# Challenge Java 2024

## Enunciado

Se requiere implementar una aplicación java que exponga un api por http para cumplir los siguientes casos de uso:

1 Caché de puntos de venta

La aplicación deberá contener un cache en memoria de "puntos de venta", de los que se conoce su identificador numérico y el nombre de este.

Información inicial de puntos de venta:

Id, Nombre

1, CABA

2, GBA\_1

3, GBA\_2

4, Santa Fe

5, Córdoba

6, Misiones

7, Salta

8, Chubut

9, Santa Cruz

10, Catamarca

Para este cache de puntos de venta se requerirán endpoints HTTP para

(1) Recuperar todos los puntos de venta presentes en el cache

(2) Ingresar un nuevo punto de venta

(3) Actualizar un punto de venta

(4) Borrar un punto de venta

2 Caché de costos entre los puntos de venta

También se deberá poseer un caché en memoria que guarde el costo (numérico) de ir de un punto de venta a otro. Se comprende que

- el costo nunca podría ser menor a cero

- el costo de ir a un punto de venta a si mismo es irrisorio, 0

- el costo de ir del punto de venta A al punto de venta B siempre será igual a hacer el camino inverso.

- no todos los puntos de venta están directamente conectados, y puede que un punto de venta B sea inalcanzable desde un punto A

- si un punto de venta A está comunicado directamente con un punto de venta B ese camino es único y no se posee un camino directo paralelo

- el camino más directo entre dos puntos puede no ser el menos costoso

Información inicial de costos entre puntos de venta:

IdA, IdB, Costo

1,2,2

1,3,3

2,3,5

2,4,10

1,4,11

4,5,5

2,5,14

6,7,32

8,9,11

10,7,5

3,8,10

5,8,30

10,5,5

4,6,6

Para este cache de costos entre puntos de venta se requerirán endpoints HTTP para

(1) Cargar un nuevo costo entre un punto de venta A y un punto de venta B (crearía un camino directo entre A y B con el costo indicado)

(2) Remover un costo entre un punto de venta A y un punto de venta B (removería en caso de existir un camino directo entre A y B)

(3) Consultar los puntos de venta directamente a un punto de venta A, y los costos que implica llegar a ellos

(4) Consultar el camino con costo mínimo entre dos puntos de venta A y B. (indicar el costo mínimo, y el camino realizado, aprovechando los nombres de los puntos de venta del caché del punto anterior)

3 Acreditaciones

Además del control de cache de los puntos de venta se requiere otro end point http que reciba "acreditaciones".

La información relevante de la acreditación que se recibirá de forma externa es un importe y un identificador de punto de venta.

Con esa información provista debemos enriquecer esta acreditación con

(1) la fecha en la que se recibió el pedido

(2) el nombre del punto de venta que le corresponde consultando el cache en memoria, o fallando si el punto de venta no existe

Esta información enriquecida deberá ser persistida en una BBDD (preferentemente no una en memoria, y que sea externa a la aplicación) los atributos que deben persistirse son

- importe

- identificador de punto de venta

- fecha de recepción

- nombre del punto de venta

Esta misma información deberá poder ser consultable a través de otro end point http.

Como parte de la resolución

(1) indique como debe compilarse el código fuente, incluya comandos o scripts que deban tenerse en cuenta, es deseable la posibilidad de correr la aplicación en un contenedor Docker, pero no es imperativo.

(2) indique cómo debe ejecutarse este compilado / imagen creada en el punto anterior y que requisitos requiere en el host para poder realizarlo.

(3) indique como puede probarse la aplicación en los casos de uso del enunciado

(4) En caso de asumir supuestos por favor dejen los indicados.

Tenga en cuenta

(1) La aplicación entregada deberá poder soportar un alto grado de concurrencia sin que se presenten errores causados por la implementación realizada.

(2) Serán apreciadas inclusiones de nuevas utilidades disponibles en las últimas versiones de Java.

Indicar cuales fueron en caso de incluirse, cómo funcionan, en qué versión fueron incluidas.

(3) Será apreciado el uso de patrones de diseño.

Indicar cuales fueron en caso de incluirse.

(4) Será apreciado diagramas de clases / tablas / secuencias (que considere prácticos)

No adjuntar archivos que requieran de una herramienta específica para poder ser abiertos, salvo que la misma pueda ser fácilmente utilizable de forma web.

Lo ideal sería que los diagramas puedan renderizarse de forma natural y nativa en el repositorio entregado.

(4) El código fuente debe contar con test unitarios. Es deseable que se alcance un % de cobertura mayor al 70%, sin incluir en la solución 'smells', 'bugs', 'issues' o dependencias que posean vulnerabilidades críticas (esto último en la medida de que existan nuevas versiones que posean dichas vulnerabilidades corregidas). En lo posible adjunte un reporte que indique de estos resultados.

La resolución deberá ser presentada en un repositorio git con visibilidad pública para que pueda ser evaluado.

Es posible que distintos aspectos de la resolución deberán ser defendidos durante una próxima entrevista técnica.