

PROYECTO DE CALCULO VECTORIAL

***Desarrollo de un programa
ejecutable para optimizar el
desplazamiento de los estudiantes
en el horario de clases utilizando
cálculo vectorial***

Integrantes:

- ✓ David Sumba
- ✓ Angelo Zurita
- ✓ Gabriel Delgado
- ✓ Luis Romero

Contenido

Objetivos General:	3
Objetivos Específicos:	3
Metodología	4
Definición del problema	5
Marco Teórico	6
Procedimiento	14
Beneficios:	16
Resultados:	18

Objetivos General:

El objetivo de este proyecto es desarrollar un programa ejecutable que utilice conceptos y técnicas del cálculo vectorial para optimizar el desplazamiento de los estudiantes en el horario de clases, reduciendo el tiempo que deben caminar entre aula y aula. El proyecto busca mejorar la eficiencia y comodidad de los estudiantes al moverse dentro del campus universitario.

El proyecto busca mejorar la eficiencia y comodidad de los estudiantes al moverse dentro del campus universitario.

Objetivos Específicos:

1. Recopilar información sobre el campus universitario y los horarios de clases, incluyendo ubicación de aulas, rutas de desplazamiento y horarios de cada asignatura.
2. Estudiar y aplicar conceptos y técnicas del cálculo vectorial, como el cálculo de distancias, rutas óptimas y minimización de tiempos de desplazamiento.
3. Desarrollar un programa ejecutable utilizando un lenguaje de programación adecuado, que utilice los datos recopilados y los conceptos del cálculo vectorial para calcular las rutas óptimas de desplazamiento entre aulas en función de los horarios de clases.
4. Construir una maqueta del campus universitario que represente fielmente las ubicaciones de las aulas y las rutas de desplazamiento, utilizando tecnologías de impresión 3D y materiales adecuados.

5. Realizar pruebas y ajustes en el programa ejecutable para garantizar su precisión y confiabilidad en la optimización del desplazamiento de los estudiantes.
6. Establecer indicadores de evaluación para medir la eficiencia del programa ejecutable, como la reducción del tiempo promedio de desplazamiento entre aulas.
7. Preparar material informativo y visualmente atractivo que explique el funcionamiento del programa ejecutable, la construcción de la maqueta y los beneficios del proyecto en términos de optimización del desplazamiento estudiantil.

Metodología

1. **Investigación bibliográfica:** Estudiar conceptos y técnicas del cálculo vectorial aplicables a la optimización del desplazamiento estudiantil, así como métodos de programación y algoritmos para el desarrollo del programa ejecutable.
2. **Recopilación de datos:** Obtener información detallada sobre el campus universitario, incluyendo la ubicación de las aulas, horarios de clases y rutas de desplazamiento.
3. **Desarrollo del programa ejecutable:** Utilizar un lenguaje de programación adecuado para desarrollar el programa ejecutable que calcule las rutas óptimas de desplazamiento entre aulas en función de los horarios de clases.
4. **Pruebas y ajustes:** Realizar pruebas exhaustivas del programa ejecutable, utilizando casos de prueba con diferentes horarios de clases y evaluar su eficacia en la optimización del desplazamiento estudiantil.
5. **Evaluación de resultados:** Analizar los resultados obtenidos por el programa ejecutable y evaluar la eficiencia en términos de reducción del tiempo promedio de

desplazamiento entre aulas.

6. **Preparación de la presentación:** Elaborar material informativo, visualmente atractivo y accesible para presentar el proyecto, explicando el funcionamiento del programa ejecutable, la construcción de la maqueta y los beneficios de la optimización del desplazamiento estudiantil.

Definición del problema

En la Escuela Superior Politécnica del Litoral (ESPOL), se ha identificado una situación en la que algunos estudiantes tienden a tomar rutas más largas para llegar a sus clases. Esta circunstancia puede deberse a la falta de conocimiento acerca de rutas más cortas y eficientes en el campus, así como a la posible recepción de información incorrecta o poco precisa.

El objetivo fundamental de este proyecto es abordar este problema y proporcionar a los estudiantes herramientas y recursos para que puedan encontrar las rutas más cortas y óptimas hacia sus clases. Esto no solo reduciría el tiempo que los estudiantes emplean para desplazarse, sino que también les permitiría llegar puntualmente a sus clases, optimizando así su experiencia académica y su eficiencia en el manejo del tiempo.

En este contexto, se plantea la necesidad de desarrollar una solución que integre tecnologías modernas y enfoques de optimización de rutas. Esta solución podría tomar la forma de una aplicación móvil o una plataforma en línea dedicada. A través de esta herramienta, los estudiantes tendrían la posibilidad de ingresar sus horarios de clases y ubicaciones de inicio, lo que permitiría a la plataforma calcular de manera automática y precisa las rutas más cortas hacia sus destinos.

El proyecto involucra varias etapas esenciales:

- **Recopilación de Datos:** Se requeriría la obtención y sistematización de información geoespacial, incluidos los mapas del campus, la distribución de aulas y las calles circundantes. Esto proporcionaría la base necesaria para calcular las rutas.
- **Desarrollo de la Plataforma:** La creación de una plataforma intuitiva y amigable es crucial. Los estudiantes deberían poder ingresar sus datos de manera sencilla, y la plataforma debería proporcionar información clara y detallada sobre las rutas recomendadas.
- **Algoritmos de Optimización:** La implementación de algoritmos de optimización, permitiría determinar las rutas más cortas en función de los datos ingresados. Estos algoritmos considerarían factores como la distancia, el tiempo estimado y cualquier posible restricción, como caminos peatonales o calles cerradas.

Marco Teórico

El siguiente proyecto se construirá alrededor de la Escuela Superior Politécnica del Litoral (ESPOL), prestigiosa institución educativa ubicada en el campus Gustavo Galindo, km 30. 5 vía Perimetral Guayaquil - Ecuador. 2269016 - 2269017, reconocida por su excelencia en la educación superior y su enfoque en la formación de profesionales altamente calificados en múltiples disciplinas. Establecida el 29 de octubre de 1958, la ESPOL ha jugado un papel fundamental en el panorama académico y científico de 1 y ha sido fundamental en el desarrollo social, tecnológico y económico de la región.

ESPOL se distingue por su compromiso con la educación de calidad, la investigación de punta y el enfoque en la innovación. Con una amplia gama de programas académicos, la

universidad ofrece oportunidades educativas en áreas que van desde ingeniería y ciencias naturales hasta humanidades y ciencias sociales. Su enfoque interdisciplinario y la colaboración con la industria y el sector público han construido una reputación de formación de profesionales capaces de enfrentar los desafíos del mundo moderno.

La ESPOL también se distingue por su campus, que cuenta con modernas instalaciones, laboratorios de última generación y espacios diseñados para apoyar el aprendizaje, la investigación y la interacción de los estudiantes. Además de su compromiso con la excelencia académica, la institución valora el enriquecimiento cultural y la participación comunitaria al patrocinar una amplia gama de actividades y eventos extracurriculares que enriquecen la experiencia de los estudiantes.

En el marco de esta investigación, la ESPOL se convierte en el escenario de referencia ya que sus propiedades y dinámicas internas juegan un papel fundamental en la identificación y solución del problema de optimización de los movimientos estudiantiles.

El proyecto se basa en conceptos y técnicas de cálculo vectorial, así como en algoritmos de optimización y programación.

El lenguaje de programación que usaremos en este proyecto es Python. Python es un lenguaje de programación altamente interpretado que generalmente se usa para desarrollar aplicaciones y software. Fue desarrollado por Guido van Rossum a fines de la década de 1980 y fue muy popular debido a su buena legibilidad y su sintaxis simple y clara. Python es conocido por centrarse en la legibilidad del código, lo que hace que los programas sean más fáciles de escribir y comprender, lo que a su vez fomenta el desarrollo de aplicaciones rápido y eficiente. La optimización de rutas es el proceso de decisión de ruta más eficiente

en términos de variables (costo, tiempo, etc.) es más complicado que simplemente encontrar el camino más corto entre dos puntos. Se deben tener en cuenta todos los factores relevantes, como el número y la ubicación de las paradas necesarias en el camino y el tiempo de entrega.

Otros factores son:

- El número de giros o vueltas en la ruta
- Giros a la izquierda (cruzando la línea de movimiento)
- El mejor conductor o el más cercano que se puede enviar en la ruta
- Volumen de tráfico para la hora actual
- El mejor acceso a la estación a lo largo de la ruta Las diversas opciones de ruta pueden sumar rápidamente.

Los algoritmos de optimización incluyen el algoritmo de Dijkstra, que según Carvalho, el desarrollado por E.W. Dijkstra, conocido como el algoritmo de Dijkstra, es una de las técnicas para calcular la ruta de menor costo entre los nodos de un diagrama. Dado un nodo de inicio denominado A como punto de partida (por ejemplo, la ciudad de inicio en la red de rutas), este algoritmo calcula la distancia más corta entre ese nodo y los otros nodos del diagrama, en este caso la segunda ciudad de destino.

El proceso comienza con una estimación inicial de la distancia mínima, inicialmente asumida como infinita (∞). Luego, el algoritmo ajusta gradualmente esta estimación de distancia. Una ciudad se considera "cerrada" si ya se ha encontrado la distancia mínima entre la ciudad seleccionada como ciudad de origen y esta ciudad. Con cada iteración del algoritmo, las distancias estimadas se examinan y ajustan a medida que se examinan los

diversos enlaces entre los nodos.

El algoritmo de Dijkstra es una herramienta esencial para resolver problemas de optimización de rutas, ya que permite encontrar la ruta más corta entre dos puntos considerando los costos asociados a las conexiones entre nodos. Este método es ampliamente utilizado en la planificación de trayectos, logística y otras aplicaciones donde se busca determinar la ruta más eficiente o económica en términos de distancia o tiempo.

El algoritmo de Dijkstra (E.W. Dijkstra) es uno de los algoritmos que calcula el camino de coste mínimo entre los vértices de un grafo. Una vez elegido el vértice A como raíz de la búsqueda (ciudad origen), este algoritmo calcula la distancia mínima desde este vértice a todos los demás vértices del grafo, es decir, las ciudades restantes. Este algoritmo parte de una estimación inicial de la distancia mínima, que se considera infinita (∞). considerada infinita (∞), y va ajustando sucesivamente esta distancia. Considera que una ciudad estará "cerrada" cuando ya se haya obtenido un camino de distancia mínima desde la ciudad tomada como origen de la búsqueda hasta ella. (2008, pág. 2)

El metodo que vamos a utilizar para este proyecto es la formula de Haversine, la cual según Dauni La Fórmula Haversine establece sus propias reglas fundamentadas en la consideración de la Tierra como una esfera en todas sus ecuaciones, omitiendo cualquier influencia derivada de la leve elipticidad terrestre (factor elipsoidal). Esta particularidad se presenta como un caso excepcional dentro de un marco más amplio de una fórmula general en trigonometría esférica, la cual guarda una intrincada relación con los distintos lados y

ángulos que conforman un triángulo trazado en la superficie esférica de la Tierra. Como resultado de este enfoque, el cálculo de la distancia entre dos puntos en la superficie terrestre se ve modificado por una cierta medida de curvatura que refleja las características geométricas únicas de la esfera terrestre.

La Fórmula Haversine tiene su propia ley, en la que todas las ecuaciones se basan en la forma de una Tierra esférica, eliminando el factor de que la Tierra es ligeramente elíptica (factor elipsoidal). Esto es un caso especial de una fórmula general en trigonometría esférica, que está relacionada con los lados y ángulos de un triángulo esférico. El cálculo de la distancia entre un punto y otro en la superficie de la Tierra se ve afectado por un cierto grado de curvatura. (2019, pág.4)

Para poder implementar la formula de haversine, primero debemos de saber que es un algoritmo de búsqueda, ya que con ello podremos implementar la formula de Haversine. Los algoritmos de búsqueda son métodos y técnicas utilizadas para encontrar soluciones en un espacio de posibilidades, como un grafo, una matriz o una estructura de datos, con el objetivo de encontrar un estado o nodo objetivo que cumpla con ciertas condiciones. Estos algoritmos son ampliamente utilizados en campos como la inteligencia artificial, la optimización y la resolución de problemas en general.

Los algoritmos de búsqueda se aplican en situaciones donde tienes un conjunto de opciones o estados posibles y deseas encontrar una secuencia de acciones o decisiones que te lleven desde un estado inicial a un estado objetivo. Algunos ejemplos de aplicaciones de

algoritmos de búsqueda incluyen:

1. **Ruta más corta en un mapa:** Encontrar la ruta más corta entre dos ubicaciones en un mapa, como encontrar el camino más rápido para llegar de un lugar a otro en una ciudad.
2. **Resolución de laberintos:** Encontrar una salida en un laberinto, ya sea mediante la búsqueda de todos los caminos posibles o utilizando algoritmos más eficientes.
3. **Planificación de rutas para robots:** Encontrar una ruta segura y eficiente para que un robot navegue a través de un entorno desconocido sin chocar con obstáculos.
4. **Resolución de juegos:** Encontrar la mejor secuencia de movimientos en juegos de estrategia como el ajedrez o el Go para maximizar las posibilidades de ganar.
5. **Planificación de tareas:** Encontrar la mejor secuencia de tareas para minimizar el tiempo total o el costo en un programa de producción.

Otros factores a considerar al implementar la fórmula de Haversine son la latitud y la longitud, en pocas palabras, latitud y longitud, que son coordenadas geográficas que se utilizan para determinar la ubicación de un punto en la superficie terrestre.

- **Latitud:** Esta es la distancia angular medida al norte o al sur del ecuador terrestre. La latitud se mide en grados, minutos y segundos y varía de 0° en el ecuador a 90° en los polos norte y sur.

- **Longitud:** esta es la distancia angular medida al este o al oeste del primer meridiano, el meridiano de Greenwich en Londres, Reino Unido. La longitud también se mide en grados,

minutos y segundos y varía desde 0° en el Primer Meridiano hasta 180° Este y 180° Oeste.

grados de latitud y longitud juntos forman un sistema de coordenadas que le permite identificar de manera única cualquier lugar en la tierra. Son ampliamente utilizados en navegación, cartografía, sistemas de posicionamiento global (GPS) y diversas aplicaciones relacionadas con la cartografía y localización precisa de lugares del planeta.

Nuestro proyecto se basa en la optimización de rutas en conjunto con el cálculo vectorial, al implementar estos dos métodos nos referimos a la aplicación de conceptos y técnicas de cálculo vectorial para resolver problemas relacionados con la planificación y optimización de rutas, como en los campos de la logística, el transporte y la navegación.

El cálculo de vectores y la optimización de rutas tienen mucho en común debido a su intersección en la resolución de problemas de tráfico, espacio y planificación eficiente.

Algunos de los enlaces más importantes son:

1. Representación de cantidades dirigidas: El cálculo vectorial y la optimización de trayectorias se basan en la idea de trabajar con cantidades que tienen una magnitud y una dirección. En el cálculo vectorial, los vectores representan magnitudes con direcciones específicas, mientras que, en la optimización de rutas, los vectores pueden representar velocidades, aceleraciones y direcciones de movimiento en diferentes momentos y en diferentes lugares.

2. Modelado de trayectorias: el cálculo de vectores y la optimización de rutas consisten en modelar trayectorias y movimientos en el espacio. El cálculo vectorial proporciona las herramientas matemáticas necesarias para describir los cambios en las magnitudes de los vectores a lo largo del tiempo o entre diferentes puntos en el espacio. La optimización de

ruta utiliza estas descripciones para encontrar la ruta óptima entre dos puntos minimizando o maximizando una función objetivo como la distancia, el tiempo o el costo.

3. Derivadas y gradientes: En ambas áreas, las derivadas y gradientes tienen una importancia fundamental. En cálculo vectorial, las derivadas vectoriales se calculan para describir tasas de cambio en múltiples direcciones. En la optimización de rutas, los gradientes se pueden calcular para encontrar la dirección en la que la función objetivo aumenta o disminuye más rápidamente, guiando así el proceso de búsqueda de rutas óptimas.

4. Variación de parámetros: El cálculo vectorial y la optimización de trayectorias resuelven problemas donde una o más cantidades varían dependiendo del parámetro. En cálculo vectorial, pueden ser funciones vectoriales que dependen de un parámetro como el tiempo. En la optimización de rutas, los parámetros pueden ser límites de carretera, velocidad, tiempos, etc.

5. Aplicaciones a problemas del mundo real: el cálculo vectorial y la optimización de rutas encuentran aplicaciones prácticas en situaciones del mundo real, como planificación de rutas, navegación GPS, logística, distribución y planificación urbana. Ambas disciplinas le permiten tomar decisiones informadas para lograr resultados más efectivos y eficientes.

En resumen, el cálculo vectorial proporciona las herramientas matemáticas para describir y analizar el movimiento y las cantidades direccionales, mientras que la optimización de rutas utiliza estas descripciones para encontrar trayectorias óptimas en función de diferentes objetivos. La combinación de estas dos áreas permite abordar problemas complejos de planificación y movimiento con precisión y eficiencia.

Una vez analizados los conceptos del cálculo vectorial y la optimización de rutas , se puede decir que la planificación efectiva de rutas no solo facilita que las personas se desplacen de manera directa, sino que también minimiza el tiempo necesario para llegar a su destino. En el ámbito universitario, la optimización de rutas presenta una serie de beneficios significativos. Estos enfoques mejorados no solo agilizan los desplazamientos, sino que también aportan mejoras integrales a la experiencia universitaria. Entre los múltiples beneficios que la optimización de rutas aporta a la universidad, se destaca la optimización de la eficiencia y la gestión del tiempo.

Procedimiento

Como se menciona anteriormente utilizaremos la fórmula de haversine, para lo cual seguiremos los siguientes pasos:

1. **Importar bibliotecas:** Primero, debes importar las bibliotecas necesarias. Para realizar cálculos de distancias utilizando la fórmula de Haversine, necesitarás las funciones trigonométricas de la biblioteca math y, posiblemente, también alguna biblioteca para manejar estructuras de datos como heapq o queue para los algoritmos de búsqueda.

```
2. import dash
3. import dash_bootstrap_components as dbc
4. import os
5. import numpy as np
6. import plotly.express as px
7. import plotly.graph_objects as go
8. from scipy.interpolate import interp1d
9. import pandas as pd
10. import math
```

11. **Definir la fórmula de Haversine:** La fórmula de Haversine te permite calcular la

distancia entre dos puntos geográficos en base a sus latitudes y longitudes. Aquí tienes una función que implementa la fórmula de Haversine:

```
python Copy code

def haversine_distance(lat1, lon1, lat2, lon2):
    R = 6371 # Radio de la Tierra en kilómetros
    dlat = math.radians(lat2 - lat1)
    dlon = math.radians(lon2 - lon1)
    a = math.sin(dlat / 2) ** 2 + math.cos(math.radians(lat1)) * math.cos(m
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    distance = R * c
    return distance
```

12. **Definir los nodos y las conexiones:** Define tus ubicaciones como nodos en un grafo y las conexiones entre ellas como arcos ponderados por la distancia calculada con la fórmula de Haversine.

```
python Copy code

nodes = {
    "A": (lat_A, lon_A),
    "B": (lat_B, lon_B),
    # Agrega más nodos aquí
}

connections = {
    ("A", "B"): haversine_distance(lat_A, lon_A, lat_B, lon_B),
    ("A", "C"): haversine_distance(lat_A, lon_A, lat_C, lon_C),
    # Agrega más conexiones aquí
}
```

13. **Implementar el algoritmo de búsqueda:** Puedes utilizar el algoritmo A* o el algoritmo de Dijkstra para encontrar la ruta más corta entre dos nodos. Para el algoritmo A*, también necesitarás una función heurística que estime la distancia

restante entre el nodo actual y el nodo objetivo.

python

Copy code

```
def a_star(start, goal):
    open_list = [(0, start)] # Cola de prioridad con la distancia acumulada
    came_from = {} # Diccionario para rastrear el camino recorrido

    while open_list:
        current_distance, current_node = heapq.heappop(open_list)

        if current_node == goal:
            # Reconstruir el camino desde "goal" hasta "start" usando "came_from"
            path = [current_node]
            while current_node in came_from:
                current_node = came_from[current_node]
                path.append(current_node)
            path.reverse()
            return path
```

```
        for neighbor, distance in connections.items():
            if neighbor[0] == current_node:
                new_distance = current_distance + distance
                heapq.heappush(open_list, (new_distance, neighbor[1]))
                came_from[neighbor[1]] = current_node

    return None # No se encontró un camino

# Llamar a la función para encontrar la ruta más corta
shortest_route = a_star("A", "B")
```

Beneficios:

- Ayudar a los Estudiantes a no Perderse:

Mejora de la Orientación: La optimización de rutas no solo implica encontrar la ruta más corta, sino también proporcionar indicaciones claras y detalladas a los

estudiantes. Esto los ayudará a familiarizarse con el campus y a encontrar ubicaciones específicas sin dificultad.

Minimización de la Confusión: Al reducir la posibilidad de perderse, los estudiantes experimentarán menos frustración y ansiedad al explorar áreas desconocidas del campus. Esto facilita su integración a la vida universitaria.

- **Encontrar la Ruta Menos Poblada:**

Reducción del Estrés: Optar por rutas menos concurridas puede brindar a los estudiantes una experiencia de desplazamiento más relajada. Evitar multitudes y congestiones puede reducir la sensación de agobio y facilitar un ambiente más tranquilo.

- **Encontrar la Ruta Más Directa entre Edificios:**

Ahorro de Tiempo: La ruta más directa no solo implica distancia reducida, sino también menos interrupciones en el camino. Los estudiantes podrán optimizar su tiempo y enfocarse en sus actividades académicas y extracurriculares.

Mayor Eficiencia Energética: La ruta más corta y directa implica menos consumo de recursos, como combustible para vehículos o energía para el desplazamiento a pie. Esto contribuye a la sostenibilidad y la responsabilidad ambiental.

- **Optimización del Recorrido Académico:** La optimización de rutas no solo implica el desplazamiento entre edificios, sino también la planificación eficiente de la secuencia de clases. Esto permitirá a los estudiantes minimizar la distancia y el tiempo entre clases sucesivas, facilitando la organización de su horario académico.
- **Incremento en la Puntualidad:** Al encontrar rutas más eficientes, los estudiantes podrán llegar puntualmente a sus clases y actividades extracurriculares. Esto

mejorará su asistencia, participación y aprovechamiento de la experiencia educativa.

- **Mejora en la Satisfacción Estudiantil:** La capacidad de moverse rápidamente y de manera eficiente en el campus contribuirá a una experiencia estudiantil más placentera y sin contratiempos. Los estudiantes estarán menos propensos a sentirse abrumados por la prisa o la frustración debido a desplazamientos largos e ineficientes.
- **Reducción del Estrés:** La optimización de rutas reducirá el estrés asociado con la posibilidad de llegar tarde a las clases y actividades. Los estudiantes podrán disfrutar de un entorno más tranquilo y concentrarse en sus estudios y actividades académicas
- **Mejora de la Planificación Logística:** La optimización de rutas no solo beneficia a los estudiantes, sino también a profesores, personal administrativo y visitantes al campus, mejorando la organización general de las actividades en la universidad.

Resultados:

Un proyecto de optimización de la movilidad en un campus universitario utilizando la fórmula de Haversine y el concepto de cálculo vectorial ha logrado con éxito sus objetivos generales y específicos. Gracias a la investigación bibliográfica, la recopilación de datos y el desarrollo de un programa ejecutable, fue posible mejorar la eficiencia y la comodidad del movimiento de los estudiantes dentro de la instalación.

Mejora de la eficiencia y la comodidad: un programa ejecutable desarrollado con la fórmula de Haversine y algoritmos de optimización que calculó las rutas óptimas entre clases en función de los horarios. Como resultado, el tiempo promedio que los estudiantes deben

pasar entre clases ha disminuido significativamente, mejorando su experiencia en el campus y haciéndoles más fácil llegar a tiempo a las actividades académicas.

1. Mejor orientación: Al incluir direcciones paso a paso en las rutas calculadas, los estudiantes podrían familiarizarse con el campus y encontrar ubicaciones específicas más fácilmente. Esto redujo el riesgo de que los estudiantes se perdieran y disminuyó la frustración de navegar por un territorio desconocido.

2. Reducción del estrés: al encontrar rutas menos concurridas y más directas, los estudiantes experimentan menos estrés cuando viajan. Evitar las multitudes y los atascos de tráfico ha ayudado a crear un entorno de viaje más relajado y tranquilo.

3. Optimización del horario de lecciones: el programa ejecutable también ayuda a optimizar el orden de las lecciones para que los estudiantes puedan minimizar el tiempo y el espacio entre lecciones consecutivas. Esto ha simplificado la planificación y mejorado la eficiencia de su viaje académico.

4. Mayor puntualidad y asistencia: la optimización de rutas ha llevado a una mayor puntualidad en clase y actividades extracurriculares. Esto mejoró la asistencia de los estudiantes, la participación y el disfrute del aprendizaje.

En resumen, el proyecto demostró cómo el uso de la fórmula de Haversine y los conceptos de cálculo vectorial pueden tener un impacto significativo en la mejora de la eficiencia y la comodidad de los estudiantes en un campus universitario. La combinación de tecnologías de programación, datos recopilados y herramientas de visualización ha dado como resultado un enfoque integral que beneficia a toda la comunidad universitaria. Las lecciones aprendidas del proyecto pueden servir como base para futuras mejoras en la planificación

de viajes al campus y la optimización de otras áreas logísticas dentro de la instalación.



Bibliografía

¿Qué es Optimización de Ruta? | Verizon Connect México. (s. f.). Verizon Connect.

<https://www.verizonconnect.com/mx/glosario/que-es-optimizacion-de-ruta/#:~:text=La%20optimizaci%C3%B3n%20de%20rutas%20es,m%C3%A1s%20corto%20entre%20dos%20puntos>

Qué es la logística y para qué sirve - Ferrovial. (2022, 22 septiembre). Ferrovial.

<https://www.ferrovial.com/es/recursos/logistica/#:~:text=La%20log%C3%ADstica%20es%20la%20actividad,la%20forma%20m%C3%A1s%20eficiente%20posible.>

Stewart, J. (2015). Cálculo vectorial. Cengage Learning. 2. CORMEN, T. H., LEISERSON, C.

E., Rivest, R. L., & Stein, C. (2009).

Introduction to Algorithms. MIT Press. 3. Downey, A. (2019).

Think Python: How to Think Like a Computer Scientist. O'Reilly Media. 4.

Tufte, E. R. (1990). Envisioning information. Graphics Press.

Carvalho, B. M. (2008). *Universidade de Coimbra*. Obtenido de

<https://student.dei.uc.pt/~brunomig/cp/Artigo.pdf>

Dauni, P., Firdaus, M. D., Asfariani, R., Saputra, M. I. N., Hidayat, A. A., & Zulfikar, W.

B. (2019, December). Implementation of Haversine formula for school location tracking. In *Journal of Physics: Conference Series* (Vol. 1402, No. 7, p. 077028).

IOP Publishing.