Desarrollo de Quickmatch: Backend & Frontend

Este informe presenta el progreso realizado en el desarrollo del backend y frontend de la plataforma Quickmatch, incluyendo la elección de tecnologías, resolución de problemas, y los siguientes pasos a seguir.

Quickmatch: Tecnologías Escogidas

Lenguajes

Java, JavaScript, y CSS.

Framework

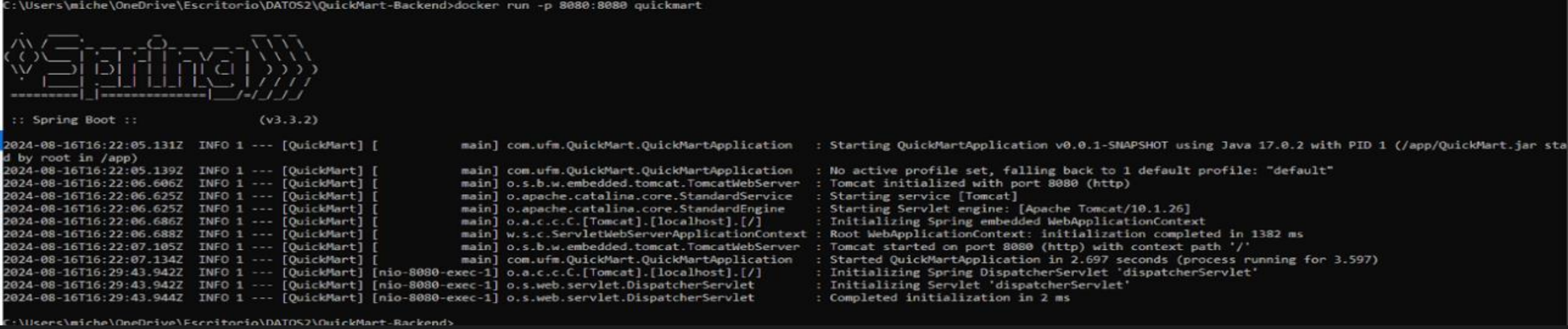
SpringBoot.

Data

SQL / NoSQL.

ORM

JPA (Java Persistence API).



Primera Etapa: Selección y Resolución

1

1. Selección de Lenguajes

Se seleccionaron los lenguajes Java, JavaScript, y CSS para la construcción de la plataforma.

2

2. Framework & ORM

Se eligió SpringBoot como framework principal y JPA para la gestión de la base de datos.

3

3. Problemática con PATHS

Se encontró un problema con la gestión de rutas en el backend, resolviendo el acceso a archivos.

Estructura del Backend

Config

Configuración de la aplicación, incluyendo la configuración de la base de datos, el servidor, y otros aspectos.

Controllers

Controladores que reciben las peticiones HTTP del frontend y las procesan.

DTO

Objetos de transferencia de datos que se utilizan para intercambiar información entre el backend y el frontend.

Entities

Representación de los datos en la base de datos.

Repositories

Interfaz para interactuar con la base de datos.

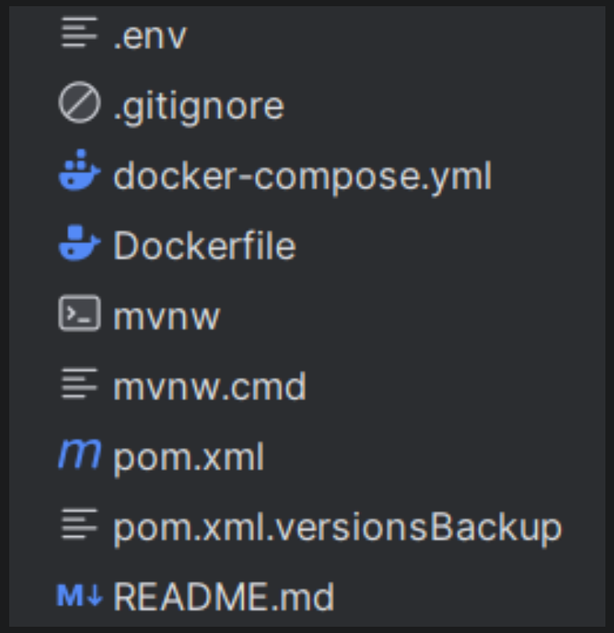
Services

Lógica de negocio que realiza las operaciones con los datos.

Pruebas & Desarrollo en Docker

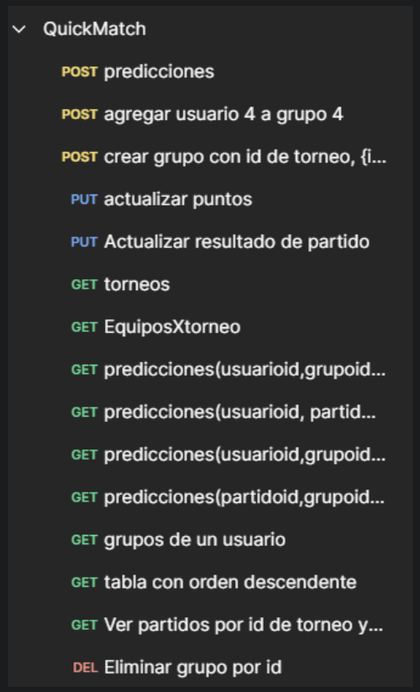
Docker

Docker se utiliza para el desarrollo, pruebas, y despliegue de la plataforma.



Postman

Postman se utiliza para probar las API del backend.



Next Steps: Hitos Importantes

1. Completar el backend de la API.
2. Finalizar la landing page del frontend.
3. Entender la indexación y caché en la plataforma.
4. Implementar la indexación y caché.
5. Realizar pruebas con JMeter para evaluar el impacto del caché.
6. Utilizar un profiler para medir la utilización de memoria.

Uso de @Caché: Optimización de Rendimiento

76

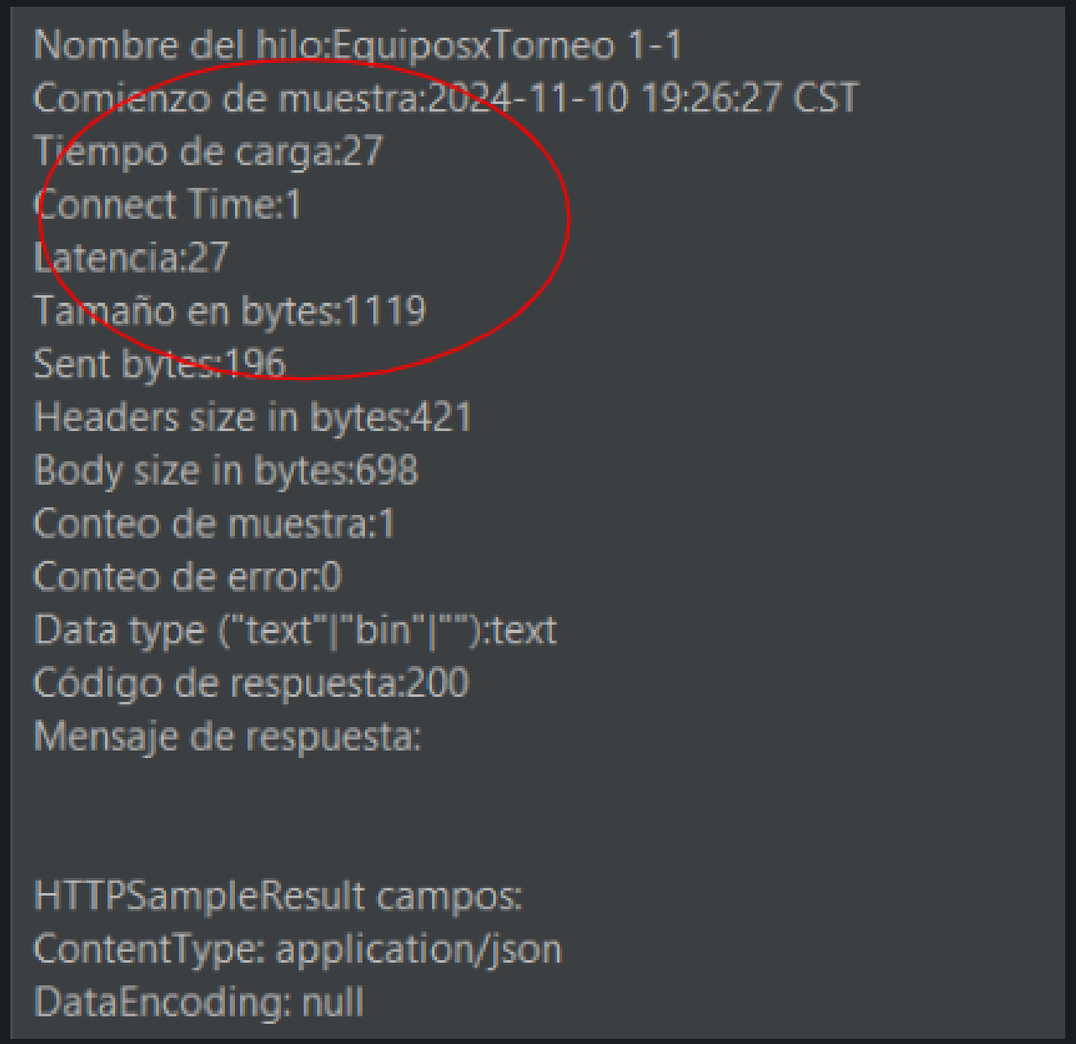
Sin @Caché

Tiempo de carga inicial.

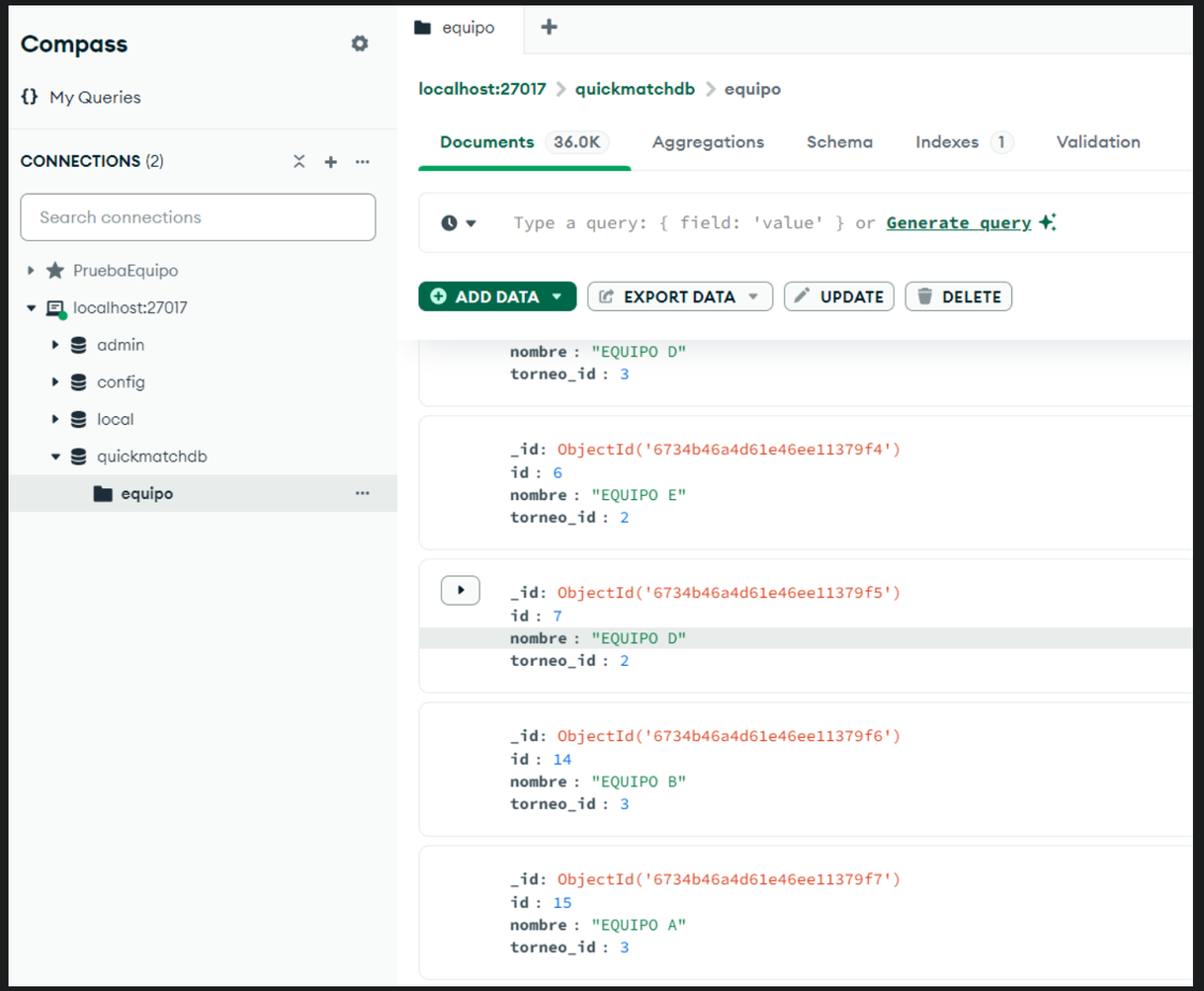
27

Con @Caché

Tiempo de carga con caché.



Nosql: Evolución hacia MongoDB



Nombre del hilo:Grupo de Hilos 1-1
Comienzo de muestra:2024-11-13 08:25:13 CST
Tiempo de carga:135
Connect Time:51
Latencia:127
Tamaño en bytes:829
Sent bytes:200
Headers size in bytes:421
Body size in bytes:408
Conteo de muestra:1
Conteo de error:0
Data type ("text"|"bin"|""):text
Código de respuesta:200
Mensaje de respuesta:

HTTPSampleResult campos:
ContentType: application/json
DataEncoding: null

La primera llamada es de 135 (vs 192 usando MySQL)