**Laboratorio 03-II: API Gateway (Total= 5 ptos)**

**Pre-requisitos:**

* Lenguaje de Programación - Libre

**Pre-Work: Create Microservices**

1. Crear un microservicio que dado el nombre de una canción devuelva la letra(lyrics).

Dataset: <https://www.kaggle.com/datasets/notshrirang/spotify-million-song-dataset>

1. Crear un microservicio que dado el nombre de la canción(track) te saque data relevante como duración o fecha de lanzamiento.

Dataset:

<https://www.kaggle.com/datasets/kapturovalexander/spotify-data-from-pyspark-course>

**Parte I: API Gateway as Facade.**

1. Crear un API Gateway que exponga el siguiente endpoint:

GET /artist-lyrics?name=\*\*\*

Request:

NA

Response:

{

artist: ABBA,

songs: [

“A1” : {

duration: xxx,

lyrics:

},

“B1” : {

duration: xxx,

lyrics:

}

etc..

]

}

Hint: usar los dos microservicios creados previamente.

**Parte II: API Gateway as Centralized Authentication Point**

1. Sus endpoints están expuestos y son vulnerables. Implemente un mecanismo de autenticación(no autorización) que proteja los dos endpoints.

**Warning**: esto se debe implementar en un punto único y los microservicios no deben ser modificados.

**Parte III: API Gateway as Format Converter**

1. Un nuevo requerimiento solicita que el archivo se retorne como csv y xml. Ya que por sistemas legacy necesitan consumir la data.
2. Modificar el endpoint para que soporte un nuevo parámetro ***format*** cuyos valores pueden ser CSV o XML.

Ejemplo: GET /artist-lyrics?name=\*\*\*?format=XML

**Parte IV: API Gateway for Versioning**

1. Por tema de performance y que la letra(lyrics) pesan demasiado se desea que se haga opcional los lyrics mediante un parámetro nuevo. Pero se desea mantener la versión previa ya que algunos clientes tienen sistemas con mejor rendimiento y consideran este cambio innecesario.

Ejemplo:

GET v2/artist-lyrics?name=\*\*\*?format=XML?Lyrics=OFF

Entregable:

* Repo en github con solución.
* Evidencia en este archivo word de las ejecuciones y responses obtenidos.