derecho de propiedad intelectual de terceros y asumimos las consecuencias legales y económicas de cualquier demanda o reclamación que puedan derivarse del caso.

Ciudad de México., a 6 de diciembre de 2022.

Atentamente



BUENDIA VALVERDE JOSE PABLO



ESPINOSA OREGEL PAOLA



RUIZ MEDINA RODRIGO

ÍNDICE

RESUMEN	6
1. INTRODUCCIÓN	7
1.1 DEFINICIÓN DEL PROBLEMA	8
1.2 OBJETIVO GENERAL	9
1.2.1 <i>Objetivos Específicos</i>	9
1.3 JUSTIFICACIÓN	10
1.4 HIPÓTESIS	11
1.5 ALCANCE.....	12
1.6 ESTADO DEL ARTE	13
2. CAPÍTULO 1. FUNDAMENTOS TEÓRICOS	16
2.1 ESPECIFICACIÓN DE FEED DE TRANSITO GENERAL (GTFS).....	17
2.1.1 <i>Antecedentes de GTFS</i>	17
2.1.2 <i>Definición GTFS</i>	17
2.1.3 <i>GTFS Estático</i>	18
2.1.4 <i>GTFS Dinámico</i>	19
2.1.5 <i>Archivos de conjuntos de datos para GTFS estático</i>	19
2.1.6 <i>Requisitos de archivo</i>	18
2.1.7 <i>Secuencia recomendada para la creación de archivos</i>	20
2.2 RECURSOS PARA LA CONSULTA Y USO DE LOS ARCHIVOS GTFS	21
2.2.1 <i>Descripción general del Software</i>	21
2.2.2 <i>Librerías</i>	24
2.2.3 <i>OpenStreetMap</i>	26
2.2.4 <i>Algoritmo de Dijkstra</i>	27
2.3 HERRAMIENTAS PARA EL DISEÑO DE LA APLICACIÓN WEB	28
2.3.1 <i>HTML</i>	28
2.3.2 <i>Framework Flask</i>	29
2.4 CIUDADES INTELIGENTES	30
2.4.1 <i>Cambios imprescindibles en la sociedad</i>	31
2.5 DATOS ABIERTOS.....	32
2.5.1 <i>Normativa de datos abiertos</i>	32
3. CAPÍTULO 2. MÉTODOS PARA EL DESARROLLO DE LA APLICACIÓN WEB	34
3.1 PROCEDIMIENTOS Y TÉCNICAS UTILIZADAS.....	35
3.1.1 <i>Consulta y depuración de archivos GTFS</i>	35
3.1.2 <i>Funcionamiento del Algoritmo</i>	41
4. CAPÍTULO 3. METODOLOGÍA	45
4.1 BACK-END	46
4.2 FRONT END	50
4.2.1 <i>Interfaz grafica</i>	55
4.3 VISUALIZACIÓN DE LA INTERFAZ DE LA APLICACIÓN WEB	60
4.4 ANÁLISIS DE RESULTADOS	61
5.	
6. 1. <i>Archivos Obligatorios del estándar GTFS estático</i>	62
6.2. <i>Archivos opcionales del estándar GTFS estático</i>	70
7. REFERENCIAS ELECTRÓNICAS.....	72

ÍNDICE DE IMÁGENES Y FIGURAS

<i>Imagen 1 Representación gráfica de un array</i>	24
<i>Imagen 2 Contenido del archivo GTFS. ZIP</i>	35
<i>Imagen 3 Conversión de los archivos .txt a .xlsx</i>	36
<i>Imagen 4 Primeras dos columnas de datos del archivo agency.xlsx</i>	36
<i>Imagen 5 Muestra de las últimas dos columnas del archivo agency.xlsx</i>	37
<i>Imagen 6 Datos contenidos en el archivo routes.xlsx que muestran los resultados de la agencia “METRO”</i>	37
<i>Imagen 7 Datos contenidos en el archivo trips.xlsx</i>	38
<i>Imagen 8 Datos contenidos en el archivo stop_times.xlsx</i>	39
<i>Imagen 9 Datos contenidos en el archivo stop.xlsx</i>	40
<i>Imagen 10 Código escrito para realizar Algoritmo de Dijkstra</i>	47
<i>Imagen 11 Líneas de código para la generación del mapa</i>	48
<i>Imagen 12 Mapa generado en OpenStreetMap</i>	48
<i>Imagen 13 Asignación de pesos de las entradas numéricas</i>	49
<i>Imagen 14 Programación realizada con HTML para insertar origen, destino y hora del viaje</i>	51
<i>Imagen 15 Estructura de Interfaz de Usuario con HTML antes de usar CSS</i>	52
<i>Imagen 16 Estructura de Interfaz de Usuario usando CSS</i>	53
<i>Imagen 17 Descripción del funcionamiento general de la aplicación web</i>	55
<i>Imagen 18 Pantalla principal al ingresar a la aplicación web</i>	55
<i>Imagen 19 Campos para ingresar Origen y Destino</i>	56
<i>Imagen 20 Ejemplo de llenado de los campos para el cálculo de ruta</i>	56
<i>Imagen 21 Resultado que arroja con la Ruta 1 basada en la distancia</i>	57
<i>Imagen 22 Resultado que arroja con la Ruta 2 basada en el tiempo</i>	58
<i>Imagen 23 Sección que proporciona los mapas de las rutas propuestas</i>	59
<i>Imagen 24 Mapa resultante de la Ruta 1: Pantitlán-San Juan de Letrán</i>	59
<i>Figura 1 Diagrama a bloques de la composición del modelo de archivos GTFS</i>	18
<i>Figura 2 Diagrama de Flujo del funcionamiento general del algoritmo</i>	42
<i>Figura 3 Diagrama de Flujo del funcionamiento general de la Inteligencia Artificial</i>	43
<i>Figura 4 Diagrama de Flujo del funcionamiento general del Algoritmo de Dijkstra</i>	44

RESUMEN

El presente trabajo contiene el planteamiento de la problemática que se presenta al viajar en el transporte público de la Ciudad de México la cual, en parte, se debe a la falta de herramientas tecnológicas que permitan mejorar la planeación de los viajes, y facilitar el uso de la red de transportes públicos.

Por lo tanto, se propone el desarrollo y creación de una aplicación web que permita la visualización de las rutas disponibles del transporte público, para planear un viaje o conocer la ruta si no has viajado antes en el transporte público de la CDMX, se piensa que la planeación anticipada, ayudará en la movilidad de los usuarios.

Para automatizar la predicción de la ruta se realizan una serie de pasos para calcular la distancia mínima desde la estación de partida hasta el destino y ofrecer una solución a los usuarios de cuál ruta es más conveniente para trasladarse.

La obtención de los datos relevantes, acerca del transporte público, se consultó en el portal oficial de datos abiertos de la Ciudad de México. La información recopilada, representa un pilar fundamental para el funcionamiento del proyecto, así como una garantía de que los datos utilizados en este proyecto son fidedignos.

La idea de obtener la lista de estaciones que se recorrerán durante el viaje se complementa con la generación de un mapa gráfico de cada propuesta de viaje. Además, se calcula un porcentaje de satisfacción basado en diferentes parámetros, el cual te indica cuál de las dos rutas proporciona menos incomodidades para llegar al destino. Se considera que estas herramientas facilitan la búsqueda, uso y aprendizaje de una ruta en el transporte público.

Finalmente, se trabaja en la forma de presentar al usuario un entorno agradable a la vista, sencillo y funcional donde solo se ingresa su origen y destino para obtener dos propuestas para su viaje, así como sus respectivos mapas y un porcentaje de satisfacción de cada ruta, en cualquier navegador web.

1. INTRODUCCIÓN

El avance significativo en las tecnologías que usamos exige que todo nuestro entorno se actualice para solventar las nuevas necesidades que se presentan día a día. Por lo cual se debe echar un vistazo a nuestro entorno y conocer las necesidades para crear diferentes soluciones, en el caso de este proyecto, se hizo un enfoque al problema que se tiene en nuestra Ciudad respecto a los servicios de transporte público, ya que algunos están desactualizados y no se tiene una buena experiencia de uso.

Cabe señalar que se analizaron previamente algunos de los elementos que influyen en la elección de una ruta, esto con la finalidad de crear un método que permita proponer alternativas al usuario y aportar una herramienta de planeación aplicable al transporte público en la ciudad. Se presenta entonces el desarrollo de una aplicación web para planear con anticipación una ruta que facilite a los usuarios realizar su viaje.

Por lo tanto, al final, se evalúa si la solución creada logra cumplir con el objetivo de obtener una nueva herramienta en favor de los usuarios del transporte público de la Ciudad de México.

1.1 Definición del problema

Actualmente en la Ciudad de México los servicios de transporte público están sufriendo diversas complicaciones, principalmente han provocado molestias entre los usuarios debido a la saturación de personas que viajan diario, por ejemplo, el STC Metro en el año 2019, según datos oficiales, se transportan aproximadamente 3 millones de personas al día, siendo éste el transporte más utilizado.

Otro de los inconvenientes de los transportes es que los que llevan más años en operación tienen ya un atraso importante en la tecnología con la que cuentan, por lo que no hay muchos recursos que ayuden o faciliten a los usuarios para realizar su viaje sin mayor complicación.

Hoy en día, en la red de movilidad integrada de la Ciudad de México podemos ver 6 distintos transportes:

- Sistema de Transporte Colectivo Metro
- Metrobús
- Servicio de Transporte Eléctrico (Trolebús / Tren Ligero)
- Red de Transporte de Pasajeros (RTP)
- Mexibús
- Tren suburbano

Sin embargo, no todas las personas exploran las diferentes rutas que cada uno ofrece, ni tampoco como se conectan entre los distintos transportes. Resulta entonces que elegir una ruta, se vuelve algo complejo, incluso para muchos usuarios elegir un determinado transporte es una decisión que puede afectar su viaje y retrasar o hasta impedirle la llegada a su destino.

La problemática se compone principalmente de estos tres aspectos, falta de innovaciones tecnológicas, el flujo excesivo de pasajeros y la falta de conocimiento de la red de transportes, por lo que viajar en transporte público se vuelve un caos, es estresante, nada agradable y en general, hace que el concepto que todas las personas tiene de dichos servicios sea completamente negativo.

1.2 Objetivo General

Desarrollar una aplicación web que contribuya a mejorar la eficiencia del uso del transporte público de la Ciudad de México ofreciendo información veraz y oportuna a los usuarios para desplazarse a su destino.

1.2.1 Objetivos Específicos

- Difundir información veraz acerca del transporte público STC Metro de la Ciudad de México por medio de una aplicación web.
- Consultar y hacer uso de la información contenida en los archivos GTFS (General Transit Feed Specification).
- Desarrollar una aplicación web para ayudar a los usuarios del transporte público STC Metro de la Ciudad de México a planear su viaje de manera anticipada y sencilla.
- Facilitar el uso del transporte público STC Metro de la Ciudad de México a personas que no estén familiarizadas con el servicio, tal como visitantes de otros estados y/o extranjeros.

1.3 Justificación

Los diferentes transportes públicos en México y en algunas otras grandes ciudades del mundo han sido poco eficientes, están saturados y representan una mala experiencia para el usuario.

En estos últimos años las ciudades siguen presentando un aumento en su población debido a que las personas que viven en zonas rurales migran para tener mejores oportunidades y servicios, por lo que es necesario dar solución a este problema e implementar herramientas que contribuyan y actualicen la forma de transportarse en las grandes ciudades.

La ITU (Unión Internacional de Telecomunicaciones) se dio a la tarea de definir el concepto de ciudad inteligente:

“Una ciudad inteligente y sostenible es una ciudad innovadora que utiliza las tecnologías de la información y la comunicación (TIC) y otros medios para mejorar la calidad de vida, la eficiencia de la operación y los servicios urbanos y la competitividad, al tiempo que garantiza que satisfice las necesidades de las generaciones presentes y futuras con respeto a los aspectos económicos, sociales, medioambientales y culturales”. [1]

Debido a este concepto podemos percatarnos de la importancia que representa dar solución al problema de la movilidad con innovaciones en beneficio de la comunidad, presente y futura, en los distintos transportes públicos.

Con ayuda de las TIC se tendrá la capacidad de intercambiar, producir, almacenar, procesar y transferir datos del transporte público a través de una red confiable y así contribuir con un sistema de información que cambie la percepción de lo que hoy representa usar este servicio.

Por último cabe mencionar que, como parte de la realización de este proyecto, es necesario apoyarse de diversas tecnologías que faciliten el desarrollo del mismo, en este caso, para obtener los datos de los transportes públicos oficiales, se utilizará el estándar GTFS que formará una parte fundamental del sistema propuesto y que se encuentra ligado a la geolocalización la cual por medio del GPS, basado en la comunicación con satélites, señales de microondas y demás elementos propios de las telecomunicaciones, ayudará a trazar las rutas y mostrar las ubicaciones de los diferentes transportes públicos de la Ciudad.

1.4 Hipótesis

Una aplicación web que brinde datos de la red de transporte público de la Ciudad de México mejorará la experiencia de los usuarios al realizar un viaje. El presente proyecto hará más sencillo el traslado debido a la disponibilidad de información oficial de los diferentes transportes, se presentará siempre actualizada y podrá ser consultada por cualquier usuario. En esta aplicación se mostrarán las rutas posibles para llegar sin mayor complicación a cualquier destino, esto sin necesidad de tener conocimiento previo de la red de transporte público. Debido a esto ayudará a la movilidad de las personas que diario utilizan estos servicios, así como a las personas que no residen en esta ciudad o no tienen conocimientos suficientes pues actuará como una herramienta fundamental para cambiar la forma de usar el transporte público.

1.5 Alcance

Debido a la investigación que se hizo respecto a los diferentes transportes y considerando el tiempo determinado para la culminación del proyecto, se toma únicamente el Sistema de Transporte Colectivo Metro como modelo para el desarrollo de la aplicación web. Debido a que es el más utilizado y el que transporta el mayor número de personas en la Ciudad de México, es una buena opción para probar el funcionamiento de la aplicación web que se tendrá como resultado y obtener el análisis y las conclusiones correspondientes. Además, este transporte es de los que necesita más herramientas para modernizar la forma en que sus usuarios la utilizan.

Para la obtención de los datos no se tuvo ningún problema ya que los datos se obtuvieron de la página oficial del portal de datos abiertos, ahí encontramos disponible la información de todos los transportes públicos, sin embargo, al sólo necesitar los datos específicos del STC Metro, se tuvo que hacer un filtrado de datos con los campos requeridos, lo que nos llevó a generar archivos propios limitando la información contenida.

Por último, la aplicación web resultante cuenta con un diseño simple pero que representa las bases de un proyecto escalable. Está hecha en Python, un software gratuito que facilitó el rápido desarrollo de esta aplicación ya que cuenta con muchas herramientas de diseño y no fue difícil aprender su lenguaje o utilizar sus herramientas. Entonces, la aplicación se diseñó únicamente para ser consultada en un navegador web y para brindar los datos de las rutas recomendadas con sus respectivos mapas.

1.6 Estado del arte

BusChecker

BusChecker es una aplicación utilizada en el Reino Unido (UK) que se centraliza principalmente en el transporte por Autobuses, permite conocer la llegada de autobuses a las estaciones de manera instantánea y actualizada para cada autobús. Esta aplicación permite observar las salidas de cada autobús para cada "parada" en todo el Reino Unido, con esto se permite tener una planificación de un viaje más sencillo y mejor planeado. Incluso en ciertas zonas la aplicación permite realizar la compra de boletos desde el teléfono celular, lo cual ayuda con ahorro de tiempo en traslados e incluso en seguridad para el manejo de dinero en efectivo.

"Lleve en el bolsillo los horarios de los autobuses en tiempo real. BusChecker brinda información precisa, confiable y en directo a lo largo del recorrido del autobús" [2]

Además de trabajar en el reino unido se está probando su uso en ciudades como es New York City que es una de las ciudades más grandes y transitadas de EUA. Dentro del sistema de navegación cuenta con GPS desarrollado por Google que nos permite ver en tiempo real la ruta y el estado de ella y cuando estas fuera del Bus te indica la parada de autobús más cercana y el tiempo promedio en que tardarías en llegar ahí para después iniciar con la ruta más apropiada hasta el destino.

Enrútrate

Enrútrate es una aplicación desarrollada por María Georgina Patrón Camacho, egresada del Tecnológico de Monterrey de la carrera de Ingeniería Industrial y de Sistemas. Su idea nació cuando tuvo la oportunidad de realizar un viaje a Francia donde se percató que su sistema de transporte está muy bien organizado, podía consultar mapas, la página web oficial y diversas aplicaciones que facilitaban el uso de los diferentes servicios de transporte. Cuando regresó a México, decidió iniciar un proyecto similar y su punto de partida fue Culiacán. Los primeros mapas se difundieron de forma física en una revista, ahí se plasmaban las diferentes rutas y lugares de interés, sin embargo, Georgina Patrón siempre pensó en llevar su proyecto a otro escenario más atractivo y migrar a lo digital pero no estaba capacitada para lograrlo por si sola, entonces, decidió presentar su idea ante EXATEC, una comunidad de exalumnos del Tecnológico de Monterrey, y se logró asociar con una empresa dedicada al desarrollo de software para comenzar con la creación de la aplicación.

"Al principio fue difícil conseguir la información y encontrar las rutas de los camiones, debido a la falta de actualización de ellas" [3]

Cabe mencionar que este proyecto comenzó en el año 2013, en ese entonces fue difícil encontrar información de las rutas por lo que se decidió basarse en los mapas existentes o preguntar a las personas que diario tomaban el transporte público para obtener la información correcta, incluso cuando se lanzó la aplicación también recopilaron algunos comentarios de los mismos usuarios.

El servicio que ofrecía la versión inicial sólo consistía en tener un catálogo de rutas que los usuarios podían consultar, posteriormente ya podías ver los distintos transbordos, hoy en día la aplicación te ofrece la ubicación en tiempo real del camión para que sepas cuando llegará y un análisis de información para los transportistas donde puedan conocer las tendencias de sus usuarios, por ejemplo, las rutas más utilizadas, horarios concurridos, destinos más buscados, etc.,

Actualmente, gracias a la implementación de los servicios de GPS en el desarrollo de la aplicación y con ayuda de la Coordinación de Rutas y Medios de Transporte, se ha logrado implementar en 9 ciudades del norte de México. Así, Georgina Patrón espera seguir cumpliendo el objetivo que se planteó desde un inicio, mejorar la experiencia al viajar en transporte público.

Moovit

Moovit es una aplicación de servicios de mapeo y transporte público desarrollado por una compañía de origen Israel llamada Moovit Inc. Esta aplicación incluye diversas funciones entre las que se encuentran, planificación de viajes, horarios que te permiten conocer la hora de salida y llegada de los transportes en tiempo real, mapas de paradas, itinerarios actualizados, estaciones locales, alertas de servicios, así como diversos avisos que podrían influir en algún viaje.

“Nuestro objetivo es dar a las ciudades, operadores y ciudadanos la posibilidad de disfrutar de una mejor calidad de vida eliminando las limitaciones y barreras de la movilidad urbana”.[4]

En la actualidad está disponible para dispositivos móviles Android, IOS, Windows Phone, además de la versión web. Esta aplicación se encuentra entre las más utilizadas en el mundo con un poco más de 140 millones de usuarios. Se encuentra presente en 3,400 ciudades, en 112 países y 45 idiomas. Proporciona información relevante de diferentes transportes, los cuales incluyen autobuses, metros, trenes, tranvías, trolebuses, entre otros. Cualquier usuario puede acceder a un mapa en tiempo real, visualizar estaciones, paradas cercanas que son referenciadas a partir de la ubicación GPS actual, así mismo la planificación de viajes a través de diferentes medios de transporte basados en información en tiempo real.

Moovit difiere de las demás aplicaciones tradicionales de este tipo puesto que proporciona información para áreas geográficas en las que no se cuenta con la suficiente información oficial o simplemente no existe, mediante la creación de un “editor comunitario” que permite a editores voluntarios generar información sobre mapas e itinerarios que puedan ser cargados en la aplicación.

Ha recibido distintos premios a lo largo de su trayectoria, entre los más recientes destacan Transport Ticketing Global: la mejor implementación de información de pasajeros en tiempo real del año (2020), Smart Cities Connect-Prêmio Smart 50 (2020) y GovTech100: proveedor líder de movilidad como servicio (MaaS) y aplicación de transporte público número 1 (2020)

“Desarrollado por la mayor comunidad de movilidad del mundo, ayudamos a crear ciudades más limpias, seguras y mejores para vivir, donde la movilidad se convertirá en un facilitador para que personas y empresas descubran infinitas oportunidades”.[5]

2. Capítulo 1. Fundamentos Teóricos.

En este capítulo se presentan los conceptos fundamentales para el diseño y desarrollo del proyecto, entre los más relevantes se encuentran las definiciones del estándar GTFS, Ciudades Inteligentes, algoritmo de Dijkstra, Python y Flask.

2.1 Especificación de Feed de Transito General (GTFS)

2.1.1 Antecedentes de GTFS

Las primeras agencias públicas que pensaron en resolver un problema común como el que resulta al viajar en transporte público y no conocer las rutas que ofrecen o los tiempos de recorrido, fue TriMet en Portland, Oregón junto a Google y esto se ha logrado usando datos abiertos que pueden ser compartidos con el público en general. Google trabajo con TriMet para que se pudiera estandarizar un formato que fuera fácil de manejar y no tuviera problemas al importarlo a Google Maps. Así se conoció GTFS (Especificación de Feeds de Tránsito de Google).

En la actualidad los GTFS están siendo utilizados por un sinnúmero de programadores que utilizan estos datos para crear aplicaciones de diversos servicios, creando los mapas con la información necesaria. Cabe mencionar que, en 2010, para no generar confusión respecto a que se pensaba que era una herramienta única de Google y que únicamente serviría en sus diversas aplicaciones, se cambió el nombre a Especificación de Feeds de Transito General.

GTFS siempre ha sido pensado para resolver necesidades en la comunicación de datos del transporte público, por lo que resulta muy sencillo leerlo o crear aplicaciones para publicar datos que sirvan a distintas causas.

Cabe destacar que el GTFS tiene 2 versiones del estándar; la primera trabaja con información estática y es la que se estudiará en esta investigación, la segunda versión es agregando información en tiempo real. es importante mencionar la diferencia entre ambas versiones, en la versión estática permite obtener un formato común que puede ser utilizado para suministrar los datos y que básicamente se basa en los aspectos del servicio del transporte público como puede ser; rutas, viajes, horarios, etc. y como complementación se encuentra la versión en tiempo real, que siendo un extra a la versión estática permite obtener la información inmediata de los servicios de transporte público como puede ser: estaciones cerradas, retrasos, rutas o líneas que no se encuentran en funcionamiento, etc. Trabajar con la versión uno permite un gran avance dentro del campo de divulgación de información en la ciudad de México.

2.1.2 Definición GTFS

La Especificación de alimentación de tránsito general (GTFS) es una especificación de datos que permite a las agencias de transporte público publicar sus datos de tránsito en un formato que puede ser consumido por una amplia variedad de aplicaciones de software. En la actualidad, miles de proveedores de transporte público utilizan el formato de datos GTFS.

GTFS se divide en un componente estático que contiene información sobre horarios, tarifas y tránsito geográfico y otro con un componente en tiempo real que contiene predicciones de llegada, posiciones de vehículos y avisos de servicio. [6]

2.1.3 GTFS Estático

El GTFS estático, suele usarse para estar al tanto de la operación del transporte público y toda información geográfica que tiene consigo, se compone de básicamente archivos de texto relacionados unos con otros, que permiten modelar los aspectos del transporte público como las agencias, rutas, viajes, frecuencias, horarios, etc.

En la Figura 1, se muestra un diagrama de los diferentes archivos de los que se compone un feed de GTFS que contiene la información de un tránsito estático.

Podemos observar que existen dos tipos de archivos en el diagrama de bloques, los de color verde, son archivos requeridos, es decir, no podemos prescindir de ninguno de ellos. Por otra parte, los que vemos con el color morado, son archivos opcionales que cada programador puede incluir o no, de acuerdo con sus necesidades. Estos archivos están relacionados mediante columnas llave, o bien, identificadores únicos que también se pueden observar en las flechas que apuntan hacia el bloque sucesivo.

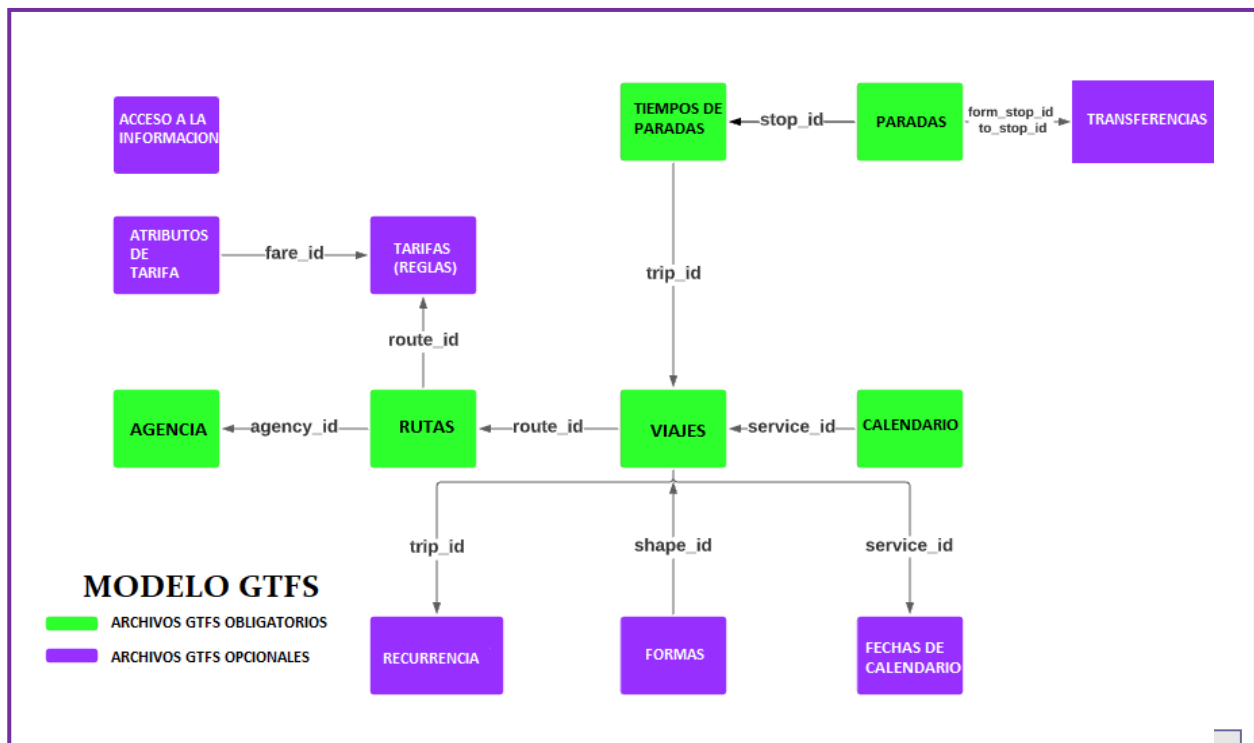


Diagrama a bloques de la composición del modelo de archivos GTFS

Figura 1

2.1.4 GTFS Dinámico

En esta especificación las empresas de transporte público proporcionan información en tiempo real, esto permite tener una mejora en la experiencia ya que permite informar a los usuarios de los horarios actuales de llegada y salida, de esta manera los pasajeros conocerán de manera inmediata los retrasos e imprevistos y se les permitirá cambiar de ruta o esperar para planear mejor su viaje.

Este formato del feed GTFS en tiempo real complementa al GTFS estático.

Actualmente se admite para esta especificación los siguientes tipos de información:

- Actualizaciones de viaje, que contempla retrasos, cancelaciones o cambios de ruta.
- Alertas de servicio, donde se obtiene información acerca de eventos imprevistos que afectan alguna estación o red y acerca de los traslados de paradas.
- Posiciones de vehículos, permite conocer la ubicación y el nivel de embotellamiento.

Un feed combina distintas entidades de diferentes tipos si es necesario, se entregan a través del Protocolo de Transferencia de Hipertexto, HTTP (por sus siglas en inglés) y se actualizan con frecuencia. El archivo es binario por lo que cualquier servidor web puede alojarlo y entregarlo.

2.1.5 Archivos de conjuntos de datos para GTFS estático

Cada archivo tiene especificaciones para su contenido, están divididos en campos los cuales hacen referencia a las propiedades de un objeto. Los campos tienen un valor de diferente tipo, pueden contener texto, números, una URL o un Identificador único (ID). Como se mencionó anteriormente con los archivos opcionales y obligatorios, también aquí contamos con campos obligatorios, opcionales y condicionalmente obligatorios que dependen del uso y las necesidades de cada programador.

Los obligatorios deben incluirse y contener un valor para cada registro; los opcionales pueden omitirse y los condicionalmente obligatorios son requeridos solo con determinadas condiciones.

A continuación, presentamos de forma detallada los archivos requeridos y de manera general los opcionales, tomadas de la referencia oficial del GTFS [7].

2.1.6 Requisitos de archivo.

Los siguientes requisitos son tomados de la página oficial [8] y se requieren para cumplir con el formato GTFS

- Todos los archivos deben guardarse como texto delimitado por comas.
- La primera línea de cada archivo debe contener nombres de campo. Cada subsección de la sección [definiciones de campo](#) corresponde a uno de los archivos en un conjunto de datos GTFS y enumera los nombres de campo que se pueden usar en ese archivo.
- Todos los nombres de campo distinguen entre mayúsculas y minúsculas.
- Los valores de campo no pueden contener tabulaciones, retornos de carro o nuevas líneas.

- Los valores de campo que contienen comillas o comas deben ir entre comillas. Además, cada comilla del valor del campo debe ir precedida de una comilla. Esto es coherente con la forma en que Microsoft Excel genera archivos delimitados por comas (CSV). El siguiente ejemplo demuestra cómo aparecería un valor de campo en un archivo delimitado por comas:
 - **Valor de campo original:** Contains "quotes", commas and text
 - **Valor de campo en archivo CSV:** "Contains ""quotes"", commas and text"
- Los valores de campo no deben contener etiquetas HTML, comentarios o secuencias de escape.
- Elimine los espacios adicionales entre los campos o los nombres de los campos. Muchos analizadores consideran que los espacios son parte del valor, lo que puede provocar errores.
- Cada línea debe terminar con un carácter de salto de línea CRLF o LF.
- Los archivos deben estar codificados en UTF-8 para admitir todos los caracteres Unicode. Los archivos que incluyen el carácter de marca de orden de bytes (BOM) Unicode son aceptables.
- Todos los archivos de conjuntos de datos deben estar comprimidos juntos

2.1.7 Secuencia recomendada para la creación de archivos.

Debido a que los diferentes archivos se vinculan entre sí por medio de un identificador único es recomendable seguir un orden.

- Primero se puede comenzar con el archivo de información de la agencia, el campo que será la llave o identificador es *agency_id*. También crear en este punto el archivo opcional *feed_info* o información del feed nos permitirá tener una mejor organización y será de ayuda para quien haga posterior uso de estos datos.
- Siguiendo con la realización de los archivos de *stops* o paradas y *routes* o rutas, con sus respectivos identificadores *stop_id* y *route_id*.
- Después se crea el archivo de *calendar* o calendario que contiene fechas y horarios, su identificador es *service_id*.
- A esto le siguen los archivos de *stop_times* o tiempo de viajes y el archivo de *trips* o viajes. Para el primero se hará referencia a *trip_id* contenido en el archivo de *trips* o viajes y al campo *stop_id* del archivo de *stops* o paradas. Para el segundo, se hará referencia a *route_id* del archivo *routes* y a *service_id* del archivo de calendario o *calendar*.
- Finalmente, para que el conjunto de datos este completo, se recomienda la creación de los archivos opcionales *fare_attributes* o atributos de tarifa y *fare_rules* o reglas de tarifa, seguido del archivo de *shapes* o formas y al final el archivo de *translation* o transferencias.

Ya que todos los archivos estén creados se deben guardar juntos en una carpeta comprimida para después colocar dicha carpeta en un servidor web público donde las futuras aplicaciones podrán acceder a ella. Si en algún momento el servicio del transporte al que perteneces esos datos cambia, se pide la actualización y publicación de los nuevos datos GTFS comprimidos.

2.2 Recursos para la consulta y uso de los archivos GTFS

2.2.1 Descripción general del Software

Para hacer uso de la información contenida en los archivos GTFS se tienen que escribir, mejor dicho, programar, recursos que nos permitan incorporar a las distintas aplicaciones la información de los transportes públicos.

Para este proyecto se hace indispensable utilizar el lenguaje de programación de PYTHON.

Python es un lenguaje de programación de propósito general de alto nivel que se puede aplicar a muchas clases diferentes de problemas. Es un lenguaje claro y poderoso, podría ser comparado con Java.

Características notables de Python:

- Es fácil de leer ya que su sintaxis es “elegante”
- Es ideal para desarrollar prototipos ya que es fácil de utilizar.
- Cuenta con una gran biblioteca que admite distintas tareas de programación como conectarse a servidores web, buscar texto con expresiones regulares, leer y modificar archivos.
- Cuenta con un modo interactivo el cual te permite probar pequeños fragmentos del código de una forma sencilla.
- Se amplía fácilmente añadiendo nuevos módulos implementados en un lenguaje compilado como C o C ++. También se puede integrar en una aplicación para proporcionar una interfaz programable.
- Se ejecuta en cualquier lugar, incluidos [Mac OS X](#) , [Windows](#) , [Linux](#) y [Unix](#) , con versiones no oficiales también disponibles para [Android](#) e iOS.
- Es software libre en dos sentidos. No cuesta nada descargar o usar Python, o incluirlo en su aplicación. Python también se puede modificar y redistribuir libremente porque, si bien el lenguaje tiene derechos de autor, está disponible bajo una licencia de código abierto.

Algunas características del lenguaje de programación de Python [9] son:

- Hay una variedad de tipos de datos básicos disponibles: números (enteros de punto flotante, complejos y de longitud ilimitada), cadenas (tanto ASCII como Unicode), listas y diccionarios.
- Python admite la programación orientada a objetos con clases y herencias múltiples.
- El código se puede agrupar en módulos y paquetes.

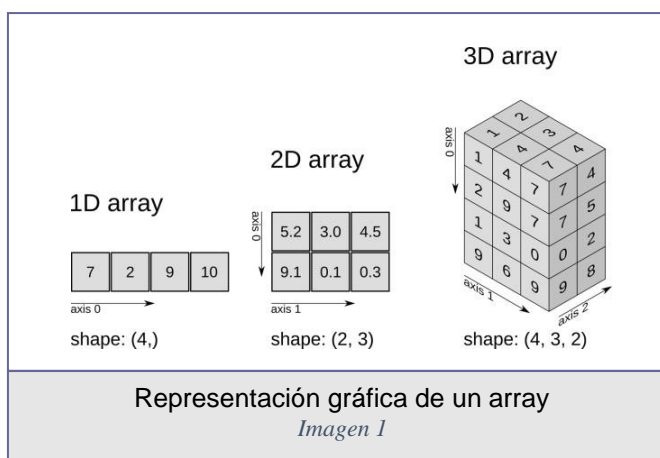
- El lenguaje admite la generación y captura de excepciones, lo que resulta en un manejo de errores más limpio.
- Los tipos de datos se escriben de forma fuerte y dinámica. Mezclar tipos incompatibles (por ejemplo, intentar agregar una cadena y un número) provoca que se genere una excepción, por lo que los errores se detectan antes.
- Python contiene funciones de programación avanzadas como generadores y listas por comprensión.
- La gestión automática de la memoria de Python le libera de tener que asignar y liberar memoria manualmente en su código.

2.2.2 Librerías

A continuación, se describen las librerías más importantes que se utilizaron para la realización del proyecto

NumPy

Esta librería se especializa en el cálculo numérico y el análisis de datos, específicamente para datos que se presentan en gran volumen. Incorpora objetos “Arrays” que nos brindan una mejor representación de una colección de datos de un mismo tipo en diferentes dimensiones, así como funciones eficientes para su manejo. Un array, como se mencionó antes es una estructura de datos de un mismo tipo, y esta se puede organizar en forma de tabla o cuadrícula con una, dos o tres dimensiones de acuerdo con las necesidades del programador.



Pandas

La librería Pandas es una librería que se especializa en el manejo y análisis de estructuras de datos.

Entre sus características más importantes están:

- Definir nuevas estructuras de datos basadas en los arreglos de la librería NumPy, pero con nuevas funciones.
- Nos permite leer y escribir ficheros de una manera sencilla en formatos CSV, EXCEL y bases de datos SQL.
- Podemos acceder a los datos por medio de índices o nombre de filas o columnas.
- Cuenta con métodos para reordenar, dividir y combinar conjuntos de datos.
- Por último, podemos concretar que realiza de manera eficiente todas estas operaciones.

NetworkX

NetworkX es un paquete de Python para la creación, manipulación y estudio de la estructura, dinámica y funciones de redes complejas. [10]

Esta librería nos ayuda a modelar patrones por medio de gráficos, hay distintas cosas que se realizan con el mapeo de los gráficos, por ejemplo, algunos patrones de tráfico, para poder administrar redes de distribución de agua incluso análisis de redes sociales.

Openpyxl

Es la librería que nos permite procesar un archivo de Excel, así podemos controlar el documento de Excel sin necesidad de abrir la aplicación. Se pueden realizar diversas acciones como leer los datos contenidos, escribir en el documento, cambiar el nombre de la hoja, modificar la hoja de trabajo y hasta cambiar el estilo en la hoja.

Folium

Folium es una librería de visualización de datos en Python, cuyo origen fue principalmente para poder visualizar datos geoespaciales, es decir, se pueden crear mapas de cualquier ubicación del mundo con ayuda de esta herramienta. Para ser más específicos, Folium es una biblioteca de JavaScript para trazar mapas interactivos.

De igual manera permite el enlace de datos a un mapa para visualizaciones como el paso de visualizaciones vectoriales (ráster /HTML) como marcadores del mapa. Esta librería cuenta con una serie de mosaicos integrados de OpenStreetMap, Mapbox y Stamen, las cuales son herramientas que permiten crear rutas y mapas dentro de sus aplicaciones. Folium permite conjunto de mosaicos personalizados con clave API de Mapbox o cloudmade además soporta superposiciones de imagen, video, GeoJSON y TopoJSON.

2.2.3 OpenStreetMap

El 9 de agosto de 2004, Steve Coast, un estudiante de física inglés registró openstreetmap.org, ya que él pensó que era una buena idea salir, realizar mapas y subirlos a internet para ahorrar el dinero que cobraban por los mapas La Agencia Nacional de Geografía, en Gran Bretaña. Desde ese momento, este proyecto se pensó como algo libre y que todos pudieran contribuir.

OpenStreetMap proporciona datos de mapas para diversos sitios web, aplicaciones móviles y dispositivos de hardware [11]

Es una herramienta de datos abiertos, o sea que se puede utilizar libremente para diferentes propósitos y es construida a partir de diversos colaboradores que aportan datos de caminos, senderos, lugares, establecimientos, estaciones y todo lo que contribuya para generar los diferentes mapas.

Sus datos siempre están actualizados gracias a que los colaboradores verifican la información utilizando imágenes aéreas, GPS, mapas, entre otras fuentes. En esta comunidad podemos encontrar cartógrafos, profesionales de GIS, ingenieros o personas que elaboran simplemente mapas en sus zonas cercanas.

Así que el único requisito para utilizar los mapas que se encuentran disponibles es dar crédito a OpenStreetMap y a sus colaboradores.

2.2.4 Algoritmo de Dijkstra

Por definición, un algoritmo es una serie de pasos organizados, que describe el proceso que se debe seguir, para dar solución a un problema específico.

Entonces, debemos pensar en la forma de resolver un problema paso a paso, tomando en cuenta que, si un paso no se resuelve, no puede pasar al siguiente.

Lo que caracteriza a un algoritmo es que debe ser:

- Preciso, es decir, que sea claro en cada uno de sus pasos.
- Definido, esto conlleva a delimitar sus funciones para que no realice operaciones que no estén contempladas, así se asegurará que, si el algoritmo funciona en repetidas ocasiones con los mismos datos, siempre se obtendrá el mismo resultado.
- Finito, o sea que siempre tendrá un fin.

Hoy en día podemos encontrar distintos tipos de algoritmos aplicados a la vida cotidiana, estos algoritmos resultan de suma importancia puesto que modifican constantemente nuestra forma de vida haciendo más fáciles diversas tareas.

Uno de los más relevantes es el **algoritmo de Dijkstra**, (también conocido como el algoritmo de caminos mínimos) que fue una parte esencial para lograr los principales objetivos de este proyecto y el cual se utiliza en diversas aplicaciones en general, de diferentes propósitos. Un ejemplo, en las comunicaciones, es la capacidad que Dijkstra tiene para encontrar el camino más corto disponible por donde se puede enviar un paquete de información entre los diferentes nodos de una red de comunicación; otra aplicación importante es en la asistencia para el enrutamiento de aviones y tráfico aéreo, o en el reconocimiento del lenguaje hablado pudiendo así determinar cuál es la frase más utilizada en un idioma específico.

Este modelo se puede clasificar dentro de los algoritmos de búsqueda, su principal objetivo es determinar una ruta de menor coste con respecto a otras posibilidades desde un nodo origen a cualquier otro nodo de la red.

Trabaja con una metodología basada en iteraciones (repeticiones conocidas comúnmente) y por lo tanto la facilidad de desarrollo se encuentra en redes de menor tamaño, como aumenta el tamaño aumenta la dificultad.

El algoritmo de Dijkstra fundamentalmente realiza el uso y definición de etiquetas a partir del nodo origen y para cada nodo consecuente, la información que contienen estas etiquetas es relacionada a un valor acumulado del tamaño de los arcos y con la procedencia más próxima de la ruta. Estas etiquetas pueden ser temporales o permanentes dependiendo el caso, para las temporales se refiere a Etiquetas que se pueden modificar si existe una mínima posibilidad de encontrar una ruta más corta

mediante este nodo, en caso de no tener esta posibilidad se convierte en un nodo permanente.

2.3 Herramientas para el diseño de la Aplicación Web

2.3.1 HTML

El lenguaje HyperText Markup Language (Lenguaje de Marcado de Hipertexto) fue creado en 1991 por **Tim Berners-Lee** quien es considerado como padre de la WWW (World Wide Web). Para su definición se tomó como base un metalenguaje de marcado llamado Standard Generalized Markup Language (SGML), este estándar permite especificar otros lenguajes de marcado, y por lo cual fue buena referencia para la creación de HTML.

HTML se utiliza para la elaboración de páginas web, así mismo define una estructura básica y un código para el contenido de una página web como: imágenes, textos, videos, etc.

Las etiquetas son el elemento fundamental del código HTML. Estas etiquetas son rodeadas por corchetes angulares (<HTML>) y hacen referencia al código con las instrucciones dadas a una plataforma computacional. Estas instrucciones representan todos los enlaces que vinculan los contenidos de un sitio en específico.

Los elementos son fundamentales dentro de la estructura básica de HTML y tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene restricciones para que se considere válido el documento. Este lenguaje puede ser creado y editado con cualquier editor de textos básico, como Gedit, GNU/LINUX, el bloc de notas de Windows o cualquier editor de texto sin formato GNU, TextPad, Vim, etc.

Con HTML se utiliza CSS u hojas de Estilo en Cascada, este no es un lenguaje de programación como tal, solo te permite aplicar estilos de forma selectiva en elementos contenidos en documentos en formato HTML

2.3.2 Framework Flask

Framework es una herramienta que nos facilita un esquema de trabajo y nos ayuda a la construcción de páginas web dinámicas. Permite agilizar los procesos al desarrollar porque evita repetir en distintas ocasiones código, asegura entonces buenas prácticas y consistencia del código.

Algunas características de usar un framework son:

- El desarrollador dispondrá de una estructura o esqueleto sobre el cual desarrollar la aplicación, lo que representa un ahorro de tiempo.
- Define estándares de programación lo que facilita que otros colaboradores puedan ayudar en el desarrollo
- Las herramientas para utilizarlo están disponibles, así como módulos e información en general. Existe una comunidad de desarrolladores que pueden ayudar a responder consultas.
- Generalmente proporciona una mayor seguridad, al tener resueltas varias vulnerabilidades.

Por lo general los frameworks están asociados a un lenguaje de programación, en este caso, utilizaremos Flask para trabajar en Python.

Flask se puede catalogar como un micro Framework y es útil para desarrollar una aplicación básica o que se requiera desarrollar rápidamente. Incluye un servidor web de desarrollo por lo que no necesita una infraestructura con un servidor web para que se realicen pruebas de las aplicaciones si no que dentro del mismo entorno se puede correr para ir visualizando los resultados que se van generando.

Ahora, si se comete un error en el código que se está construyendo, se cuenta con un depurador y soporte incluido, como dijimos es compatible con Python. Tiene un buen manejo de rutas, soporta cookies seguras, tiene la opción de utilizar sesiones, funciona para construir aplicaciones de contenido estático y, por último, es open source amparado con una licencia BSD.

Por lo tanto, Flask es una excelente alternativa para la construcción de aplicaciones Webs, tiene una buena curva de aprendizaje, lo que facilita su aprendizaje en un menor tiempo.

2.4 Ciudades inteligentes

Como se menciona brevemente en la justificación, la población suele migrar a zonas urbanas con el objetivo de encontrar mejores condiciones de vida lógicamente este fenómeno produce un aumento en la demanda de recursos naturales, energía, agua, saneamiento educación, movilidad, servicios de salud, entre otros. Las Tecnologías de la Información y la Comunicación (TIC) tienen la capacidad de dar soluciones económicamente viables y amigables con el entorno por lo que son de gran ayuda al desarrollar herramientas para la gestión de fuentes de agua, eficiencia energética, residuos sólidos, infraestructura de transporte público ayudando en la reducción de la congestión en el mismo, crecimiento de la infraestructura de las TIC, impacto ambiental, etc.

Por lo anterior algunas de ciudades en el mundo se han comenzado a etiquetar como Ciudades inteligentes.

El Grupo Temático de la UIT-T sobre Ciudades Inteligentes y sostenibles ha conceptualizado y generalizado la definición de Ciudad Inteligente y las características de una ciudades inteligentes y sostenibles.

Existen cuatro temas básicos presentes en una ciudad inteligente y sostenible:

- Sociedad: La ciudad es para sus habitantes (es decir, los ciudadanos)
- Economía: La ciudad debe tener la capacidad de prosperar en aspectos como el empleo, las finanzas y el desarrollo económico.
- Medio Ambiente: La ciudad tiene que ser sostenibles en su funcionamiento para las generaciones del presente y futuro.
- Gobierno: La ciudad tiene que ser lo suficientemente robusta en cuestión de capacidades administración, así como de otros elementos, por ejemplo, de algunas cuestiones políticas, etc.

Por lo tanto una ciudad inteligente sostenible es una metrópoli que usa las TIC de manera confiable, adaptable, escalable, accesible, segura, cautelosa y adaptable con el fin de mejorar la calidad de vida de los ciudadanos, garantizar una mejoría económica, mejorar el bienestar de los ciudadanos en aspectos como educación, atención médica y seguridad física, pero siempre estableciendo una visión ambientalmente responsable y sostenible, reforzando la funcionalidad de prevención y el manejo de los desastres naturales, incluyendo el correcto manejo de los impactos del cambio climático y algunos otros problemas propiciados por los humanos. Así mismo la modernización de servicios que están basados en infraestructura física, tales como el transporte, agua, servicios públicos, telecomunicaciones y los sectores manufactureros.

Por último, pero no menos importante, es la capacidad de generar una normativa balanceada y con alta eficacia, así como mecanismos de cumplimiento, gobernabilidad mediante políticas y procesos más eficientes y equitativos, todos de una manera estandarizada y uniforme.

Se puede notar por ejemplo en la movilidad, donde la llamada movilidad inteligente propicia el movimiento de las personas y mercancías, paralelamente mejora los

recursos ambientales, económicos y humanos, priorizando los viajes articulados entre diferentes modos de transporte. Pero para que esto se realice, es necesaria una gobernabilidad inteligente, que incluye la participación política y activa, algunos servicios de ciudadanía, etc.

2.4.1 Cambios imprescindibles en la sociedad

Sin duda alguna se está pasando por uno de los cambios más profundos en el ámbito económico y social impulsados por el avance de la tecnología, la revolución tecnológica de la que muchos estudiosos hablan, está transformando constantemente nuestro entorno, desde nuestra forma en la que vivimos en nuestra cotidianidad hasta la manera en que se trabaja y se relaciona la sociedad. Uno de los soportes de esta revolución es la tecnología de la información y las telecomunicaciones. Los nuevos desarrollos tecnológicos usan como piedra angular la creciente capacidad para el intercambio, producción y procesamiento de datos que son procesados y almacenados en la nube mediante nuevas herramientas como la IA y la analítica de datos. Sin embargo, para la transferencia de datos masivos son necesarias las redes de telecomunicaciones, cada vez con una mayor capacidad y velocidad. Al final tecnologías como el 5G permiten potenciar esas tecnologías, no precisamente por el desarrollo de tecnologías específicas sino por una convergencia entre ellas.

Ahora bien, la movilidad forma parte de los grandes desafíos de las urbes del siglo XXI, por lo que resulta necesario avanzar en modelos que permitan planificar y gestionar dando un enfoque sostenible, que den garantía del funcionamiento del sistema económico, permitan una equidad social sana y con el medio ambiente, es importante mencionar que la producción de datos masivos de geolocalización es uno de los atributos de suma importancia y del cual es imprescindible analizar y gestionar constantemente. Se suelen generar importantes cantidades de datos geolocalizados los cuales son registrados a través del GPS que forma parte esencial de los teléfonos móviles. Por lo que la huella digital de los humanos y la localización de estos en el espacio y tiempo permite seguir en ocasiones minuciosamente el desplazamiento de un individuo. Los datos masivos que captan información de cada individuo contienen un campo en el que se tiene el identificador de usuario, Esta es la manera en que se puede rastrear la localización espacial de una persona, por lo consiguiente analizar los patrones de movilidad, entre mayor sea la resolución obtenida de manera temporal y espacial, entonces la información será más precisa. Así mismo para captar a detalle la relación de viajes son usadas las aplicaciones instaladas en los teléfonos móviles como Google Maps, localizadores de vehículos en autobuses urbanos, servicios de movilidad compartida aportando información de alta resolución espacial.

El servicio que ofrece Google Maps, así como la misma WEB, permitiendo calcular matrices origen destino en los diversos modos existentes gracias a los ficheros GTFS proporcionados por las autoridades de transporte de las ciudades, por lo que estas aproximaciones son el ámbito teórico de acuerdo con los horarios y frecuencias, no

de los tiempos de viaje reales. Los GTFS pueden ser cargados para analizar la cobertura que tienen las redes de transporte público, las velocidades y superposición de las líneas, cálculo de franjas horarios que pueden usarse en algunos estudios dinámicos propios de la accesibilidad.

2.5 Datos Abiertos

2.5.1 Normativa de datos abiertos

Primero debemos conocer el significado de “Gestión de Datos” lo cual se refiere a un conjunto de prácticas, procesos y procedimientos que se llevan a cabo en el manejo de los datos. Existe una política dentro de la CDMX referente a esta gestión de datos, el cual es un documento normativo que permite conocer las pautas que los organismos públicos deben seguir para el manejo de todos los datos públicos.

Esta política está integrada por:

1. Marco para la Gestión de Datos
2. Modelo de Madurez
3. Lineamientos, estándares y buenas practicas

El Marco para la gestión de Datos es un tipo de guía que define y delimita la implementación de la política de gestión de datos. Mientras que su función es ser la base para los objetivos que se encuentren claros, razonables y medibles. Y se encuentra formada por 8 componentes, que pueden ser mediante acciones o practicas presentes durante toda la vida útil de los datos.

Estos 8 componentes son los siguientes:

1. Integración e Inteligencia de Datos
2. Datos no Estructurados
3. Arquitectura de Datos
4. Datos Maestro y de Referencia
5. Metadatos
6. Seguridad de datos
7. Calidad de datos
8. Y Finalmente como conjunto general es la gobernanza y Cultura de datos.

Estos en conjunto permiten que se tenga un almacenamiento y acceso seguro, eficiente y oportuno cuando se trabaja una gran cantidad de datos. Además, esto permite facilitar el acceso a datos integrados y consistentes para la toma de decisiones que se basan en evidencias.

Modelo de madurez

Este modelo de madurez es un instrumento metodológico que facilita medir el grado de avance de la implementación de la política de Gestión de Datos y el término de madurez de la gestión se refiere al grado en que los componentes del MGD-CDMX evolucionan desde una línea base.

Este modelo tiene 4 fases durante el proceso:

Nivel de Madurez Inicial: aquí el Ente de la Administración Pública no tiene definidos los procesos necesarios para mantener las prácticas de gestión de datos. Aquí se depende mucho de individuos con conocimiento nulo en la gestión.

Nivel de madurez repetible: el ente de la administración pública tiene conocimiento a nivel interno sobre la gestión de datos, así como buenas prácticas en el manejo de estos.

Nivel de madurez definido: El ente de la Administración pública ya utiliza un conjunto de procesos definidos que se comparten como unas recomendaciones para el uso correcto de los datos.

Nivel de madurez gestionado: como último punto del modelo se tiene el nivel de madurez gestionado donde ya se utiliza el conjunto de los procesos definidos, que se publican para el uso obligatorio.

Como puntos para tener en cuenta dentro del marco legal para el uso y publicación de datos se tiene:

- Ley de Operación e innovación Digital para la Ciudad de México.
- Política de gestión de Datos de la Ciudad de México
- Lineamientos en Materia de seguridad de la información en la administración pública de la ciudad de México
- Ley de Transparencia, Acceso a la información Pública y Rendición de Cuentas de la Ciudad de México
- Ley de Archivos del Distrito Federal

Se debe de tener un correcto gestión de datos ya que con una mala gestión los datos son de mala calidad, poco seguros y difíciles de integrar con otros. Esto significa que los costos y riesgos aumentan dificultando su uso y aprovechamiento.

3. Capítulo 2. Métodos para el desarrollo de la aplicación web.

Este capítulo tiene la finalidad de mostrar cómo se realizó la consulta y depuración de la información encontrada en los archivos del estándar GTFS.

Así mismo, contiene los diagramas de flujo correspondientes al diseño del algoritmo que hace posible el funcionamiento de la aplicación web.

3.1 Procedimientos y técnicas utilizadas.

3.1.1 Consulta y depuración de archivos GTFS

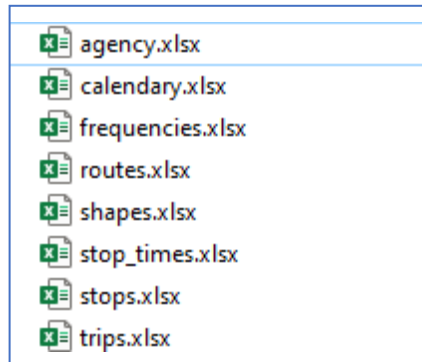
Como se ha mencionado antes, los archivos del estándar GTFS de los transportes públicos de la Ciudad de México se encuentran en el portal de datos abiertos, en la categoría de movilidad.

Ahí se encontró un archivo comprimido con extensión ZIP, esto se debe a que así lo marca el estándar GTFS en sus especificaciones.

Al descargar el archivo y descomprimirlo, se pueden extraer 8 archivos en formato de texto (.txt):



Después de extraer cada archivo, se realiza una conversión a formato Excel (.xlsx). Este paso fue necesario debido a que el proyecto se tuvo que delimitar para extraer únicamente los datos del Sistema de Transporte Colectivo Metro, ya que estos archivos generales, incluyen todos los servicios de la Ciudad de México, por lo que se considera que en un archivo .txt, no se podría realizar el análisis de la estructura de los datos, ni depurarla.



Conversión de los archivos .txt a .xlsx

Cada archivo cuenta con una llave o columna ID que lo relaciona con otro archivo, por lo que se tuvo que seguir la conexión entre ellos de acuerdo con la organización del modelo del GTFS y obtener todos los datos asociados que necesitábamos conocer.

El primero que se abrió fue el archivo agency.txt, el cual cuenta con cuatro campos que nos indican, el identificador único ID, el nombre de la agencia de transportes, la zona y la página web.

Column1	Column2
agency_id	agency_name
CC	Corredores Concesionados
METRO	Sistema de Transporte Colectivo Metro
MB	Metrobús
TROLE	Trolebús
RTP	Red de Transporte de Pasajeros
TL	Servicio de Tren Ligero
SUB	Ferrocarriles Suburbanos

Primeras dos columnas de datos del archivo agency.xlsx

Imagen 4

Column3	Column4
agency_timezone	agency_url
America/Mexico_City	https://datos.cdmx.gob.mx/explore/dataset/rutas-y-corredores-del-transporte-publico-concesionado/table/
America/Mexico_City	https://www.metro.cdmx.gob.mx/
America/Mexico_City	https://www.metrobus.cdmx.gob.mx
America/Mexico_City	https://www.ste.cdmx.gob.mx/
America/Mexico_City	https://www.rtp.cdmx.gob.mx
America/Mexico_City	https://www.ste.cdmx.gob.mx/tren-ligero
America/Mexico_City	http://www.fsuburbanos.com/

Muestra de las últimas dos columnas del archivo agency.xlsx

Imagen 5

Entonces, en la columna *agency_id* se puede encontrar el ID que se relacionará en el siguiente archivo, siendo **METRO** el dato de interés.

Ahora se abre el archivo routes.xlsx donde se encuentra la información de las rutas de los transportes, aquí está contenido el campo de *agency_id* y la información de su llave que es *route_id*.

Para fines prácticos, solo se hará el desglose de cómo se realiza la búsqueda de la estación politécnico, donde al final se puede encontrar su ubicación en coordenadas geográficas y el tiempo promedio que sucede entre la estación Hangares a Politécnico, todo esto siguiendo la relación de las columnas llave de cada archivo.

agency_id	route_color	route_desc	route_id	route_long_name	route_short_name
METRO	B99E51		CMX020L12	Tláhuac - Mixcoac	12
METRO	118751		CMX0200LB	Ciudad Azteca - Buenavista	B
METRO	A02D96		CMX0200LA	Pantitlán - La Paz	A
METRO	512826		CMX0200L9	Pantitlán - Tacubaya	9
METRO	118749		CMX0200L8	Garibaldi - Constitución de 1917	8
METRO	E87511		CMX0200L7	El Rosario - Barranca del Muerto	7
METRO	D81E05		CMX0200L6	El Rosario - Martín Carrera	6
METRO	F9D616		CMX0200L5	Politécnico - Pantitlán	5
METRO	7FBCAA		CMX0200L4	Santa Anita - Martín Carrera	4
METRO	AD9B0C		CMX0200L3	Indios verdes - Universidad	3
METRO	0071C1		CMX0200L2	Cuatro Caminos - Tasqueña	2
METRO	F94F8E		CMX0200L1	Pantitlán - Observatorio	1

Datos contenidos en el archivo routes.xlsx que muestran los resultados de la agencia "METRO"

Imagen 6

En la **Imagen No.6** se puede observar que la primera columna es la agencia y que al hacer la búsqueda de la palabra “METRO”, arroja 12 resultados, los cuales corresponden a las rutas con las que cuenta este transporte específicamente.

Existen distintos campos como el color de la ruta en hexadecimal, el nombre largo, el nombre corto y hay una columna vacía puesto que se puede añadir una descripción, pero no ha sido colocada para este transporte.

De la columna *route_id* se obtiene el ID **CMX0200L5** para la conexión con el archivo **trips.xlsx**

En el archivo *trips.xlsx* se encuentra la información de los viajes, de la misma forma que en el archivo anterior, al buscar el *route_id* **CMX0200L5** se encuentran 3 viajes de ida y vuelta para esta ruta.

trip_id	trip_short_name	direction_id	route_id	service_id	shape_id	trip_headsign
02200L3000_1	to Indios Verdes	1	CMX0200L3	2STC2	SH0200L3000_1	Indios Verdes
02200L3000_0	to Universidad	0	CMX0200L3	2STC2	SH0200L3000_0	Universidad
02100L3000_1	to Indios Verdes	1	CMX0200L3	2STC1	SH0200L3000_1	Indios Verdes
02100L3000_0	to Universidad	0	CMX0200L3	2STC1	SH0200L3000_0	Universidad
02300L4000_1	to Santa Anita	1	CMX0200L4	2STC3	SH0200L4000_1	Santa Anita
02300L4000_0	to Martín Carrera	0	CMX0200L4	2STC3	SH0200L4000_0	Martín Carrera
02200L4000_1	to Santa Anita	1	CMX0200L4	2STC2	SH0200L4000_1	Santa Anita
02200L4000_0	to Martín Carrera	0	CMX0200L4	2STC2	SH0200L4000_0	Martín Carrera
02100L4000_1	to Santa Anita	1	CMX0200L4	2STC1	SH0200L4000_1	Santa Anita
02100L4000_0	to Martín Carrera	0	CMX0200L4	2STC1	SH0200L4000_0	Martín Carrera
02300L5000_1	to Politécnico	1	CMX0200L5	2STC3	SH0200L5000_1	Politécnico
02300L5000_0	to Pantitlán	0	CMX0200L5	2STC3	SH0200L5000_0	Pantitlán
02200L5000_1	to Politécnico	1	CMX0200L5	2STC2	SH0200L5000_1	Politécnico
02200L5000_0	to Pantitlán	0	CMX0200L5	2STC2	SH0200L5000_0	Pantitlán
02100L5000_1	to Politécnico	1	CMX0200L5	2STC1	SH0200L5000_1	Politécnico
02100L5000_0	to Pantitlán	0	CMX0200L5	2STC1	SH0200L5000_0	Pantitlán
02300L6000_1	to El Rosario	1	CMX0200L6	2STC3	SH0200L6000_1	El Rosario
02300L6000_0	to Martín Carrera	0	CMX0200L6	2STC3	SH0200L6000_0	Martín Carrera
02200L6000_1	to El Rosario	1	CMX0200L6	2STC2	SH0200L6000_1	El Rosario
02200L6000_0	to Martín Carrera	0	CMX0200L6	2STC2	SH0200L6000_0	Martín Carrera
02100L6000_1	to El Rosario	1	CMX0200L6	2STC1	SH0200L6000_1	El Rosario
02100L6000_0	to Martín Carrera	0	CMX0200L6	2STC1	SH0200L6000_0	Martín Carrera

Datos contenidos en el archivo trips.xlsx
Imagen 7

En la **Imagen No.7** se puede ver el contenido del archivo donde aparece la información del identificador del viaje, su nombre corto, el identificador de dirección,

el identificador de ruta, el identificador de servicio, el identificador de figura y el encabezado del viaje.

Para continuar con la búsqueda se requiere **el identificador del viaje**, aunque existen seis posibles identificadores, uno por cada viaje realizado, se toma cualquiera que vaya con dirección a Politécnico (to Politécnico). Por comodidad, se elige el primero **02300L5000_1**.

Ahora, se consulta el archivo *stop_times.xlsx* donde se observa el tiempo de inicio y llegada de cada viaje, la secuencia de las estaciones y el identificador de la parada. Aquí se puede demostrar que si en el paso anterior, se hubiera elegido un *trip_id* diferente, no cambiaría el *stop_id* que estamos buscando para que nos brinde la información final de cada estación

trip_id	stop_id	stop_sequence	timepoint	departure_time	arrival_time
02300L5000_1	0200L5-POLITÉCNICO	13	0	12:22:50 a. m.	12:22:50 a. m.
02200L5000_1	0200L5-PANTITLÁN	1	0	12:00:00 a. m.	12:00:00 a. m.
02200L5000_1	0200L5-HANGARES	2	0		
02200L5000_1	0200L5-TNALAÉREA	3	0		
02200L5000_1	0200L5-OCEANÍA	4	0		
02200L5000_1	0200L5-ARAGÓN	5	0		
02200L5000_1	0200L5-EDUARDOMOLI	6	0		
02200L5000_1	0200L5-CONSULADO	7	0		
02200L5000_1	0200L5-VALLEGÓMEZ	8	0		
02200L5000_1	0200L5-MISTERIOS	9	0		
02200L5000_1	0200L5-LARAZA	10	0		
02200L5000_1	0200L5-AUTOBUSESNTE	11	0		
02200L5000_1	0200L5-ITOPETRÓLEO	12	0		
02200L5000_1	0200L5-POLITÉCNICO	13	0	12:22:50 a. m.	12:22:50 a. m.
02100L5000_1	0200L5-PANTITLÁN	1	0	12:00:00 a. m.	12:00:00 a. m.
02100L5000_1	0200L5-HANGARES	2	0		
02100L5000_1	0200L5-TNALAÉREA	3	0		
02100L5000_1	0200L5-OCEANÍA	4	0		
02100L5000_1	0200L5-ARAGÓN	5	0		
02100L5000_1	0200L5-EDUARDOMOLI	6	0		
02100L5000_1	0200L5-CONSULADO	7	0		
02100L5000_1	0200L5-VALLEGÓMEZ	8	0		
02100L5000_1	0200L5-MISTERIOS	9	0		
02100L5000_1	0200L5-LARAZA	10	0		
02100L5000_1	0200L5-AUTOBUSESNTE	11	0		
02100L5000_1	0200L5-ITOPETRÓLEO	12	0		
02100L5000_1	0200L5-POLITÉCNICO	13	0	12:22:50 a. m.	12:22:50 a. m.

Datos contenidos en el archivo *stop_times.xlsx*

Imagen 8

Finalmente, se llega al último archivo de datos llamado *stops.xlsx* donde se busca el identificador **0200L5-POLITÉCNICO** para terminar la búsqueda.

stop_id	stop_lat	stop_lon	stop_name	wheelchair_boarding
0200L5-PANTILÁN	19.4151	-99.07433	Pantitlán	
0200L5-HANGARES	19.42435	-99.08813	Hangares	
0200L5-TNALAÉREA	19.43427	-99.08808	Terminal Aérea	
0200L5-OCEANÍA	19.44502	-99.08687	Oceania	
0200L5-ARAGÓN	19.45126	-99.09621	Aragón	
0200L5-EDUARDOMOLI	19.4514	-99.1053	Eduardo Molina	
0200L5-CONSULADO	19.45514	-99.11354	Consulado	
0200L5-VALLEGÓMEZ	19.45887	-99.11949	Valle Gómez	
0200L5-MISTERIOS	19.46311	-99.13024	Misterios	
0200L5-LARAZA	19.46967	-99.13652	La Raza	
0200L5-AUTOBUSES NTE	19.47907	-99.14065	Autobuses del Norte	
0200L5-ITOPETRÓLEO	19.48934	-99.14471	Instituto del Petróleo	
0200L5-POLITÉCNICO	19.50068	-99.14921	Politécnico	
0200L4-SNTAANITA	19.40289	-99.1217	Santa Anita	
0200L4-JAMAICA	19.41092	-99.12174	Jamaica	
0200L4-FRAYSERVANDO	19.42159	-99.12053	Fray Servando	
0200L4-CANDELARIA	19.42876	-99.11943	Candelaria	

Libro	Hoja	Nombre	Celda	Valor	Fórmula
stops.xlsx	stops		\$A\$2682	0200L5-POLITÉCNICO	

1 celda(s) encontradas

En la Imagen No.9 se pueden

Datos contenidos en el archivo stop.xlsx
Imagen 9

observar los datos de interés, la latitud y longitud de cada estación, estos datos son los que alimentan al algoritmo de Dijkstra para arrojar las recomendaciones de cada ruta basándose en estos puntos.

En todo el seguimiento y recorrido por los contenidos de los archivos, se puede destacar la cantidad de información acerca de las estaciones y de los diferentes transportes, además se pueden complementar o crear nuevos campos que sigan ayudando al desarrollo de nuevas funciones en las aplicaciones.

3.1.2 Funcionamiento del Algoritmo

Para explicar el funcionamiento del algoritmo que se desarrolló para la aplicación web, se presenta el siguiente diagrama de flujo que refleja los pasos que se siguen de manera general para utilizar los datos que se obtuvieron del estándar GTFS, generar las rutas, implementar la aplicación de IA y mandar estos resultados juntos a la interfaz gráfica de la aplicación web.

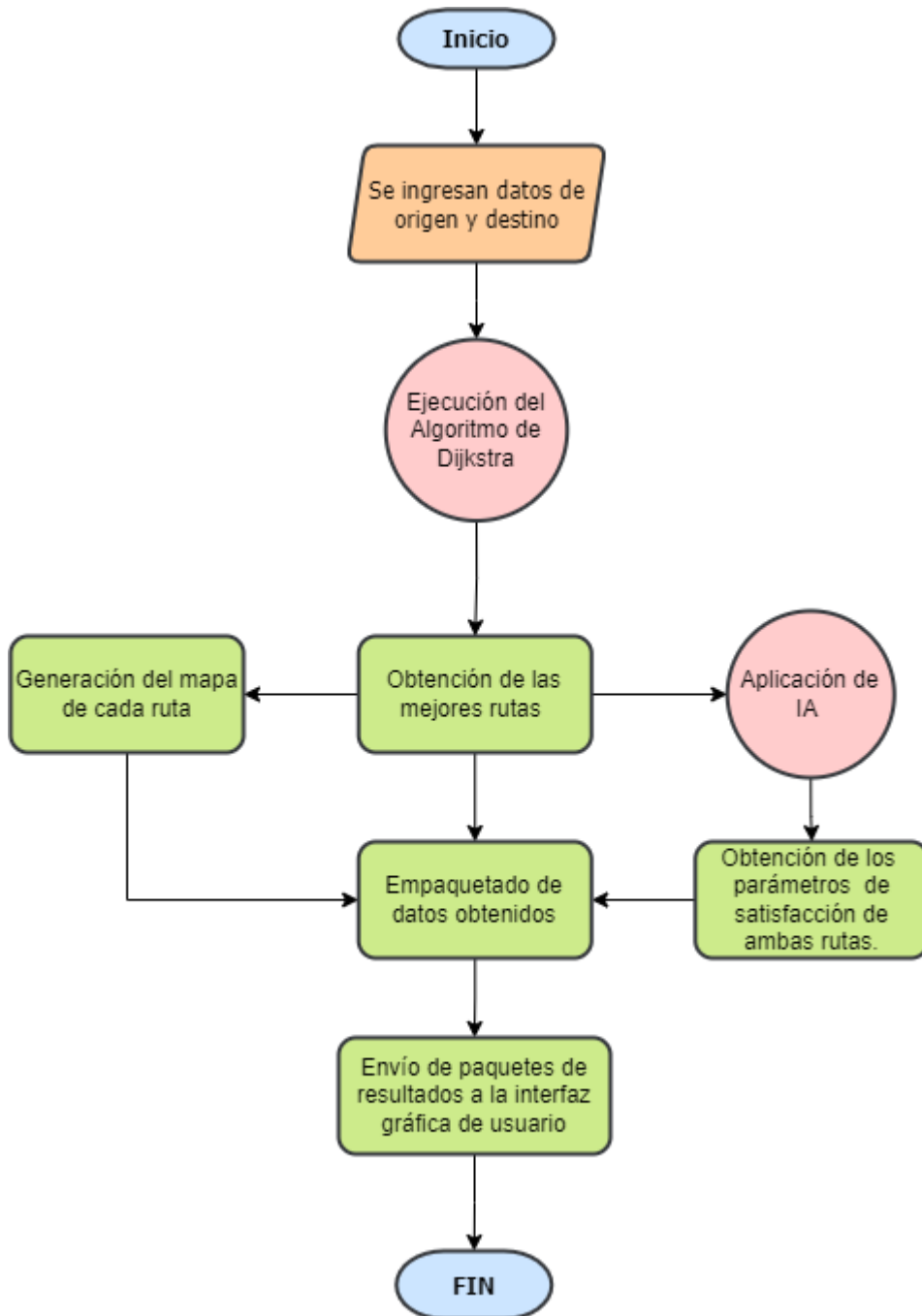


Diagrama de Flujo del funcionamiento general del algoritmo.

Figura 2

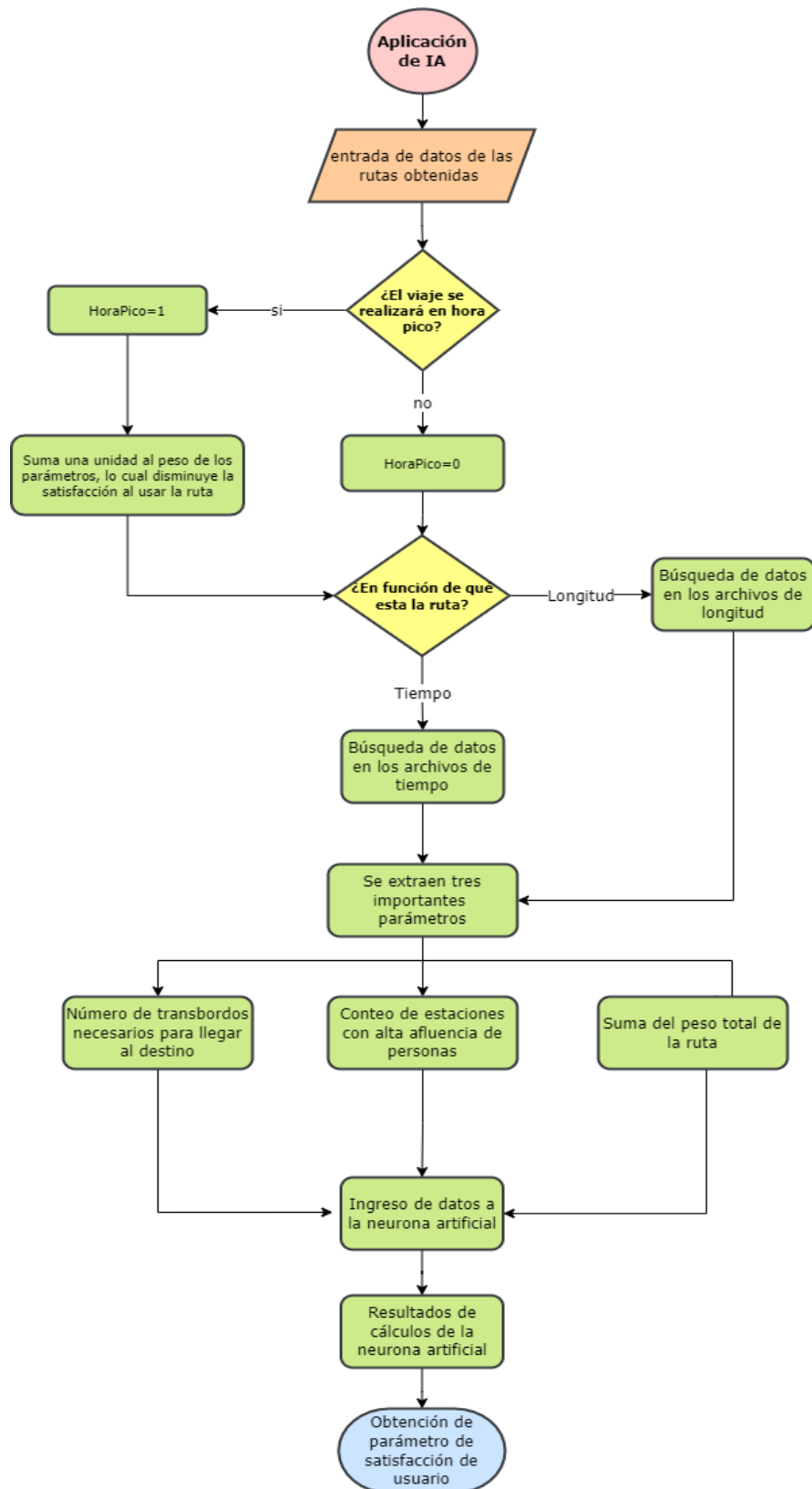


Diagrama de Flujo del funcionamiento general de la Inteligencia Artificial

Figura 2

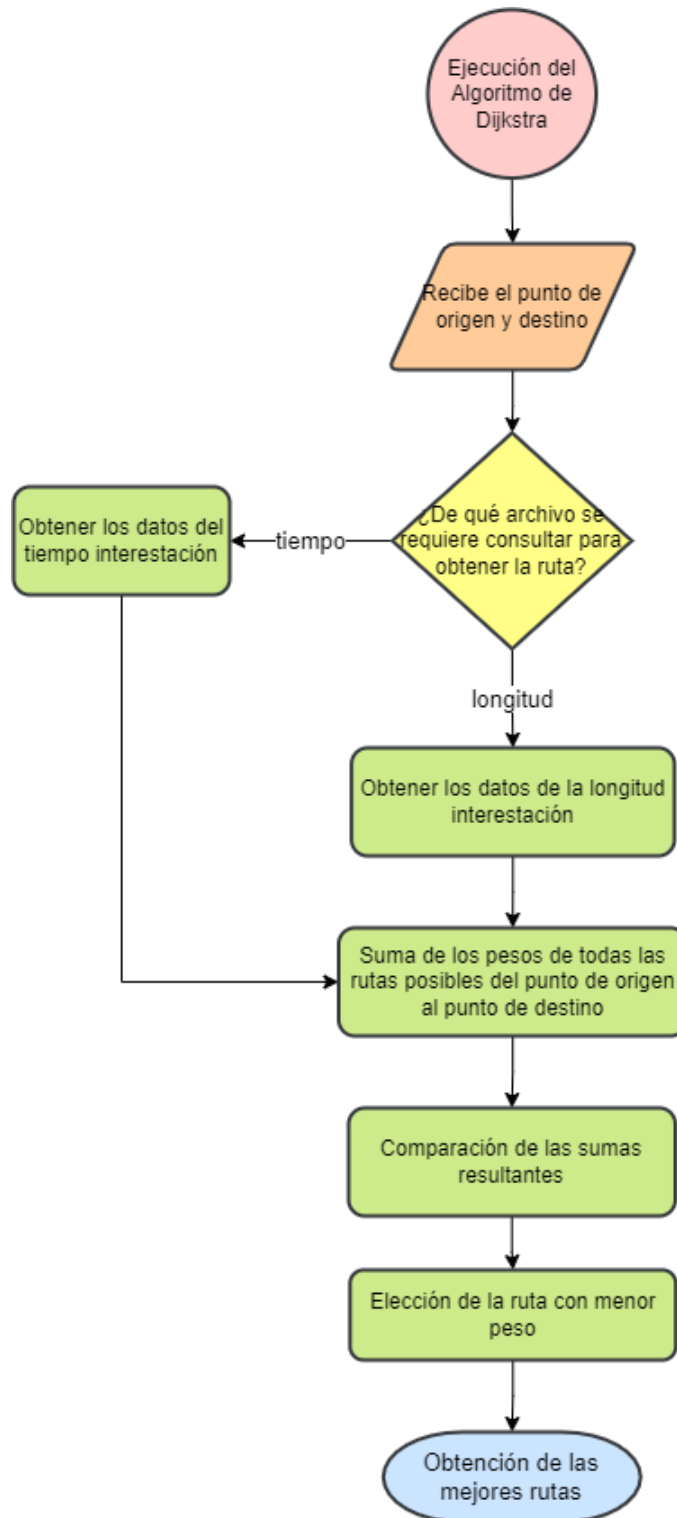


Diagrama de Flujo del funcionamiento general del Algoritmo de Dijkstra

4. Capítulo 3. Metodología

En este capítulo se muestra la metodología que permitió la creación de la aplicación WEB, el entorno gráfico al que se llegó y las pruebas del funcionamiento al calcular un ejemplo de ruta.

4.1 Back-End

El Back-End se refiere a toda la arquitectura interna del sitio que asegura que todos los elementos desarrollen una función adecuada, ésta no se encuentra visible para el usuario y no incluye ningún elemento gráfico, únicamente código.

Para este proyecto, esta programación se realizó en Python como se ha mencionado antes, desde la obtención de los datos de los archivos GTFS hasta el algoritmo de IA. Sin embargo, se requirió de un Framework para tener un esquema de trabajo y diversas utilidades y funciones en la creación de la aplicación web.

Específicamente el Framework Flask, se usó para conseguir la interacción entre el Front-End y el Back-End.

Lo primero que se realizó una vez que se obtuvo la información recopilada de los archivos de GTFS fue programar el algoritmo de Dijkstra para obtener las rutas más cortas, una en relación con la longitud Inter estación y otra en relación con el tiempo Inter estación. Aquí se requirió utilizar también librerías como Pandas y Openpyxl. Mientras que para el caso del algoritmo de Dijkstra se simplifico el proceso gracias a la librería NetworkX.

A continuación en la Imagen No. 10, se muestra el código implementado para hacer los pasos propuestos en el algoritmo de Dijkstra, obteniendo directamente desde el Front-End los parámetros que se solicitan, en primer lugar, con ayuda de estos parámetros se le asignan a la variable “num1” y “num2”, para que sean buscados en el archivo correspondiente, luego se aplica el algoritmo de Dijkstra y se calcula la ruta más optima conforme a longitud y tiempo Inter estación, puesto que para la obtención de ambas rutas se usa el mismo método. También se puede notar en la Terminal las dos estaciones que se ingresaron en la interfaz de usuario, en este caso solo para tener de manera gráfica los datos ingresados.

```
42 # PARA EL NUMERO DE ESTACIONES QUE RECORRIO
43 if request.method == 'POST':
44     hora_pico = request.form["hora_pico"]
45     num1 = request.form['num1']
46     num2 = request.form['num2']
47     resultado_final=[]
48     print(num1)
49     print(num2)
50     df = pd.read_excel('app2\metro.xlsx')
51     print(df)
52 #Cálculo de ruta en función de longitud interstación
53 METRO = nx.from_pandas_edgelist(df, source='Origen', target='Destino', edge_attr=('Longitud de interstación'))
54 djk_path = nx.dijkstra_path(
55     METRO, source=num1, target=num2, weight='Longitud de interstación')
56 djk_path
57
58 print(djk_path, "LISTA CON RESPECTO A LONGITUD INTERESTACIÓN")#resultado de ruta propuesta
59 resultado_final.append("RUTA 1: EN FUNCIÓN DE LA LONGITUD INTERESTACIÓN")
60
61 for i in djk_path:
62     resultado_final.append(i)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
10.0.0.5 - - [24/Jan/2022 14:45:29] "GET /static/css/ruta2.png HTTP/1.1" 304 -
10.0.0.5 - - [24/Jan/2022 14:45:29] "GET /static/2.png HTTP/1.1" 304 -
10.0.0.5 - - [24/Jan/2022 14:45:29] "GET /static/css/port.png HTTP/1.1" 304 -
Pantitlán L5
Politécnico L5
```

Código escrito para realizar Algoritmo de Dijkstra

Imagen 10

Posteriormente se programó el trazado de las rutas en los mapas correspondientes conforme a las rutas obtenidas con el algoritmo de Dijkstra, de acuerdo con las coordenadas de cada estación tal y como se muestra a continuación.

En la **Imagen No.11** se pueden observar las líneas de código que corresponden a la programación para la obtención del mapa de las rutas generadas por Dijkstra con las características necesarias. Estos mapas generados son guardados en un HTML, para que con su ruta correspondiente sea colocado como enlace de acceso en los iconos pertinentes a “RUTA 1” y “RUTA 2”.

```

print(nueva, "ESTA ES LISTA NUEVA")#lista de longitud y latitud
for i in nueva:
    #print(i)
    num=i

    m=folium.Map(location= num , zoom_start=16)
for i in nueva:

    #icon=folium.Icon(color="black")

    folium.Marker(i, popup="Estación", tooltip="Estacion1").add_to(m)

route = folium.PolyLine(

    nueva, #Conecta los puntos de coordenadas

    peso = 8, # el tamaño de la línea es 8

    color = 'blue', #El color de la línea es naranja

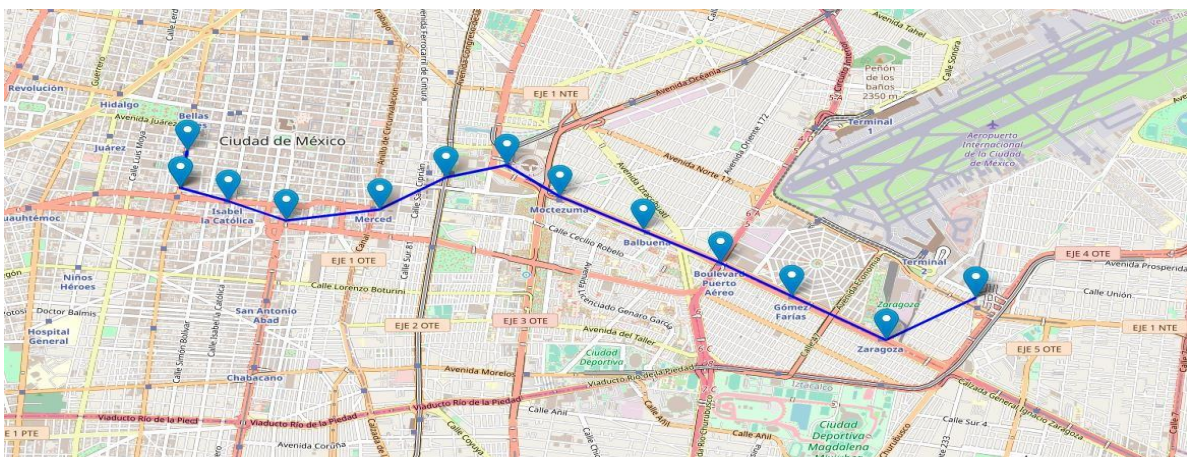
    opacidad = 1, # Transparencia de línea
) .add_to (m) #Agregue esta línea al área m justo ahora

```

Líneas de código para la generación del mapa

Imagen 11

Y en la Imagen No.12 se observa el trazado del mapa con el resultado de los pasos anteriores.



Mapa generado en
OpenStreetMap

Finalmente, se programó la parte de la IA.

El algoritmo necesita obtener los datos teniendo el siguiente orden de importancia o peso de cada parámetro. Se realizan dos principales preguntas para que el algoritmo pueda tomar una decisión para la recomendación de ruta.

1. Tiempo total de ruta aproximado
2. ¿Se viaja en hora pico?

Después la información se termina de complementar con:

3. Número de estaciones con alta afluencia de personas
4. Numero de transbordos posibles

Una vez obtenidos los 4 parámetros anteriores se realizan los cálculos pertinentes en el algoritmo de IA, usando el principio de un perceptrón.

Un perceptrón funciona tomando algunas entradas numéricas junto con lo que se conoce como pesos y un sesgo. Luego multiplica estas entradas con los pesos respectivos (esto se conoce como la suma ponderada). La función de activación toma la suma ponderada y el sesgo como entradas y devuelve una salida final. Con ayuda de este algoritmo se obtienen los porcentajes de satisfacción de usuario, una vez obtenidos los dos porcentajes se hace una comparación entre ambos para que finalmente el sistema indique cual es la ruta más conveniente.

En las siguientes líneas de código mostradas en la Imagen No.13 se muestran los elementos mencionados anteriormente, se configuro el algoritmo con los siguientes pesos, donde entre mayor sea el peso del parámetro se tomará con mayor importancia en los cálculos del algoritmo.

```
entrada, peso = [], [0.8,1,0.7,1.35]
num=[num_dif_lineas,hora_pico2,afluent,tiempo_totalrut_long]
```

Asignación de pesos de las entradas
numéricas.

Alojamiento

La aplicación web que se desarrolló, se encuentra alojada aún en un entorno local incluido en el Framework Flask el cual, proporciona una infraestructura para la realización de pruebas, lo cual permite al desarrollador revisar el correcto funcionamiento y corregir algún posible error que dificulte el funcionamiento de la aplicación.

Por lo que la aplicación se encuentra aún en dicho entorno de prueba y no se encuentra alojada en ningún servidor externo o en alguna solución de almacenamiento en la nube.

4.2 Front End

4.2.1 Interfaz grafica

La Interfaz Gráfica de Usuario permite la interacción de la maquina y las personas, por lo que su diseño, utilidad y manejo representan una parte fundamental para el buen funcionamiento de una aplicación web. Esta interfaz representa todo el trabajo que se hizo al realizar la programación del código backend y permite al usuario la simplificación de tareas, por eso se requiere de una correcta organización y conjunción de imágenes, letra clara, colores, iconos o videos.

Usando HTML5 y CSS (Hojas de Estilo en Cascada) se hizo posible la colocación de imágenes, botones y texto para realizar la búsqueda de las rutas propuestas, así como la consulta de los mapas.

HTML

Todo lo que vemos en Internet está programado con un código interno y cuando accedes a una página web, el navegador lee este código y lo traduce para que veas el diseño que desarrollo el creador.

HTML 5 es la última versión del estándar HTML y nos permite, entre otras funcionalidades, programar aplicaciones web, es compatible con los diseños adaptativos para que se reconozca casi cualquier dispositivo y que se adapte a la web y, por último, es soportado por todos los navegadores.

En el presente proyecto, con HTML5 se desarrolló la estructura para organizar y definir las dimensiones de las secciones para la búsqueda y resultados, también se crearon campos que permiten la introducción de datos por parte de los usuarios. Se agregaron Iconos y textos que ayudan al usuario a identificar cómo se realiza la búsqueda y dónde consultar los mapas.

En la **Imagen No.14** podemos observar la programación para que el usuario inserte la estación de origen y destino, así como la opción para que se pueda elegir si está viajando en hora pico y como el botón que permite calcular la ruta.

```
<label for="horapico">¿Estas viajando en hora pico?</label>
<select class="u-full-width" name="hora_pico">
  <option selected value="0"> Elige una opción </option>
  <option value="SI">Si estoy viajando en hora pico</option>
  <option value="NO">No estoy viajando en hora pico</option>
</select>
<p></p>

<label for="Origen">Punto de origen:</label>
<input
  type="text"
  placeholder="Origen"
  class="u-full-width"
  name="num1"
/>
<p></p>

<label for="Destino">Punto destino:</label>
<input
  type="text"
  placeholder="Destino"
  class="u-full-width"
  name="num2"
/>
<p></p>
```

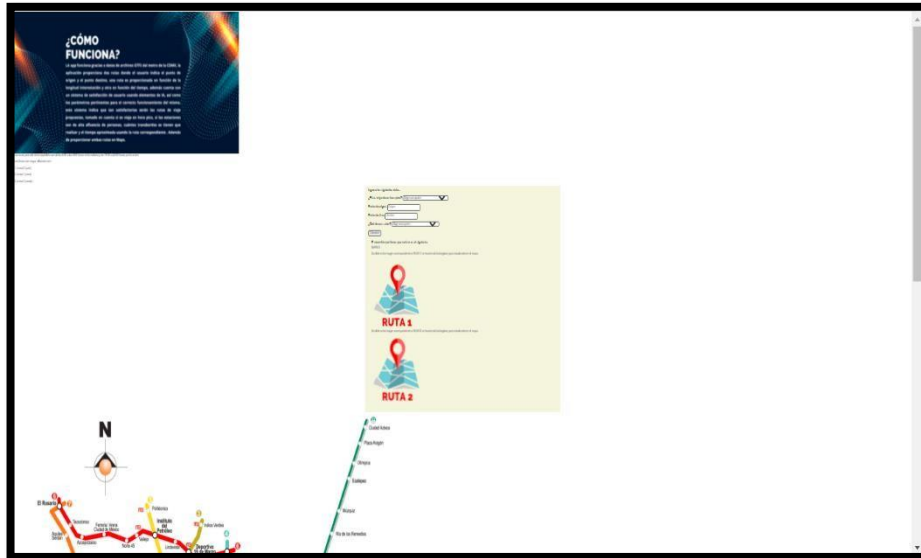
Programación realizada con HTML para insertar origen, destino y hora del viaje

Imagen 14

CSS (Cascading Style Sheets)

Es un lenguaje para manejar el diseño y presentación de las páginas web, de tal forma que HTML se emplea para dar estructura a el contenido de un sitio, mientras que CSS se usa para estructurar su presentación.

En la **Imagen No.15** podemos observar cómo se ve la estructura creada con HTML cuando aún no se incluía la programación con CSS.



Estructura de Interfaz de Usuario con HTML antes de usar CSS.

En la **Imagen No.16** se aprecia la importancia del uso de CSS y cómo influyó para mejorar la presentación de los elementos de la aplicación web.



Estructura de Interfaz de Usuario usando CSS.

En resumen, la aplicación se realizó mediante el uso del Framework Flask Back-End, un entorno para trabajar desde Python, lo que permitió que cada proceso realizado fuera eficiente al utilizar el código de una manera óptima y se tuvo como recursos editores de código, compiladores, entre otras herramientas.

Al momento que el usuario ingresa su punto de origen y destino, estos datos son guardados en variables que la aplicación pasará al algoritmo previamente explicado, para el cálculo de las rutas y sus demás parámetros.

Posteriormente despliega las rutas en renglones, donde primero se observa la ruta basada en la distancia más corta y después, la ruta basada en el menor tiempo. Al final de cada ruta, se refleja el parámetro de satisfacción, así como el tiempo aproximado que tardarías en recorrer la ruta.

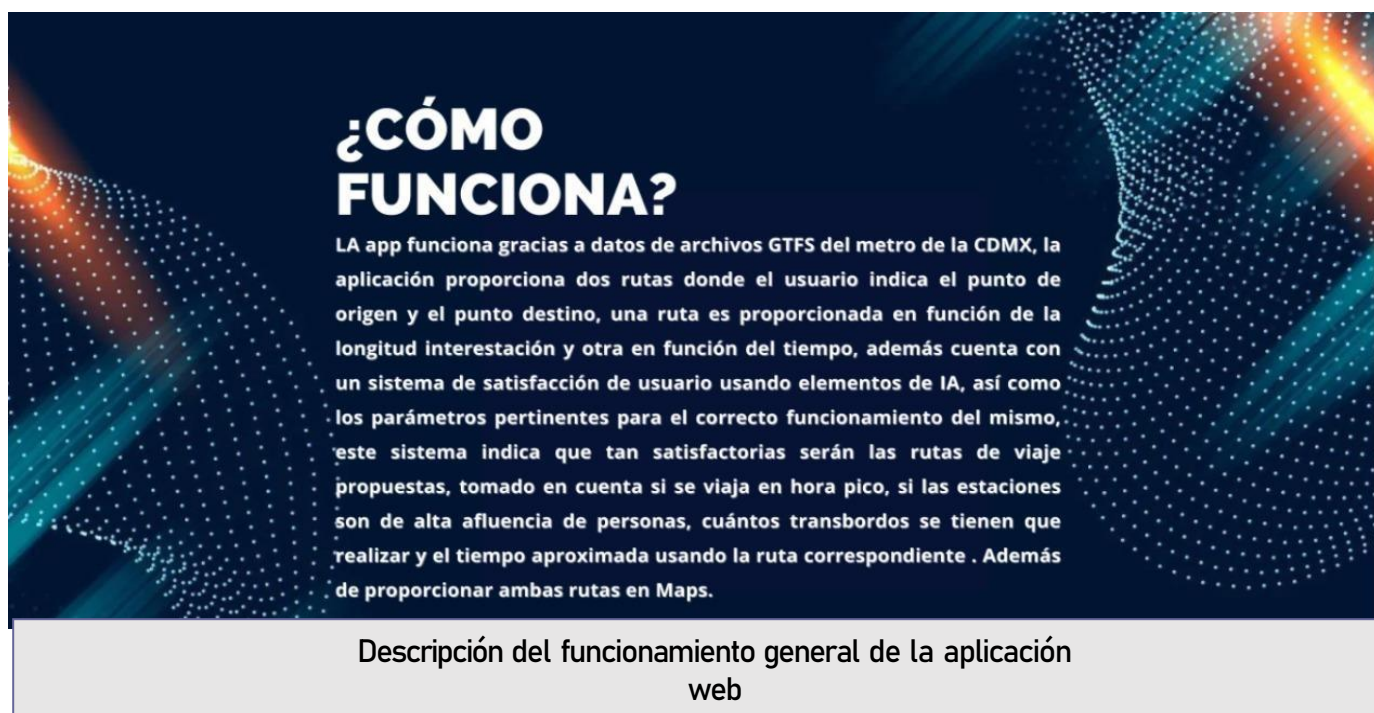
Todos estos procesos que realiza el algoritmo se hacen en el “backend”, el cual es la capa de acceso a los datos de un software, aquí se programan todas las funcionalidades y queda oculto a los ojos del usuario.

Por otro lado, todos los arreglos de diseño que se realizan para personalizar una aplicación web como color de fondo, estilos de letra, animaciones, y diversos recursos gráficos se desarrollan en el “frontend” aquí es donde los usuarios interactúan con la página web. Dentro de este desarrollo se trabaja con los lenguajes HTML y CSS como fue el caso de este proyecto y en algunos otros casos se puede trabajar con JavaScript.

Finalmente, se propuso un nombre para la aplicación: **“INTELIVIAJE”**, ya que se pensó que adoptar un sistema de planeación para realizar un viaje con base en herramientas tecnológicas, es una manera más inteligente de usar los medios de transporte.

4.3 Visualización de la Interfaz de la Aplicación web

Las siguientes imágenes muestran capturas de pantalla de cómo se ve la aplicación terminada hasta el punto de que se ha mencionado donde se observan las rutas calculadas.



Las horas pico del metro capitalino van de las 6:00 a las 9:00 horas en la mañana y de 17:00 a 22:00 horas por la noche.

Las líneas con mayor afluencia son:

1. Línea 2 (azul)
2. Línea 1 (rosa)
3. Línea 3 (verde)

Ingrese los siguientes datos:

¿Estas viajando en hora pico?

Elige una opción

Punto de origen:

Origen

Punto destino:

Destino

Campos para ingresar Origen y Destino

En la **Imagen No.19** podemos resaltar que, en la parte superior, se da una especificación al usuario de cuáles son los horarios “pico” y las líneas con mayor afluencia.

Ingrese los siguientes datos:

¿Estas viajando en hora pico?

Si estoy viajando en hora pico

Punto de origen:

Pantitlán L5

Punto destino:

San Juan de Letrán L8

¿Qué deseas saber?

Recorrido

CALCULATE

Ejemplo de llenado de los campos para el cálculo de ruta

Después de ingresar la información requerida, solo hay que dar clic en **CALCULAR** y observaremos el siguiente resultado:

El recorrido que tienes que realizar es el siguiente:

0	RUTA 1: EN FUNCIÓN DE LA LONGITUD INTERESTACIÓN
0	Pantitlán L5
0	Pantitlán L1
0	Zaragoza L1
0	Gómez Farías L1
0	Boulevard Puerto Aéreo L1
0	Balbuena L1
0	Moctezuma L1
0	San Lázaro L1
0	Candelaria L1
0	Merced L1
0	Pino Suárez L1
0	Isabel la Católica L1
0	Salto del Agua L1
0	Salto del Agua L8
0	San Juan de Letrán L8
0	EL TIEMPO APROXIMADO EN REALIZAR EL RECORRIDO DE LA RUTA 1:
0	24.510714285714286
0	EL PORCENTAJE DE QUE LLEGUES SATISFACTORIAMENTE A TU DESTINO USANDO LA RUTA 1, ES EL SIGUIENTE:

Resultado que arroja con la Ruta 1 basada en la distancia

Imagen 21

0	RUTA 2: EN FUNCIÓN DEL TIEMPO INTERESTACIÓN
0	Pantitlán L5
0	Pantitlán L9
0	Puebla L9
0	Ciudad Deportiva L9
0	Velódromo L9
0	Mixtuhca L9
0	Jamaica L9
0	Chabacano L9
0	Chabacano L8
0	Obrera L8
0	Doctores L8
0	Salto del Agua L8
0	San Juan de Letrán L8
0	EL TIEMPO APROXIMADO EN REALIZAR EL RECORRIDO DE LA RUTA 2:
0	22.59285714285714 Tiempo aproximado
0	EL PORCENTAJE DE QUE LLEGUES SATISFACTORIAMENTE A TU DESTINO USANDO LA RUTA 2, ES EL SIGUIENTE:
0	47.19918501759129
0	EL SISTEMA RECOMIENDA SEGUIR LA RUTA 2:
0	47.19918501759129 Recomendación basada en IA

Resultado que arroja con la Ruta 2 basada en el tiempo.

Imagen 22

En la sección de mapeo, podemos elegir generar el mapa de cualquiera de las dos rutas calculadas.

MAPEO

Da click en la imagen correspondiente a RUTA 1 en función de la longitud, para visualizarla en el mapa:



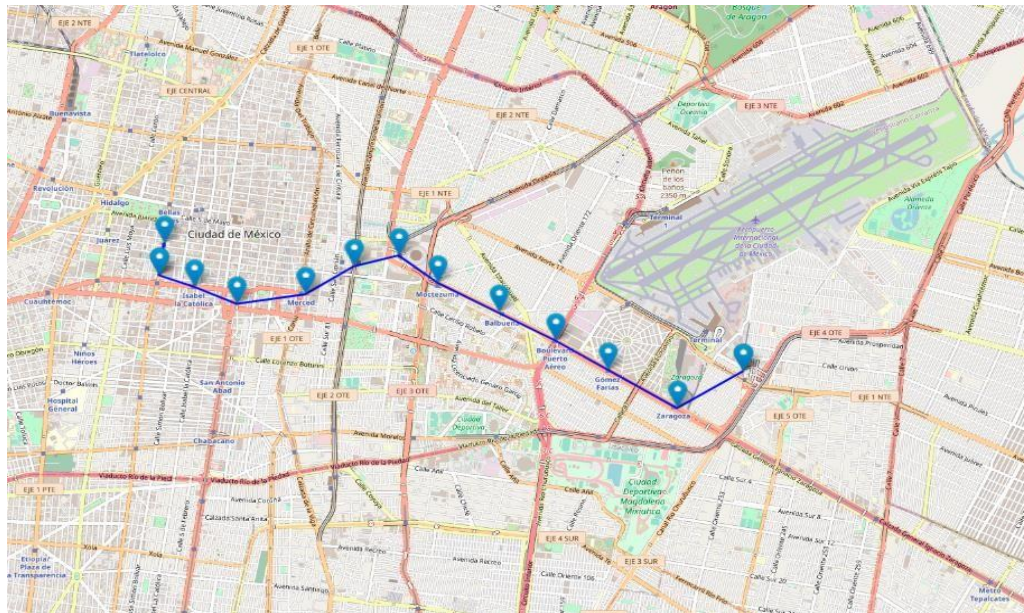
RUTA 1

Da click en la imagen correspondiente a RUTA 2 en función de la longitud, para visualizarla en el mapa:



RUTA 2

Sección que proporciona los mapas de las rutas propuestas.



Mapa resultante de la Ruta 1: Pantitlán-San Juan de Letrán

4.4 Análisis de resultados

Se realizaron diferentes pruebas para analizar el funcionamiento de la aplicación, principalmente se observaron los tiempos arrojados para cada ruta y la generación de los mapas, ya que en varias ocasiones se presentaron errores al delimitar la ruta y estos provenían directamente de un error en los archivos de información al extraer las latitudes y longitudes de cada punto.

Para la comprobación de los tiempos arrojados y debido a la actual situación de contingencia que no nos permitió realizar múltiples viajes en el STC metro, pedimos ayuda a diferentes personas que, por cuestiones laborales, no dejaron de trasladarse, para que midieran los tiempos que tardaban en tomar una u otra ruta propuesta por nuestro algoritmo. Efectivamente el tiempo aproximado era muy cercano a la realidad, sin embargo, el rango de error que se presenta se debe a que los archivos GTFS estáticos no contemplan escenarios de retraso, tal como el simple hecho de que un tren tarde en cerrar las puertas más que otro, esto debido a la cantidad de personas que se aglomeran o puede ser, que se quede parado por alguna cuestión técnica en ese momento. Aun con todo esto la variación de error más grande que se tuvo fue de 10 minutos y la más pequeña de 3 minutos.

También consultamos la eficiencia al elegir estas rutas preguntando si eran buenas y lo que al final se concluye es que las personas que viajan recurrentemente optan por las rutas de menor tiempo, aunque otras personas consideran más importante la distancia, específicamente en los transbordos, ya que algunos son muy largos y generan mucha molestia.

Finalmente, el porcentaje de satisfacción que cada persona calificó para las rutas coincidió en un 70% con el calculado, lo que nos hace pensar que se tienen que ajustar los parámetros en los que se basa el algoritmo de IA para generar un mejor resultado.

5. Conclusión

Respecto al uso del estándar GTFS, fue una excelente fuente de datos que representan un trabajo muy completo de las personas que hicieron posible esta estandarización de la información de los transportes públicos, así como los que contribuyen a mantenerlos actualizados, es de fácil acceso y contiene toda la información necesaria para utilizarlo en diferentes aplicaciones, sin mayor problema. Dentro del desarrollo del proyecto se trabajó con archivos de GTFS estáticos y únicamente con campos obligatorios. Realmente, debido al tiempo, se tuvo que limitar el alcance del proyecto, pero es un proyecto escalable y que se puede retomar para incluir muchos más datos y todos los transportes de la red de movilidad integrada de la Ciudad de México, incluso comenzar a migrar a los archivos dinámicos. Con un mayor alcance se podría observar de mejor forma el funcionamiento de la aplicación, ya que, por el momento, al estar muy limitado, las diferencias entre las valoraciones de las rutas son muy cercanas y con más información, se pueden obtener mejores resultados.

Hablando del funcionamiento de la aplicación web, se puede concluir que cumple su objetivo de poder planear una ruta con anticipación, por lo tanto, es una buena herramienta, fácil de usar y de comprender, es de mucha ayuda para personas que no vivan en la Ciudad de México o que apenas empiezan a transportarse en el sistema de transporte público, se puede incluso acondicionar en otros idiomas o generar recursos para personas con discapacidad para que esté al alcance de la mayoría de las personas.

Sin duda, la aplicación puede mejorar en diferentes aspectos, por ejemplo, en el manejo de la información, ya que como se mencionó antes, se tuvo que delimitar la información para quedarnos con la de un solo transporte, si se hubiera previsto la cantidad de información disponible, hubiera sido un mejor camino crear una base de datos para guardar todos los datos y acceder a ellos de una mejor forma sin hacer pasos extra. También se puede diseñar una aplicación móvil que sea un poco más eficaz, ya que para la aplicación web es necesario consumir más recursos para ingresar al navegador.

Finalmente se lograron tres de los objetivos particulares planteados en un inicio, la difusión de esta información no se logró, debido a que no se realizó la publicación de la aplicación. El objetivo principal si se cumplió ya que se considera que, a pesar de sus funciones limitadas, proporciona información confiable y que la mayoría de las personas podría utilizarla sin mayor complicación. Actualmente en la Ciudad de México no existe alguna aplicación desarrollada aquí, que sea robusta y que las personas reconozcan, sin embargo, la aplicación Moovit de origen israelí ya se encuentra disponible para nuestro país y nos sirvió como una fuente de inspiración de lo que se puede lograr. Consideramos que está en nuestras manos seguir innovando y poner en práctica las habilidades aprendidas hasta hoy.

6. ANEXO

6. 1. Archivos Obligatorios del estándar GTFS estático

Cabe mencionar que las tablas que se presentan a continuación fueron tomadas la página oficial de referencia del estándar GTFS y que cuentan con los requerimientos específicos y oficiales de cada campo, por lo que no se realizó ningún cambio de la información, sólo se organizaron de diferente manera para su fácil comprensión.

Nombre de archivo

Contenido

routes.txt													
Campos:													
route_id	<p>Su contenido está basado en el origen y destino de cada una de las líneas que integran el Sistema.</p> <ul style="list-style-type: none"> ➤ Agencia y numero de ruta correspondiente. <table border="1"> <thead> <tr> <th>Obligatorio</th> <th>Con este campo podremos identificar la ruta</th> </tr> </thead> <tbody> <tr> <td>Condicionalmente obligatorio</td> <td>Es un nombre corto y abstracto que los pasajeros utilizan para identificar la ruta, sin embargo, no proporciona información de su trayectoria.</td> </tr> <tr> <td>Condicionalmente obligatorio</td> <td>Nombre completo de la ruta, generalmente más descriptivo, se debe especificar al menos uno, el corto o el largo o en dado caso, ambos.</td> </tr> <tr> <td>Opcional</td> <td>Describe información útil y confiable de la ruta.</td> </tr> <tr> <td>Obligatorio</td> <td>Indica el tipo de transporte que se usa en la ruta, las opciones disponibles son: <ul style="list-style-type: none"> ➤ 0 -Tranvía, trolebús o tren ligero ➤ 1 - Metro ➤ 2 - Tren ➤ 3 - Autobús ➤ 4 - Transbordador ➤ 5 - Tranvía de cable ➤ 6 - Teleférico ➤ 7 - Funicular ➤ 11 - Trolebús ➤ 12 - Monorriel </td> </tr> <tr> <td>Opcional</td> <td>URL de la página web sobre una ruta específica.</td> </tr> </tbody> </table>	Obligatorio	Con este campo podremos identificar la ruta	Condicionalmente obligatorio	Es un nombre corto y abstracto que los pasajeros utilizan para identificar la ruta, sin embargo, no proporciona información de su trayectoria.	Condicionalmente obligatorio	Nombre completo de la ruta, generalmente más descriptivo, se debe especificar al menos uno, el corto o el largo o en dado caso, ambos.	Opcional	Describe información útil y confiable de la ruta.	Obligatorio	Indica el tipo de transporte que se usa en la ruta, las opciones disponibles son: <ul style="list-style-type: none"> ➤ 0 -Tranvía, trolebús o tren ligero ➤ 1 - Metro ➤ 2 - Tren ➤ 3 - Autobús ➤ 4 - Transbordador ➤ 5 - Tranvía de cable ➤ 6 - Teleférico ➤ 7 - Funicular ➤ 11 - Trolebús ➤ 12 - Monorriel 	Opcional	URL de la página web sobre una ruta específica.
Obligatorio	Con este campo podremos identificar la ruta												
Condicionalmente obligatorio	Es un nombre corto y abstracto que los pasajeros utilizan para identificar la ruta, sin embargo, no proporciona información de su trayectoria.												
Condicionalmente obligatorio	Nombre completo de la ruta, generalmente más descriptivo, se debe especificar al menos uno, el corto o el largo o en dado caso, ambos.												
Opcional	Describe información útil y confiable de la ruta.												
Obligatorio	Indica el tipo de transporte que se usa en la ruta, las opciones disponibles son: <ul style="list-style-type: none"> ➤ 0 -Tranvía, trolebús o tren ligero ➤ 1 - Metro ➤ 2 - Tren ➤ 3 - Autobús ➤ 4 - Transbordador ➤ 5 - Tranvía de cable ➤ 6 - Teleférico ➤ 7 - Funicular ➤ 11 - Trolebús ➤ 12 - Monorriel 												
Opcional	URL de la página web sobre una ruta específica.												
route_short_name													
route_long_name													
route_desc													
route_type													
route_url													

<i>route_color</i>	Opcional	Color de la ruta en función del material disponible para el público. Por default se coloca blanco FFFFFFFF
<i>route_text_color</i>	Opcional	Color del texto, el valor predeterminado es negro 000000
<i>route_sort_order</i>	Opcional	Ordena las rutas para ser presentadas al cliente, los valores más pequeños se muestran primero.
<i>Continuous_pickup</i>	Opcional	Indica si en cualquier punto del recorrido del vehículo se puede abordar
<i>Continuous_drop_off</i>	Opcional	Indica si un pasajero puede descender en cualquier parte del recorrido.

agency.txt

Campos:

agency_id
agency_name
agency_url
agency_timezone
agency_lang
agency_phone
agency_fare_url
agency_email

Su contenido está basado en **identificar a la empresa** que brinda el servicio en determinado transporte público.

Condicionamente obligatorio	Con este campo podremos identificar a la empresa de transporte público, en ocasiones podemos encontrar una empresa que proporcione diversos servicios. Este sólo será obligatorio si un conjunto de datos contiene datos de distintas empresas de transporte público.
Obligatorio	Aquí se coloca el nombre de la empresa.
Obligatorio	URL de la empresa.
Obligatorio	Zona horaria del lugar donde se encuentra la empresa.
Opcional	Idioma principal que usa la empresa de transporte, permite la configuración del idioma para el conjunto de datos.
Opcional	Número de teléfono, puede y debe incluir signos de puntuación para agrupar dígitos, se permite el uso de texto en la marcación
Opcional	URL de la página web donde el usuario podría comprar boletos en línea.
Opcional	Correo electrónico de Atención al Cliente de la empresa

trips.txt

Campos:

<i>route_id</i>
<i>service_id</i>
<i>trip_id</i>
<i>trip_headsign</i>
<i>trip_short_name</i>
<i>direction_id</i>
<i>block_id</i>

Contenido basado en el sentido de la ruta de transporte, variaciones o ramificaciones que puede presentar cada ruta.

Obligatorio	Nos permite identificar la ruta. ID de referencia de <i>routes.route_id</i>
Obligatorio	Permite identificar el conjunto de fechas en las que el servicio está disponible para una o más rutas .ID de referencia de <i>calendar.service_id/calendar_dates.service_id</i>
Obligatorio	Permite identificar un viaje.
Opcional	Contiene los textos que se muestran en la señalización que permite identificar el destino del viaje para los pasajeros. Se puede usar este campo para distinguir diferentes patrones de servicio en la misma ruta. Si la señal de destino cambia durante un viaje <i>trip_headsign</i> puede anularse mediante la especificación de valores para <i>stop_times.stop_headsign</i> .
Opcional	Idioma principal que usa la empresa de transporte, permite la configuración del idioma para el conjunto de datos.
Opcional	Nos indica la dirección de un viaje. Proporciona una manera de diferenciar viajes en función de su sentido al publicar los horarios Las opciones válidas son las siguientes: 0 - Viaje en una dirección (p. ej., viaje de ida) 1 - Viaje en la dirección opuesta (p. ej., viaje de vuelta)
Opcional	Identifica el bloque al que pertenece un viaje.

<i>shape_id</i>	Condicionalmente obligatorio	Identifica una forma geoespacial que describe el recorrido del vehículo para un viaje. Es condicionalmente obligatorio: Este campo es obligatorio si el viaje tiene un comportamiento continuo definido, a nivel de la ruta o del horario de la parada. De lo contrario, es opcional.
<i>wheelchair_accessible</i>	Opcional	Indica si existe acceso para sillas de ruedas. Las opciones válidas son: <ul style="list-style-type: none"> ➤ 0/en blanco - No hay información de accesibilidad para el viaje. ➤ 1 - El vehículo que se utiliza en este viaje en particular puede transportar a, al menos, un pasajero en silla de ruedas. ➤ 2- Ningún pasajero en silla de ruedas puede acceder a este viaje.
<i>bikes_allowed</i>	Opcional	Nos indica si se permite el acceso con bicicletas. Las opciones válidas son las siguientes: <ul style="list-style-type: none"> ➤ 0/en blanco- No hay información sobre el acceso con bicicletas para este viaje. ➤ 1- El vehículo que se usa en este viaje en particular puede transportar a, al menos, una bicicleta. ➤ 2- No se permite el acceso con bicicleta en este viaje.

calendar.csv

Campos:

service_id
Monday
Tuesday Wednesday Thursday Friday Saturday Sunday Start date End date

Su contenido está basado en especificar los días hábiles en que se operan los servicios de transporte.

Obligatorio	Identifica de forma exclusiva un conjunto de fechas en las que el servicio está disponible en una o más rutas. Cada valor <i>service_id</i> puede aparecer como máximo una vez en un archivo calendar.txt.
Obligatorio	Indica si el servicio funciona todos los lunes del período especificado por los campos <i>start_date</i> y <i>end_date</i> . Es importante tener en cuenta que las excepciones para fechas específicas pueden aparecer en calendar_dates.txt. Las opciones válidas son las siguientes: <ul style="list-style-type: none">➤ 1 - El servicio está disponible todos los lunes incluidos en el período.➤ 0 - El servicio no está disponible los lunes incluidos en el período.
Obligatorio	Todos los demás campos se basan en el mismo funcionamiento que el campo "Monday"

stops.csv

Campos:

Su contenido está basado en la ubicación geográfica de las estaciones, así como el nombre con el que suelen conocerse comúnmente.	
stop_id	Obligatorio Permite identificar una parada, una estación o una entrada.
stop_code	Opcional Texto o número que identifica la ubicación para los pasajeros, normalmente utilizados en sistemas de información de transporte público para teléfonos
stop_name	Condicionally obligatorio. Muestra el nombre de la ubicación, y es obligatorio para las ubicaciones que son paradas, estaciones o entradas y salidas de estaciones.
stop_desc	Opcional Muestra una breve descripción de la ubicación.
stop_lat	Condicionally obligatorio. Indica la latitud de la ubicación. Obligatorio para ubicaciones que son paradas, estaciones o entradas y salidas de estaciones.
stop_lon	Condicionally obligatorio. Indica la longitud de la ubicación. Obligatorio para ubicaciones que son paradas, estaciones o entradas y salidas de estaciones.
zone_id	Condicionally obligatorio. Señala la zona tarifaria de una parada. Es obligatorio si se proporciona información de las tarifas mediante fare_rules.txt de lo contrario es opcional.
stop_url	Opcional Contiene URL de una página web sobre la ubicación
location_type	Opcional Muestra el tipo de ubicación: parada, estación, entrada/salida área de abordaje o nodo genérico.
parent_station	Condicionally obligatorio. Establece una jerarquía entre las diferentes ubicaciones y es obligatorio para las ubicaciones que son entradas, nodos genéricos o áreas de embarque.
Wheelchair_boarding	Opcional Indica si es posible acceder en silla de ruedas en esa ubicación.

Stop_times.txt

Campos:

<i>Trip_id</i>
<i>Arrival_time</i>
<i>departure_time</i>
<i>stop_id</i>
<i>stop_sequence</i>
<i>Stop_headsign</i>
<i>Pickup_type</i>
<i>drop_off_type</i>
<i>Continuous_pickup</i>
<i>continuous_drop_off</i>

Contenido basado en los horarios en el que los vehículos llegan y salen de paradas individuales en cada viaje.

Obligatorio	Identifica un viaje
Obligatorio	Su función principal es identificar la hora de llegada a cierta parada en un viaje particular. Si no existen horarios diferentes para la llegada y salida de una parada se ingresa el mismo valor para <i>arrival_time</i> y <i>departure_time</i> . El tipo de paradas programadas en donde el vehículo respeta los horarios de llegada y salida son puntos temporales
Obligatorio	Permite indicar la hora de salida de una parada específica, si no existen horarios diferentes para la llegada y salida en una parada, se debe ingresar el mismo valor tanto para <i>arrival</i> como para <i>departure</i> time. Debe mostrar claramente los valores de tiempo que sean posibles, así como horarios estimados.
Obligatorio	Permite identificar una parada de servicio.
Obligatorio	Permite identificar el orden de las paradas en cierto viaje. Estos valores deben aumentar a lo largo del viaje.
Opcional	Contiene el texto que identifica el destino del viaje para los pasajeros.
Opcional	Señala el método de recogida de los pasajeros, permitiendo conocer si existen recogidas disponibles, o recogidas habituales y demás.
Opcional	Esto indica el método para dejar pasajeros. Mediante parada habitual, o si no existen paradas para dejar pasajeros disponibles, etc.
Opcional	Indica si en cualquier punto del recorrido un pasajero puede subir al vehículo de transporte.
Opcional	Indica si en cualquier punto del recorrido un pasajero puede descender del vehículo de transporte.

6.2. Archivos opcionales del estándar GTFS estático.

Estos archivos nos darán información complementaria que puede o no incluirse al usar o crear los archivos GTFS.

<i>calendar_dates.txt</i>	En este archivo se encuentra de manera detallada las excepciones de los servicios definidos en el archivo calendar.txt, en determinado caso de que calendar.txt sea omitido, calendar_dates.txt debe ser incluido y contener todas las fechas de los diferentes servicios del transporte público.
<i>fare_attributes.txt</i>	Contiene información de las tarifas de las distintas rutas que se ofrecen en el transporte público.
<i>fare_rules.txt</i>	En este archivo se definen las reglas para aplicar las tarifas del archivo fare_attributes, normalmente depende de tres rangos: <ul style="list-style-type: none">• Origen y destino• Zonas que atraviesa el itinerario• Ruta que utiliza el itinerario
<i>shapes.txt</i>	Las formas que describen un viaje están contenidas en este archivo, estas formas se representan por una secuencia de puntos que se recorren en orden. No es necesario que las formas corresponden exactamente a la ubicación de las paradas, pero todas las paradas de viaje deben estar a una distancia cercana de la forma de ese viaje.
<i>frequencies.txt</i>	Este archivo contiene los intervalos regulares o tiempos de viaje que contemplan los servicios de transportes y pueden ser representados dos tipos de servicio: <ul style="list-style-type: none">• Servicio basado en frecuencia, donde no se sigue un horario fijo en el día, sin embargo, los operadores del transporte respetan los intervalos determinados para los viajes.

	<ul style="list-style-type: none"> • Servicio basado en un horario, aquí se tiene intervalos estrictos e iguales para los diferentes viajes durante un cierto periodo, los operadores respetan estrictamente ese horario.
<i>tr transfers.txt</i>	<p>Cuando se calcula un itinerario, las aplicaciones que usan los archivos GTFS interpolan transbordos en función del tiempo y la proximidad de la parada. Las reglas adicionales se incluyen en este archivo.</p>
<i>feed_info.txt</i>	<p>Aquí se incluye información del conjunto de datos que se publica ya que en algunos casos es una entidad diferente quien los publica a las empresas que los usan. Este archivo es obligatorio solo si se proporciona el archivo de translations.txt.</p>

7. Referencias Electrónicas

[1] [International Telecommunication Union. \(s. f.\). Grupo Temático sobre Ciudades Inteligentes y Sostenibles. ITU.](https://www.itu.int/es/ITU-T/focusgroups/ssc/Pages/default.aspx)

<https://www.itu.int/es/ITU-T/focusgroups/ssc/Pages/default.aspx>

[2] Buschecker. (s. f.). Buschecker. Recuperado agosto de 2021, de

<http://www.buschecker.com/default.aspx>

[3] Moovit. (s. f.). Que camión me lleva | Enrútate | Transporte público. Misitio. Recuperado septiembre de 2021, de

<https://www.enrutate.info>

[4] y [5] Moovit Inc. (s. f.). Moovit, la primera aplicación de movilidad urbana del mundo. Moovit. Recuperado agosto de 2021, de

<https://moovit.com/es/>

[6] General Transit Feed Specification. (s. f.). GTFS.

<https://gtfs.org/>

[7] General Transit Feed Specification. (s. f.-b). GTFS.

<https://gtfs.org/reference/static/#field-definitions>

[8] General Transit Feed Specification. (s. f.-c). GTS Reference. Recuperado septiembre de 2021, de

<https://gtfs.org/reference/static/#file-requirements>

[9] Welcome to python. (s. f.). Python.Org. Recuperado octubre 2021, de

<https://www.python.org/about/>

[10] NetworkX. (2014). Software para Redes Complejas agosto 2021, de

<https://networkx.org/>

[11] OpenStreetMap. (s.f). OpenStreetMap. Recuperado noviembre 2021, de <https://www.openstreetmap.org/about>