

Tabla de contenido

1. Objetivo.....	2
2. Descripción.....	3
3. Requerimientos.....	5
4. Entrega	6

1. Objetivo

El objetivo de la prueba es poder evaluar los siguientes criterios de programación:

- Conocimientos en proyectos spring, énfasis en spring-boot.
- Estructura de datos.
- Cumplimiento de los requisitos descritos.
- Calidad y uso de Unit Tests.

Duración aproximada: 3-4 horas

2. Descripción

Se proporciona un fichero zip que contiene un proyecto spring-boot:

- **arvato-candidates-spring-boot.zip**

El proyecto consiste en implementar una aplicación web que pueda calcular distancias entre ciudades con un vehículo automóvil.

Los cálculos se deben realizar a partir de las estructuras de datos que se proporcionan dentro del package resources, en formato json:

- **data/distances.json**
- **data/cities.json**

Distances:

Las distancias contienen los siguientes metadatos:

- Los nombres de ciudades representados como origen y destino.
- La distancia entre ambas en kilómetros y opcionalmente se añade la distancia con peajes.

Ejemplo:

```
{
    "origin": "Barcelona"
    "destination": "Madrid"
    "distance" : {
        "toll" : 564.55
        "free" : 600.34
    }
}
```

Cities:

Las ciudades se representan mediante una coordenada GPS (latitud y longitud) en grados decimales. Se facilita el radio de la ciudad (en km) para poder establecer si una coordenada GPS pertenece a la ciudad calculando el perímetro a través del radio.

Las coordenadas GPS proporcionadas en el fichero cities.json representan el centro de la ciudad.

```
{
    "name" : "Barcelona",
    "lat" : 40.7127837,
    "lon" : -74.0059413,
    "radius" : 5
}
```

Pautas

- Se deben usar las estructuradas de datos proporcionadas, con los valores *demo*.
- No es necesario añadir nuevas ciudades y distancias.
- No es necesario proporcionar un frontend para testear las operaciones.
- No es necesario realizar cálculos con APIs de terceros.
- Se facilita una clase de utilidad *GeoUtil.java* para el cálculo de distancia según las coordenadas gps.

3. Requerimientos

Basándonos en el módulo REST ofrecido por spring.

(<https://spring.io/guides/gs/rest-service/>)

Se requiere implementar:

- 1) Cargar todas las estructuras de datos en memoria al iniciar la aplicación.
- 2) Crear un endpoint para calcular la distancia entre dos ciudades. Las coordenadas de entrada se recibirán en formato GMS (*grados, minutos, segundos y orientación*), tanto para la latitud como longitud. Opcionalmente se recibirá un parámetro de entrada para si la distancia se calcula con o sin peajes.

La respuesta será un texto plano que indicará el número de kilómetros entre ambas ciudades.

Nota: Si las ciudades no pueden ser calculadas directamente entre origen y destino, se deberían recorrer las ciudades intermedias. (*ejemplo: Barcelona – Tarragona no es una ruta directa, pero si se puede calcular si pasamos primero por Lleida, debe cubrir todas las combinaciones posibles*)

- 3) Crear un endpoint para mostrar todas las ciudades ordenadas alfabéticamente cargadas en memoria.
- 4) Crear un endpoint para mostrar todas las rutas cargadas actualmente en memoria. (no se requiere ordenación). Debe permitir dos parámetros opcionales para poder filtrar por “origen” o “destino” de la ciudad.
- 5) Crear un endpoint que reciba como parámetro una ciudad de entrada y devuelva todas las rutas posibles ordenadas de menor a mayor proximidad, Además, opcionalmente recibirá un parámetro denominado “máxima distancia a filtrar”, el cual servirá para **no mostrar las rutas** más lejanas a dicho parámetro de entrada (ejemplo: 100 km).

Opcionalmente se valorará el uso de las siguientes implementaciones:

- 1) Uso de una base de datos h2 en memoria.
- 2) Uso del módulo Spring Test para testear los controllers.

<https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-testing.html>

4. Entrega

- El proyecto se debe entregar en formato jar ejecutable
 - o `java -jar arvato-candidates-spring-bot.jar`
- Además, se debe entregar el código fuente del proyecto spring-boot en formato zip.
- Opcionalmente un fichero README.md.