Java Code Challenge

Nos gustaría tener un servicio web RESTful que almacene información de caminos y estaciones (en memoria está bien) para optimización de viajes.

Una estación esta definida por un nombre. Mientras que un camino, se encuentra definido por un costo, una estación origen, y otra estación destino (bidireccional). El objetivo es ofrecer una solución para consultar sobre el camino óptimo para transitar desde una estación A, hasta una estación B minimizando costos.

1. Por favor, completar en Spring Boot (Java o Kotlin) y en no mas de 2 días consecutivos.
2. Completar el proyecto en Github, para que podamos revisar el código.
3. No usar SQL.

Requerido:

- Tests de integración.
- Aplicación dockerizada.
- Java 11 o superior.
- Claridad del código.
- Correctitud en diseño de arquitectura.

Se valorará positivamente:

- Uso de TDD.
- Desarrollo incremental de la solución mediante el uso de commits.
- Aplicación de los principios SOLID.
- Documentacion.

Especificación del servicio

PUT /stations/$station_id

```
Body: { "name":string }
```

Codigo 1: PUT station

En dónde:

• **station_id** Es de tipo ’long’ identificador de una nueva estación. •

name Es de tipo ’string’ espcificando el nombre de la estación.

PUT /paths/$path_id

```
Body: { "cost":double, "source_id":long, "destination_id":long }
```

Codigo 2: PUT path

En dónde:

- **path_id** Es de tipo ’long’ identificador de un nuevo camino.
- **source_id** Es de tipo ’long’ espcificando el id de la estacion origen.
- **destination_id** Es de tipo ’long’ espcificando el id de la estacion destino.
- **cost** Es de tipo ’double’ espcificando el costo de transitar el camino.

Asumir que no se va a registrar mas de un camino entre dos estaciones y que todos los caminos son bidireccionales.

GET /paths/$source_id/$destination_id

```
Returns:

{ "path": [long], "cost" : double }
```

Codigo 3: GET shortest path

Los ids de las estaciones que conforman el camino optimo en base al valor del "cost" desde source_id hasta destination_id junto al valor del costo total.

Algunos ejemplos simples podrian ser:

```
PUT /stations/10 { "name": "Barcelona" } => { "status": "ok" }
PUT /stations/11 { "name": "Paris" }      => { "status": "ok" }
PUT /stations/12 { "name": "Berlin" }     => { "status": "ok" }
PUT /stations/13 { "name": "Roma" } => { "status": "ok" }


PUT /paths/1 { "cost": 50, "source_id": 10, "destination_id": 11 } => { "status": "ok" } PUT
/paths/2 { "cost": 100, "source_id": 10, "destination_id": 12 } => { "status": "ok" }
PUT /paths/3 { "cost": 60, "source_id": 10, "destination_id": 13 } => { "status": "ok" }
PUT /paths/4 { "cost": 20, "source_id": 13, "destination_id": 12 } => { "status": "ok" }


GET /paths/10/13

      => { "path": [10, 13], "cost": 60}
GET /paths/12/11
  => { "path": [12, 13, 10, 11], "cost": 130}
```

Codigo 4: Ejemplos