

ENTREGA 2 – DXL

GRUPO DXL (DEVELOPING XTREME LANGUAGE) ENTREGA 2 – 20/02/2025

CONTENIDO

1.	Introducción.....	1
2.	Toma de decisiones.....	1
2.1.	Orden de desarrollo	1
2.2.	Email como PK	2
2.3.	AuthorizationT	2
2.4.	Token = 1	2
2.5.	Top of the Tops	2
3.	Problemas.....	2
3.1.	Incompatibilidad entre HTTP Basic-Auth y Bearer Token de la api	2
3.1.	Incompatibilidad con el entorno local.....	3
3.2.	Problema de comprensión de Top of the Tops.....	3
4.	información de interés para el corrector.....	3
1.1.	Ubicación.....	3
1.2.	Testear la implementación	3

1. INTRODUCCIÓN

En este documento se redactan las decisiones tomadas para llevar a cabo la continuación del proyecto relacionado con la API de Twitch. En contraposición al anterior documento, en este nos vamos a centrar en la toma de decisiones ordenada cronológicamente y, seguidamente, los problemas más destacados que nos hemos encontrado en el proceso de desarrollo.

2. TOMA DE DECISIONES

2.1. ORDEN DE DESARROLLO

Para realizar esta práctica del conjunto del proyecto hemos decidido seguir el orden en el que se redacta el documento de premisas:

- **Registro de usuarios:** Se inició con el registro, creando la base fundamental para la autenticación, asegurando la generación y gestión de API Keys únicas.
- **Autenticación y login:** El siguiente paso fue implementar el flujo de obtención del token de sesión, utilizando la API Key generada previamente, lo que permite validar el acceso de manera segura y eficiente.
- **Endpoint “Top of the Tops”:** Finalmente, se desarrolló el endpoint que proporciona la información de los juegos y videos más populares. Así, todas las solicitudes pasaran por el proceso de autenticación previamente implementado.

2.2. EMAIL COMO PK

En la base de datos estábamos utilizando un identificador como Primary Key, no obstante, esto complicaba la autenticación, entonces decidimos que la manera de esquivar el problema era asignar el email como la PK del usuario en la base de datos. Así conseguimos simplificar el código y usar el ON DUPLICATE para no necesitar dos consultas para lo mismo.

2.3. AUTHORIZATIONT

Usar el Header de Authorization funcionaría bien en local, pero el server que estamos usando necesita que la Authorization sea la básica de http, y esta no es compatible con el Bearer token, así que en vez de Authorization a secas vamos a pasar **Authorizationt**.

2.4. TOKEN = 1

En el flujo de validación de tokens, se añadió un control específico para evitar accesos no autorizados mediante tokens genéricos: Si el valor del token recibido es igual a 1, la función ValidateAccessToken devuelve automáticamente un error de acceso inválido. Este control, aunque simple, actúa como una primera barrera de seguridad, evitando intentos básicos de evasión de la autenticación.

2.5. TOP OF THE TOPS

El problema y la decisión se explican en el punto 3.3

3. PROBLEMAS

3.1. INCOMPATIBILIDAD ENTRE HTTP BASIC-AUTH Y BEARER TOKEN DE LA API

Nuevamente, durante el desarrollo nos topamos con un problema significativo: el hosting en CDmon tiene activado por defecto el método de autenticación **HTTP Basic-Auth**, el cual solicita unas credenciales (usuario y contraseña) para cada petición que se realice a la API. Este sistema no se puede desactivar hasta donde hemos mirado.

El problema surge a raíz de que el encabezado de autorización (Authorization) se utiliza tanto para Basic-Auth como para los Bearer Tokens requeridos por la API. Esto genera una colisión, puesto que el encabezado solo puede contener uno de los dos valores. Si se incluyen las credenciales del Basic-Auth, el Bearer Token es sobrescrito, por lo que la autenticación con la API no es posible.

La solución se explica en el punto 2.3

3.2. INCOMPATIBILIDAD CON EL ENTORNO LOCAL

La base de datos hosteada en CDmon es incompatible con el entorno local. El funcionamiento base de CDmon impide las conexiones a bases de datos externas. Únicamente se puede acceder a ella a través de su propio localhost.

3.3. PROBLEMA DE COMPRENSIÓN DE TOP OF THE TOPS

Nos topamos con dudas que tuvimos que consultar a través del Slack en lo referente al desarrollo de Top of the Tops. La principal era, básicamente, que no sabíamos cuál de las siguientes dos opciones era la correcta:

- Recabar información de los 40 streams más vistos de los 3 juegos populares, agruparlos por creador y juego y devolver un string con la información agrupada de cada streamer para cada juego.
- Recabar información de los 40 streams más vistos de los 3 juegos populares, agrupar todos los streams del top del streamer con el vídeo más visto, procesar esa información para cada juego y devolver el string con el mejor de cada juego.

Dado que se trataba del Top of the Tops, la mejor opción nos pareció la segunda, ya que es lo que más se acerca al contexto que ofrece el enunciado.

4. INFORMACIÓN DE INTERÉS PARA EL CORRECTOR

1.1. UBICACIÓN

Todos los documentos se encuentran en el propio GitHub

1.2. TESTEAR LA IMPLEMENTACIÓN

Para ejecutarlo, consultar el README.md. Ahí se encuentra toda la documentación Acerca del funcionamiento y ejecución.