

### **Situacion problematica:**

La palabra "universidad" proviene del latín "universitas", que significa "el conjunto de todas las cosas". En la universidad, ingenieros, economistas, artistas, médicos y profesionales de todas las disciplinas existentes comparten un espacio común. Lamentablemente, dentro de esta comunidad diversa, existen desigualdades, una de las cuales se refiere a la movilidad. Sin embargo, dado que la universidad es un lugar inclusivo, hemos decidido poner a prueba qué tan bien está diseñada nuestra universidad ICESI para atender las necesidades de esta comunidad variada.

Por todo lo anterior, en este proyecto, se realizará la modelación de los pisos de todos los edificios de la universidad (así como lugares de interés como cafeterías, bibliotecas, coliseos, entradas, etc.) para garantizar su accesibilidad a través de rampas o ascensores. El objetivo es facilitar a las personas con dificultades de movilidad la identificación de todas las rutas accesibles desde su ubicación actual hasta el lugar deseado, así como la más corta de ellas.

Se modelarán los siguientes espacios:

- J\_1
- A\_2P2
- D1\_P3
- Tecnoquimicas\_1
- N\_2
- B\_5
- F\_1
- G\_1
- Ascensor\_M\_2P4
- Canchas\_3
- Central\_1
- M\_3
- Elevador\_D2
- A\_4
- Elevador\_E2P4
- Snack\_1
- Elevador\_E1P5
- Elevador\_D2P2
- B\_1
- Elevador\_E1P3
- Elevador\_C1P2
- C1\_P3
- Elevador\_L1P3
- B\_6
- A\_1P2
- Biblioteca\_1
- Elevador\_L2P2
- Bristo\_2
- I\_2

- 122\_1
- Ascensor\_M\_1P3
- B\_4
- C\_1
- Tecnoquimicas2\_P2
- C\_2
- Elevador\_E2P2
- Canchas\_1
- A\_1
- Elevador\_L2P5
- Ascensor\_M\_1P4
- Tecnoquimicas\_2
- C\_4
- N\_1
- Coliseo2\_2
- Biblioteca\_4
- Elevador\_L1P4
- Coliseo2\_3
- B1\_P2
- Ascensor\_M\_1P2
- Elevador\_D1
- L\_2
- Elevador\_D2P4
- D\_3
- Bristo\_1
- Cancha\_2
- K\_1
- B2\_P2
- Ascensor\_M\_2P2
- E\_3
- 122\_2
- Elevador\_D2P3
- Auditorios\_3
- G\_4
- Wonka\_1
- I\_1
- Elevador\_E2P5
- Biblioteca\_3
- Zona\_Fumar
- Biblioteca\_2
- Elevador\_E1P2
- Tecnoquimicas1\_P2
- Cancha\_1
- A\_2

- Central\_3
- Snack\_4
- Ascensor\_M\_2P3
- Elevador\_E1
- Zona\_Compostaje
- D\_2
- Elevador\_D1P2
- G\_5
- F\_2
- Elevador\_L1
- Auditorios\_2
- B\_3
- Snack\_5
- A\_4
- Auditorios\_1
- Lateral\_1
- Torre\_1
- Elevador\_E2P3
- L\_3
- Snack\_3
- E\_1
- Piscinas\_1
- G\_3
- C1\_P2
- D\_1
- M\_1
- M\_1P2
- E\_2
- Elevador\_L2P4
- Bristo\_3
- Elevador\_D1P3
- Tecnoquimicas\_4
- Elevador\_E1P4
- Ascensor\_M\_1
- D\_4
- Cancha\_3
- Elevador\_D1P4
- Snack\_2
- G\_2
- Canchas\_2
- Coliseo2\_1
- C\_3
- Ascensor\_M\_2
- M\_4

- M\_2
- Gym\_1
- Central\_2
- B\_2
- H\_1
- Elevador\_E2
- Tecnoquimicas\_3
- L\_1
- Elevador\_L2P3
- Elevador\_C1
- Elevador\_C1P3
- Elevador\_L2
- Elevador\_L1P2
- Elevador\_L1P5.

### **Paso 1: Identificación del problema:**

Se requiere una solución que modele todos los caminos existentes entre estos lugares, pero solo los cuales sean accesibles por medio de rampas o ascensores, como el tiempo es importante para todos, es necesario que la solución pueda encontrar el camino más corto de un lugar inicial hasta un lugar deseado, hallar su peso y permitir encontrar el MST del grafo.

Caso 1: Un estudiante con discapacidad motora que debe utilizar silla de ruedas para desplazarse desea conocer cual es el camino mínimo para desplazarse entre dos puntos de la universidad Icesi.

Caso 2: El mismo estudiante ha utilizado la ruta que le ha indicado el programa muy seguido, pero, quiere conocer cuanto tiempo tarda en recorrerla.

Caso 3: Un guardia de seguridad ha sufrido un accidente automovilístico, lo cual, lo ha dejado inmóvil de la cintura para abajo, sin embargo, desea seguir realizando su trabajo de patrullar la universidad, pero requiere realizar el esfuerzo mínimo, por ello, requiere conocer cuál es el árbol de expansión mínima de la universidad (sin utilizar escaleras).

### **Paso 2: Recopilación de información:**

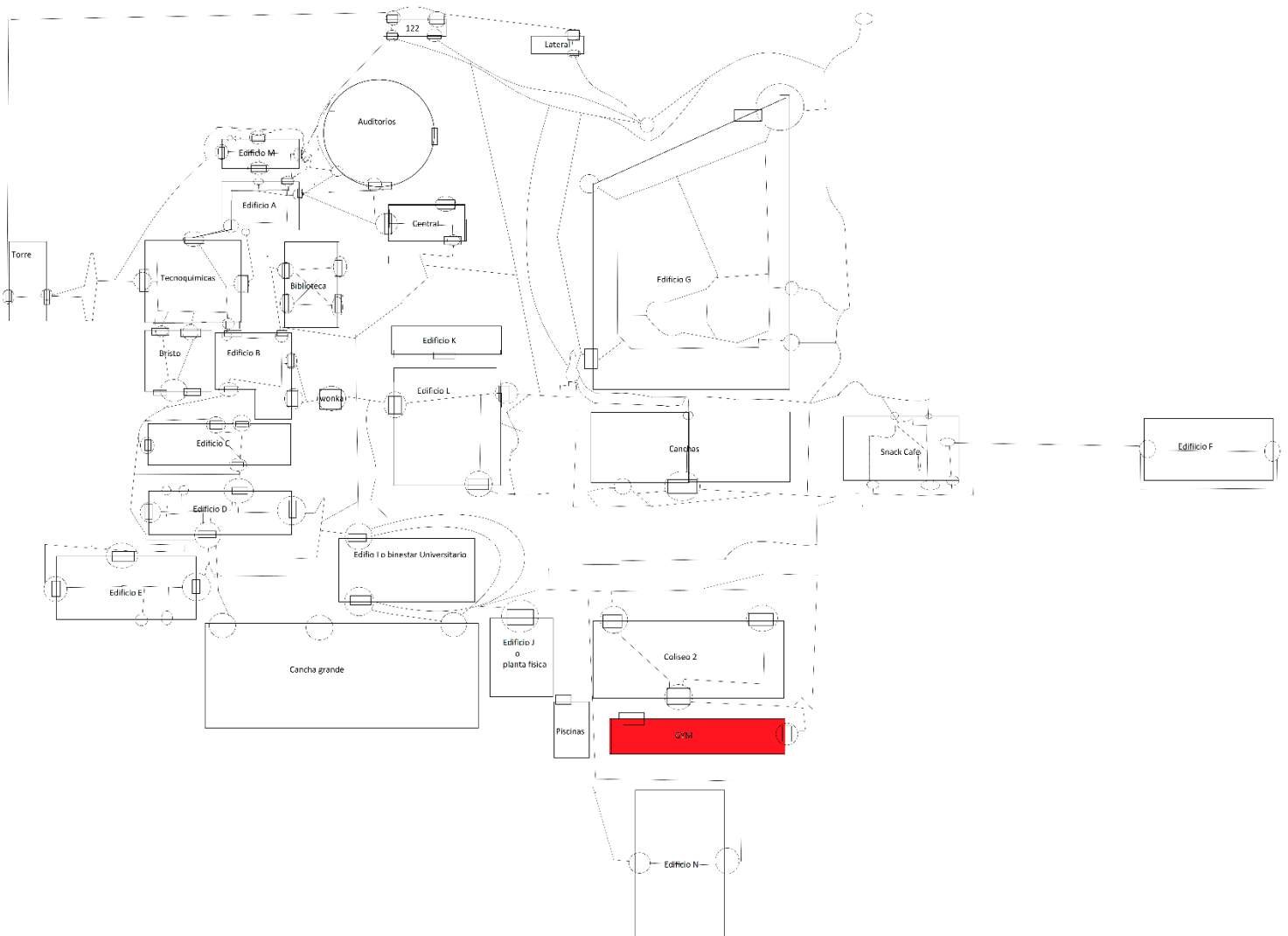
Para iniciar el proceso de recopilación de información, se acudió en primera instancia al departamento de planeación académica, lugar encargado de gestionar todos los espacios físicos de la universidad. Se esperaba obtener información precisa sobre los lugares de la institución, incluyendo las rampas y ascensores que pudieran existir en ellas. Sin embargo, lamentablemente, dicho departamento no poseía la información requerida.

Por ello, se decidió dirigirse a la oficina de planta física de la universidad, en busca de información más detallada. Allí, se nos informó que contaban con la información que se necesitaba, pero que era necesario seguir un protocolo específico para obtenerla. El proceso implicaba hacer una petición formal desde la dirección del programa, la cual debía pasar por el profesor, posteriormente la secretaría, para

luego poder solicitar un informe de los lugares, rampas y ascensores disponibles. Sin embargo, el proceso era largo y tenía una baja probabilidad de éxito en el corto plazo.

Tomando en cuenta estos aspectos y considerando el limitado tiempo disponible para la investigación, se decidió realizar un mapeo de manera manual. Así, se logró obtener la información necesaria para cumplir con el objetivo. El resultado de este proceso se materializó en un grafo que permitió visualizar de manera clara y precisa la distribución de los lugares de la universidad, junto con las rampas y ascensores existentes.

El grafo resultante fue el siguiente:



Img. Se tiene un grafo donde los **vértices** representan los edificios y áreas de interés dentro de la universidad, y las **aristas** representan los caminos por los cuales se puede desplazar, ya sea utilizando una rampa o un ascensor, entre esos puntos. Estas conexiones están ponderadas con un **peso** que indica la cantidad de segundos que se tarda en llegar de una zona a otra.

Ya que se tiene modelada la universidad y sus conexiones con la condicional previamente planteada, se requiere encontrar información sobre grafos y el cómo hallar los caminos más cortos, el resultado de la investigación es el siguiente:

### **Grafo:**

En las ciencias de la computación, un grafo es una estructura de datos no lineal compuesta por vértices (también llamados nodos) y aristas. Los nodos representan entidades individuales, y las aristas son las conexiones que existen entre ellos. Puede ser utilizado para representar relaciones y conexiones entre diferentes elementos, donde las aristas pueden estar ponderadas con pesos que indican alguna medida de distancia, costo o tiempo.

### **Camino Mínimo:**

En teoría de grafos, el problema del camino mínimo consiste en encontrar la forma de ir de un nodo (vértice) a otro de tal manera que la suma de los pesos de sus respectivas conexiones (aristas) sea menor o igual que el peso de cualquier otro camino entre el mismo par de vértices.

Los algoritmos que se usarán para resolver este problema son los siguientes:

### **Algoritmo de Dijkstra:**

Creado por Edger W. Dijkstra, en 1956, este algoritmo permite encontrar, a través del peso de las aristas, el camino más corto entre un vértice dado y todos los otros vértices del grafo en cuestión con la condición de que los pesos no sean negativos.

### **Algoritmo de Floyd-Warshall:**

El algoritmo de Floyd Warshall permite encontrar el camino más corto entre todos los pares de nodos o vértices de un grafo, y, al igual que el algoritmo de Dijkstra, no se puede emplear en grafos con pesos negativos.

### **Algoritmo de Prim:**

El algoritmo de Prim permite encontrar el árbol recubridor mínimo (MST) de un grafo no dirigido, pesado y conexo. Este algoritmo construye el árbol partiendo desde cualquier vértice del grafo, donde, atraviesa un nodo más de una vez para obtener la distancia mínima.

### **Algoritmo de Kruskal:**

El algoritmo de Kruskal tiene como objetivo construir un árbol de expansión mínima (MST) en un grafo conexo no dirigido. Comienza a construir el árbol a partir del nodo con menor peso del grafo, atravesando un nodo una sola vez, y cumpliendo con la particularidad de poder generar bosques (componentes desconectados).

## **Paso 3: Búsqueda de Ideas Creativas**

### **Problema del camino mínimo:**

- Opción 1: Para el desarrollo de la solución que satisfaga la problemática planteada, se creará un sistema en java que funcione a través de la consola, donde, se le muestre al usuario todos los puntos de interés (vértices) que puede elegir, y que, de manera manual, escriba el vértice del cual quiere partir y el vértice destino. Para que así, el programa imprima todos los lugares por los cuales tiene que pasar para llegar a su destino. Para este diseño se utilizarán los algoritmos de Dijkstra y Floyd Warshall.
- Opción 2: La implementación que se llevará a cabo para resolver el problema propuesto es construir un sistema utilizando interfaz gráfica en JavaFx, donde, se mostrará un mapa modelado de la universidad con todos los puntos y zonas de interés (nodos). El usuario podrá seleccionar uno de los puntos (oprimiendo click sobre él) el cual será el punto de partida y después seleccionará el punto de destino, generando así un camino mínimo entre ambos vértices que será mostrado en el mapa. Además, tendrá la opción de cambiar la vista entre pisos, así, si elige un punto que está en un piso diferente, se construirá el camino mínimo que lo lleve hacia una rampa o ascensor que lo dirija a su destino. Para este diseño se utilizarán los algoritmos de Dijkstra y Floyd Warshall.

### **Problema del árbol de expansión mínimo:**

- Opción 1: Para el desarrollo de la solución de la problemática planteada se utilizará una interfaz gráfica hecha en JavaFx, que, al oprimir un botón, mostrará el mapa de la universidad con el árbol de expansión mínima dibujado, partiendo desde el vértice que tenga la arista de peso mínimo del grafo, haciendo uso del algoritmo de Prim.

- Opción 2: La implementación que se llevará a cabo para resolver el problema es construir un código utilizando interfaz grafica en JavaFx, donde, al seleccionar un botón, se mostrará el MST del grafo en cuestión, haciendo uso del algoritmo de Kruskal e implementando adicionalmente una estructura de conjuntos disjuntos.

### **Problema de hallar el peso del camino mínimo y el árbol de expansión mínimo:**

- Opción 1: Para cubrir las necesidades generadas por el problema, se utilizará el mapa de la universidad construido en JavaFx, donde, al buscar un camino mínimo entre dos nodos o construir el árbol de expansión mínimo, se mostrará automáticamente el peso de la arista en cuestión o si es el MST, de todas las aristas, para ello, se utilizará el algoritmo de BFS y el de Kruskal.
- Opción 2: Para el desarrollo de la solución de este problema, a partir del mapa de la universidad construido en JavaFx, posteriormente a encontrar el camino mínimo entre dos vértices o el MST se desbloqueará un botón que permitirá ver el peso (tiempo en segundos entre dos vértices) del camino descubierto o de todo el MST, para lograr este objetivo se usará el algoritmo de Dijkstra y el de Prim.

### **Opciones descartadas:**

#### **Problema del camino mínimo:**

- Opción 1: Esta opción es descartada, porque, a pesar de representar una posible solución que satisfaga el problema en cuestión, resultaría muy tedioso y confuso para el usuario el tener que estar viendo constantemente la lista de todas las zonas de interés ponderadas en el grafo y tener que elegir las escribiendo su nombre.

#### **Problema del árbol de expansión mínimo:**

- Opción 2: Aunque esta opción sea una solución viable para el sistema, se descarta, debido a que, para hacer uso del algoritmo de Kruskal en el grafo en cuestión se tendría que implementar una estructura de conjuntos disjuntos. Y con el ejercicio, se busca hallar una solución efectiva pero que a su vez no represente un gran costo de tiempo para el grupo de trabajo.

### **Problema de hallar el peso del camino mínimo y el árbol de expansión mínimo:**

- Opción 1: Esta alternativa se descarta inmediatamente, puesto que, el algoritmo de BFS no cumple su propósito en grafos ponderados.



### Rúbrica para evaluar ideas:

- Soluciona el problema: La idea implementación soluciona el problema totalmente.
- Usabilidad: La idea implementada es de fácil usabilidad, es decir, es entendible para el usuario.
- Mantenibilidad: La idea implementada es de fácil mantenibilidad.
- Reutilización: La idea implementada permite la fácil reutilización del código para otros posibles contextos del problema.
- Uso de estructuras adecuadas: La idea implementada usa las mejores estructuras adecuadas para el desarrollo de la solución.

La evaluación se realiza asignando un puntaje del 1 al 5 en cada ítem, siendo 1 la calificación más baja y 5 la más alta.

### Problema del camino mínimo:

Opción	Soluciona el problema	Usabilidad	Mantenibilidad	Reutilización	Uso de estructuras adecuadas	Total
Opción 1	5	1	2	4	5	17
Opción 2	5	5	4	4	5	<b>23</b>

Según la evaluación, la mejor opción es la 2.

### Problema del árbol de expansión mínimo:

Opción	Soluciona el problema	Usabilidad	Mantenibilidad	Reutilización	Uso de estructuras adecuadas	Total
Opción 1	5	5	4	5	5	<b>24</b>
Opción 2	5	4	4	5	4	22

Según la evaluación, la mejor opción es la 1.

### Problema de hallar el peso del camino mínimo y el árbol de expansión mínimo:

Opción	Soluciona el problema	Usabilidad	Mantenibilidad	Reutilización	Uso de estructuras adecuadas	Total
Opción 1	1	4	4	2	2	13
Opción 2	5	4	4	4	5	<b>22</b>

Según la evaluación, la mejor opción es la 2.

**Fuentes:**

- [GeeksforGeeks. \(2023\). Graph Data Structure And Algorithms. GeeksforGeeks.](#)
- [Explained: Graphs. \(2012, 17 diciembre\). MIT News | Massachusetts Institute of Technology.](#)
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
- [algoritmo floyd warshall \[Grafos - software para la construcción, edición y análisis de grafos.\]. \(s. f.\).](#)
- [Greyrat, R. \(2022, 5 julio\). Diferencia entre el algoritmo de Prim y Kruskal para MST – Barcelona Geeks](#)
- [Árboles de peso mínimo: algoritmos de Prim y Kruskal — Matemáticas Discretas para Ciencia de Datos. \(s. f.\).](#)