DOCUMENTO DE TOMA DE DECISIONES DXL

GRUPO DXL (DEVELOPING XTREME LANGUAGE) ENTREGA 1 – 07/02/2025

CONTENIDO

1. INTRODUCCIÓN

En este documento se redacta las decisiones tomadas para llevar a cabo este primer proyecto de manera progresiva, comenzando con la estructura y tratamiento de datos, continuando con la API y, en paralelo, el sistema de control de versiones. Así mismo, queda por escrito el proceso de gestión de cambios imprevistos y problemas en el desarrollo.

2. TOMA DE DECISIONES

2.1. ESTRUCTURA Y ALMACENAMIENTO DE DATOS

2.1.1. ESTRUCTURA INICIAL

En una primera instancia, tras haber leído los requisitos del proyecto, se nos dio a entender que los datos corrían por nuestra cuenta, y que para gestionarlos necesitábamos almacenarlos nosotros mismos. En ese momento se nos

presentó la primera y más sencilla decisión: Qué estructura de base de datos y entorno emplear. Sin meditar demasiado el asunto, nos decantamos por **MySQL** por los siguientes motivos:

- Amplio conocimiento del lenguaje y herramientas asociadas dentro del equipo.
- Compatibilidad con entornos PHP, facilitando la integración.
- Popularidad y facilidad de acceso en entornos de desarrollo.
- Capacidad para gestionar eficientemente las consultas que el API iba a requerir.

2.1.2. ADAPTACIÓN A PREMISAS POSTERIORES

El viernes 07/02/2025 se nos transmitió a través de SLACK que para la realización de la práctica había que recabar los datos diréctamente desde la API de Twitch. Esto era nuevo para nosotros, ya que **en los requisitos de la práctica se establece explícitamente**:

"Todas las peticiones y respuestas están basadas en datos falsos, inventados, de prueba. Lo que importa es que esa es la información que tenéis que aseguraros que recibís y devolvéis."

tuvimos que adaptarnos rápidamente para cumplimentar los requisitos del proyecto.

Decidimos hacer que la API que ya teníamos creada se comunicase diréctamente con la API de Twitch en lugar de con la base de datos que habíamos creado.

- Se implementó un manejo de errores que, en caso de fallo de comunicación con la API de Twitch, devuelve un **error 501 (Not Implemented)**.
- Se ajustó el código para minimizar el tiempo de respuesta y optimizar las consultas a la API externa.

2.2. API

Al inicio del proyecto decidimos por unanimidad gestionar las llamadas a la API a través de Postman. Esto nos habilita un entorno de pruebas perfecto para el desarrollo.

2.2.1. DECISIÓN INICIAL

Esta fue la decisión que más tuvimos que meditar, teníamos dos opciones disponibles: alojarlo en GitHub para integrarlo con el proyecto o usar CDmon. Finalmente nos decantamos por esta última opción por los siguientes motivos:

- **Compatibilidad total con PHP:** CDmon ofrece soporte nativo para este lenguaje, evitando configuraciones adicionales que habrían sido necesarias con otras opciones.
- **Acceso público:** Permite que el API sea accesible desde cualquier ubicación, esto es ideal puesto que se especifica como un requisito del proyecto.
- **Facilidad de despliegue:** La interfaz de CDmon simplifica el proceso de configuración y despliegue del servicio.

2.2.2. ADAPTACIÓN A PREMISAS POSTERIORES

El viernes 07/02/2025 se nos comunicó que debíamos de que la API se pudiese ejecutar en local. Hemos decidido mantener el servicio activo en CDmon, no obstante, hemos reestructurado el proyecto de forma que todo se pueda ejecutar en local.

2.3. CONTROL DE VERSIONES

Aquí no hubo que tomar ninguna decisión, ya que un requisito para la práctica era que el proyecto estuviese gestionado en GitHub. Las acciones realizadas para asegurar un correcto versionado fueron:

- Creación de un repositorio público para compartir el código.
- Documentación completa en el archivo **README.md**, que incluye instrucciones sobre cómo ejecutar y levantar el proyecto.
- Uso de ramas para implementar y probar funcionalidades antes de fusionarlas en la rama principal.

3. PROBLEMAS ENCONTRADOS

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

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.

3.1.1. SOLUCIÓN

Tras reevaluar el proyecto el viernes 07/02/2025, al gestionarse en local, nos evitamos este paso y por tanto, no supone un problema

3.2. CAMBIOS EN EL PARADIGMA DEL PROYECTO

El viernes 07/02/2025, cuando se nos comunicaron los cambios en el método de ejecución del proyecto, se presentó el siguiente error: con el paradigma anterior no estábamos empleando los id de cliente ni el token de la API de Twitch, si no que estábamos introduciendo algunos inventados.

3.2.1. SOLUCIÓN

A este problema y otros menores que surgieron, les dimos solución con una serie de cambios en el código que afectaban a la comunicación, por ejemplo, que en vez de enviar requests a la base de datos, enviase a la API de Twitch.

3.3. CERTIFICADOS DE PHP

A la hora de reconstruir el proyecto para habilitarlo en un entorno local, tuvimos un problema: al realizar una petición a la API de Twitch, pese a que el código supiésemos a ciencia cierta que funcionaba corréctamente, nos devolvía todo el rato “null”, “false” y respuestas de esa índole.

3.3.1. SOLUCIÓN

Resulta que la raíz del problema residía en que no disponíamos de un certificado SSL válido. La solución ha sido descargarlo e implementarlo en el repositorio del proyecto para que la ejecución acceda a el de forma automática.

4. INFORMACIÓN DE INTERÉS PARA EL CORRECTOR

4.1. UBICACIÓN

Todos los documentos se encuentran en el propio GitHub

4.2. TESTEAR LA IMPLEMENTACIÓN

Para ejecutarlo, consultar el README.md ahí se encuentra toda la documentación a cerca del funcionamiento y ejecución.