Warp Zone Wonders

Identificación de las necesidades del proyecto

Este proyecto apunta a las necesidades que tienen los aficionados de los videojuegos de asegurarse que los juegos que van a comprar merezcan la pena teniendo en cuenta las opiniones de la comunidad.

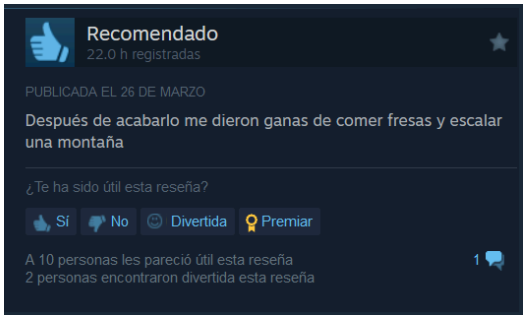
Breve análisis/comparativa con las alternativas del mercado

Como competidores potenciales podemos encontrar a Steam e IGN.

IGN suele contratar game journalists, reporteros que se dedican a jugar videojuegos antes de su salida para escribir reseñas, pero al tener tan poco tiempo para jugar antes del lanzamiento puede hacer que sus reseñas sean demasiado apresuradas.

Steam lo hace mejor en comparación ya que son los propios usuarios los que escriben las propias reseñas, por lo que suelen salir más tarde pero si encuentras un usuario que tenga gustos parecidos a los tuyos puedes encontrar nuevos juegos interesantes de forma rápida.

La mejora que me gustaría implementar para competir contra este tipo de reseñas es un sistema de comunidades en el que se pueden buscar reseñas según la comunidad por la que te sientas más identificado.



Justificación del proyecto

Este proyecto tiene grandes ventajas sobre las alternativas ya que al tener reseñas escritas por los propios usuarios le da mucha credibilidad y autenticidad a las reseñas que nos encontraremos.

El filtrado por comunidades nos da un filtrado personalizado muy en conexión con los grupos con los que se identifican los usuarios, lo cual hace que tiendan a ser más activos.

Debido a todo esto mejorará la capacidad de decisión de nuestros usuarios a la hora de comprar juegos nuevos haciendo que ahorren tiempo y dinero.

Uso de stack tecnológico

MySql (BBDD):

Mi decisión al usar sql se basa en que permite manejar grandes volúmenes de datos de manera eficiente y la capacidad de almacenar imágenes, esto es prioritario en un sistema en el que se necesite manejar una gran cantidad de datos, en nuestro caso una gran cantidad de reseñas.

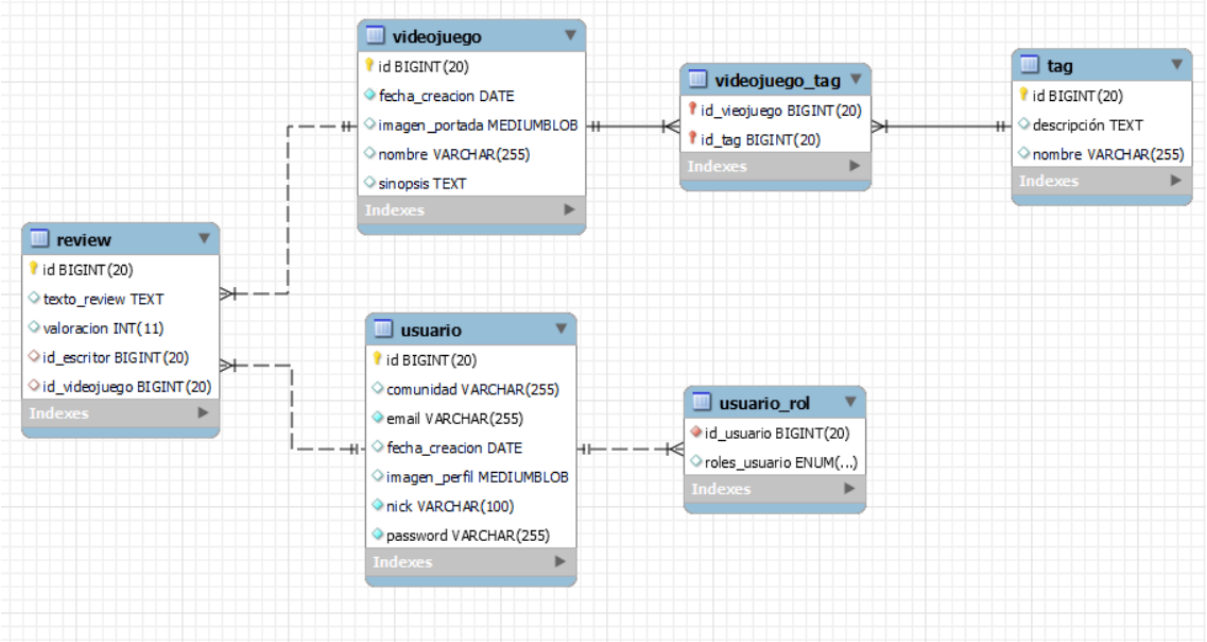
Spring Boot (Backend):

Mi decisión al usar Spring Boot se basa en la integración sencilla con MySql, las características de seguridad gracias a la configuración de web security basada en tokens y el uso de Java por su programación orientada a objetos.

Angular (Frontend):

Mi decisión al usar Spring Boot se basa en la capacidad de usar componentes para poder reutilizar código de forma eficiente y tiene la ventaja sobre otros modelos basados en JavaScript, ya que al usar TypeScript permite el tipado de los objetos y permite un entendimiento más rápido de los componentes generados previamente.

Esquema de la BD



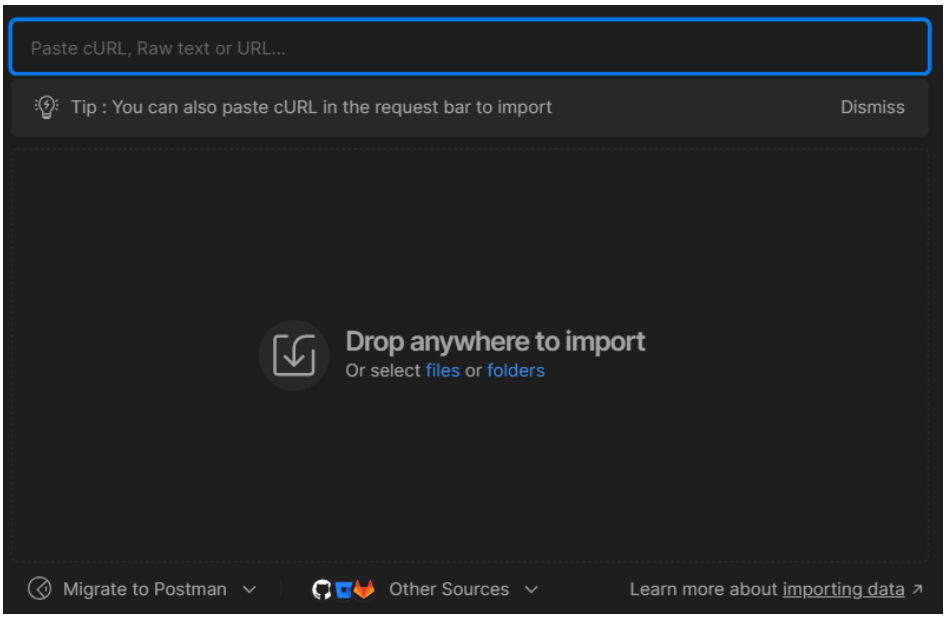
Prototipo en Figma

[Enlace a figma]

Definición API REST

La documentación de la Api la podemos encontrar en el repositorio de github del proyecto.

Es el archivo exportado de la documentación de Postman, así que será necesario importarlo mediante el importador de proyectos propio de Postman.

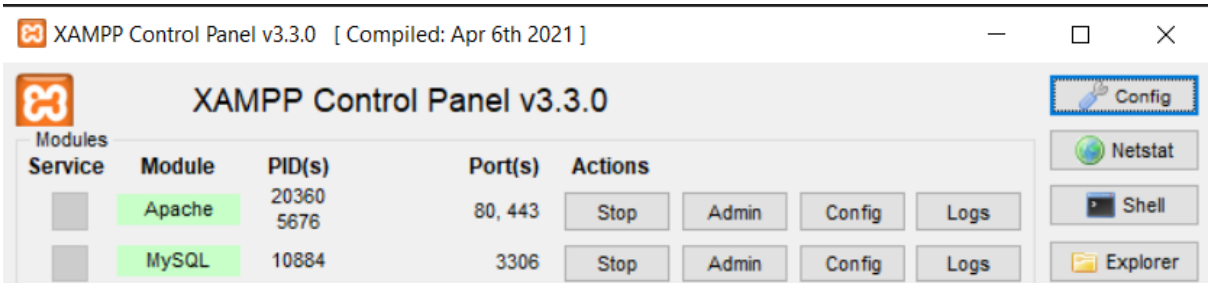


Manual de despliegue

Actualmente el despliegue se está realizando a mano, en un futuro del proyecto una vez lo queramos desplegar en una web en vez de en local se deberá generar un Docker-Compose para el proyecto.

Levantar la base de datos:

La forma más sencilla de levantarla sería lanzar una base de datos mediante XAMPP, en mi caso estoy usando el MySql estándar que viene con la instalación básica.

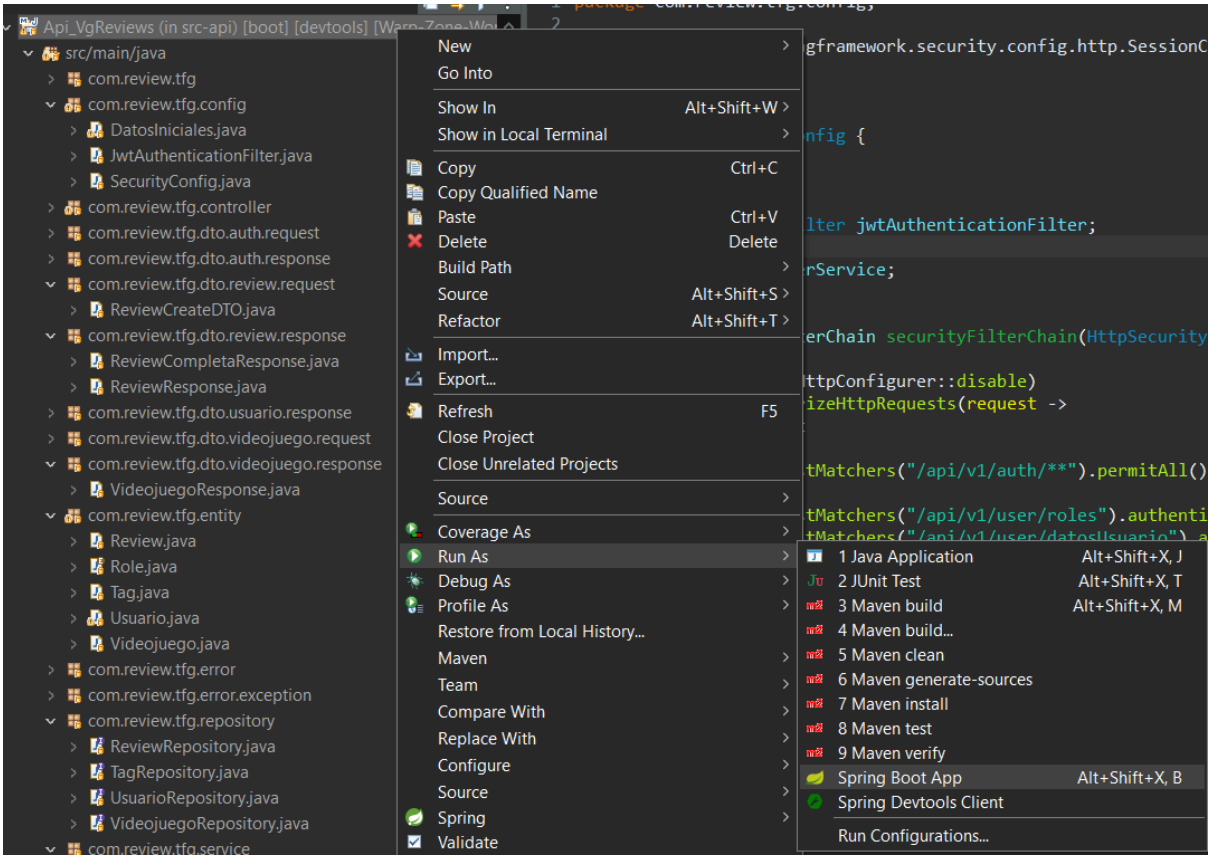


Levantar la api:

Antes de levantar la api nos dirigiremos al application.properties para introducir la dirección de la bbdd junto con las credenciales de inicio de sesión en la misma.

```
application.properties
1  server.port=8080
2  spring.profiles.active=demo
3  server.error.whitelabel.enabled=false
4
5  jwt.secret=3Vs6wFZNv7T75lC6P/KHhBulb5HIG6e8KUdLskt+ssw=
6  spring.datasource.url=jdbc:mysql://localhost:3306/prueba
7  spring.datasource.username=root
8  spring.datasource.password=
9  spring.jpa.hibernate.ddl-auto=update
10 spring.jpa.show-sql=true
11 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
12
```

Para levantar la api abriremos el proyecto con Eclipse y lo ejecutaremos como una Spring Boot App



Una vez levantemos el proyecto se creará automáticamente el usuario admin@admin.com con contraseña passwordadmin.

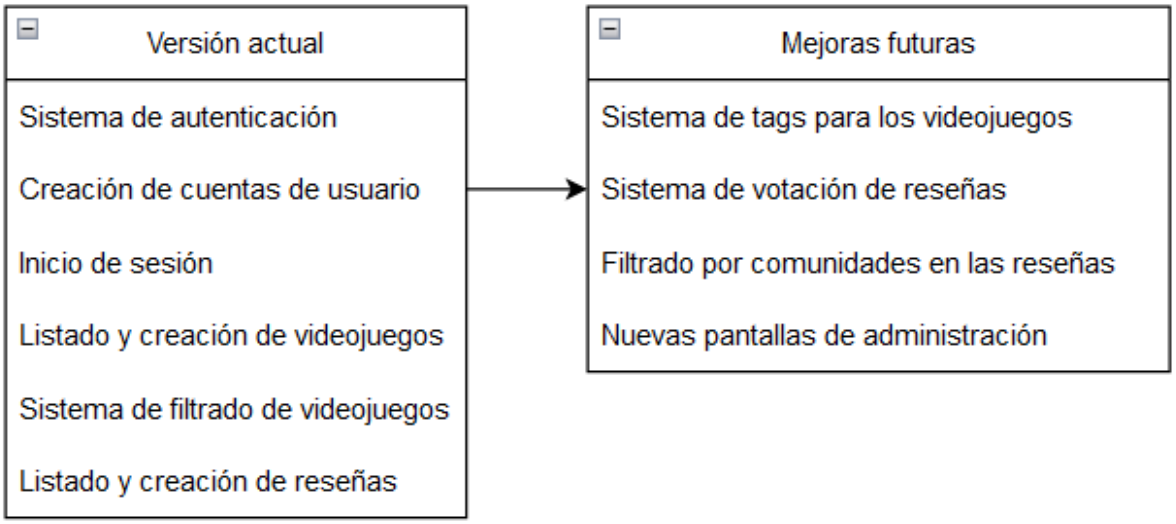
Levantar el frontend:

Para levantar el frontend nos colocaremos en la carpeta src-web del proyecto y ejecutaremos el comando “ng serve -o”, esto nos abrirá una pestaña del navegador apuntando a la url en la que está abierto.

Conclusiones y Postmortem

Al desarrollar este proyecto he encontrado muchas dificultades al momento de guardar imágenes en la base de datos, quizás usando servicios de cloud storage hubiese sido más sencillo tratar con las imágenes para no poner tanta carga en la api.

Me ha parecido que he sido muy ambicioso con el alcance del proyecto y esto ha hecho que no estén desarrolladas todas las funcionalidades a las que este proyecto puede llegar, aquí podemos encontrar un análisis de funcionalidades actuales y futuras.



Me parece que el proyecto tiene muchas salidas a futuro ya que se puede aplicar monetización gracias a las comunidades que se pueden formar junto a los videojuegos, algunos ejemplos de esto son:

- Anuncios: la gran cantidad de usuarios traerá anunciantes a nuestra página.
- Eventos en directo: las fechas de salida de videojuegos nuevos hacen períodos en los que los usuarios buscan opiniones sobre los nuevos juegos, podemos aprovechar esta oportunidad para hacer streaming en plataformas como Youtube o Twitch.
- Merchandising: una vez que nuestra aplicación crezca lo suficiente, algunos usuarios puede que estén interesados en comprar merchandising como camisetas, tazas o fundas para el móvil.

En conclusión, me parece que con suficiente tiempo este proyecto se puede transformar en una gran oportunidad lucrativa y de aprendizaje.