**MoviSimple, la odisea de los seis puntos**

**Enunciado**

En MoviSimple no hay Uber ni Cabify: aquí los conductores son héroes que recorren un mapa con sólo seis vértices y un objetivo claro: llevar al pasajero del punto A al punto B en el menor tiempo posible… o al menos intentarlo antes de que llegue el siguiente cliente con otra petición de pizza.

Cada equipo creará una aplicación web/CLI que simule este escenario, usando la clase GraphSimple para modelar el grafo de seis nodos y aplicar Dijkstra “simple”. El pasajero podrá registrarse, autenticarse y elegir origen y destino. La app calculará la ruta mínima en segundos, mostrará un progreso animado y revelará el costo del viaje, para luego reiniciar la interfaz y esperar al siguiente aventurero.

**Requerimientos**

1. Registro/Autenticación

* Pantalla de registro (nombre, correo, contraseña).
* Pantalla de login (correo, contraseña).
* Usuarios guardados en users.txt, una línea por cada usuario.

1. Carga del grafo

* Definir las 9 aristas en el código (o en edges.txt) para 6 nodos.
* Cada tripleta [u, v, w] constituye una conexión bidireccional con peso w (segundos).

1. Selección de ruta

* Tras el login, mostrar un mapa con 6 vértices numerados.
* El usuario elige origen y destino; ambos nodos cambian de color al seleccionarse.

1. Cálculo y animación

* Ejecutar dijkstraSimple(source) para obtener distancias y predecesores.
* Reconstruir la ruta mínima y dibujarla sobre el mapa.
* Mostrar un indicador o barra de progreso que avance entre nodos, respetando el peso en segundos.

1. Costo y limpieza

* Cada equipo define una tarifa por segundo (por ejemplo, $0.50/segundo).
* Mostrar “Tiempo total: T segundos” y “Costo: T × tarifa”.
* Al completar la animación, reiniciar origen, destino y ocultar animación.

1. Despliegue

* La aplicación debe estar desplegada en Vercel con URL pública funcional.

Criterios de Aceptación

| ID | Criterio |
| --- | --- |
| CA1 | El sistema permite registrar un usuario y almacena sus datos en users.txt. |
| CA2 | El sistema autentica al usuario existente con correo y contraseña. |
| CA3 | Tras el login, el mapa con 6 vértices se muestra correctamente. |
| CA4 | El usuario puede seleccionar origen y destino; los nodos seleccionados cambian de color. |
| CA5 | El grafo se construye con las aristas definidas y pesos en segundos. |
| CA6 | El algoritmo Dijkstra “simple” calcula distancias correctas desde el origen. |
| CA7 | La ruta óptima se dibuja y se muestra una animación o progress bar acorde a los pesos. |
| CA8 | Se calcula y muestra el costo del viaje según la tarifa definida por el equipo. |
| CA9 | Al finalizar el viaje, la interfaz vuelve al estado inicial para una nueva consulta. |
| CA10 | La aplicación está desplegada en Vercel con una URL pública accesible y sin errores de despliegue. |

Rúbrica de Evaluación

| Ítem | Puntos |
| --- | --- |
| Registro/Login y persistencia | 20 |
| Implementación de GraphSimple | 25 |
| Cálculo de ruta (Dijkstra) | 25 |
| Animación / Progress bar | 15 |
| Cambio de color en nodos seleccionados | 10 |
| Cálculo y presentación de costo | 10 |
| Limpieza de interfaz final | 5 |
| Despliegue en Vercel | 10 |

Mockups de la Interfaz

1. Pantalla de Registro: campos Name, Email, Password y botón Register.
2. Pantalla de Login: campos Email, Password y botón Log in.
3. Interfaz de Solicitud de Viaje:

* Mapa simplificado con 6 nodos numerados.
* Selects o clic para elegir Origen y Destino (nodos cambian de color al seleccionarse).
* Botón Calculate Route y barra de progreso animada.

