

Realizado por: Federico Rodriguez

1. PROBLEMA DE TORNEO DE VIDEOJUEGOS

A. ¿Qué arquitectura de software plantea y justifíquela?

La arquitectura que propongo para el problema es arquitectura de microservicios, ya que esta es una arquitectura sólida y para una empresa robusta como Bancolombia, las desventajas no serían un problema para el desarrollo.

Esta arquitectura permite una escalabilidad y flexibilidad continua que distribuye el trabajo en servicios independientes, además de poder agregar funciones a dichos servicios sin comprometer enteramente la aplicación, permitiendo mayor agilidad y facilidad para la entrega de versiones nuevas del aplicativo.

Además, con esta arquitectura la facilidad de mantenimiento es otra ventaja, ya que se simplifica la comprensión del código dado que cada microservicio tiene una responsabilidad definida.

Por último, esta arquitectura ofrece una disponibilidad continua, dado que si uno de los microservicios falla, los demás pueden seguir funcionando si no dependen de este, lo que minimiza el impacto en la aplicación.

B. ¿Cuál diagrama de clases plantea y justifíquelo?

Se definió un diagrama de clase por cada microservicio.

1. Microservicio de Creación de Torneos:

- **Torneo:** Esta clase es la entidad central del sistema, representando un torneo individual con sus características y relaciones asociadas. Atributos como el nombre, las fechas de inicio y finalización, y el estado del torneo permiten una gestión completa del ciclo de vida del torneo. La relación con la clase "Juego" indica qué juego se está utilizando en el torneo, mientras que la lista de transmisiones y participantes permite una gestión detallada de los eventos y la participación en el torneo.
- **Participante:** Representa a un jugador inscrito en el torneo. Los atributos incluyen información personal y de participación, como el nickname, la región y el juego en el que participa. La relación con la clase "Usuario" permite vincular cada participante a un usuario del sistema, garantizando la autenticación y el manejo de roles.

2. Microservicio de Venta de Entradas:

- **Entrada:** Representa una entrada a un torneo específico, incluyendo detalles como el precio y el estado de la entrada. La relación con la clase "Usuario" permite rastrear quién ha comprado la entrada, facilitando la gestión de las ventas y la entrada a los torneos.

3. Microservicio de Gestión de Juegos:

- **Juego:** Representa un juego individual que puede estar asociado a múltiples torneos. Los atributos de esta clase permiten identificar y clasificar el juego, y la relación con la clase "Torneo" muestra en qué torneos se está utilizando cada juego.

4. Microservicio de Gestión de Vistas Virtuales:

- **Transmisión:** Representa una transmisión virtual de un torneo, incluyendo información sobre la plataforma de transmisión, el enlace y la fecha y hora de la transmisión. La relación con la clase "Torneo" asegura que cada transmisión esté asociada a un torneo específico, permitiendo a los usuarios acceder a las transmisiones de los torneos en los que están interesados.
5. **Microservicio de Autenticación:**
- **Usuario:** Esta clase maneja la autenticación y los detalles del usuario, como el nombre de usuario, el correo electrónico, la contraseña y el rol. La relación con la clase "Entrada" permite gestionar las entradas compradas por cada usuario, garantizando que solo los usuarios autenticados puedan acceder a las funcionalidades del sistema.

Relaciones Entre Entidades:

- **Torneo y Juego:** Cada torneo está asociado a un único juego, lo que se refleja en la relación uno a uno entre "Torneo" y "Juego". Esto permite especificar claramente qué juego se jugará en cada torneo.
- **Torneo y Transmisión:** Un torneo puede tener múltiples transmisiones, lo que se modela mediante una relación uno a muchos. Esto facilita la planificación y gestión de las transmisiones en diferentes plataformas.
- **Torneo y Participante:** Un torneo puede tener muchos participantes, y un participante puede estar inscrito en varios torneos, lo que se refleja en la relación muchos a muchos.
- **Participante y Usuario:** Cada participante está asociado a un usuario, lo que permite manejar la autenticación y los roles de los usuarios.
- **Entrada y Usuario:** Cada entrada está vinculada a un usuario que la compra, facilitando el seguimiento de las ventas y la gestión de las entradas para los torneos.

C. ¿Qué modelo de base de datos plantea y justifíquela?

He decidido optar por una base de datos relacional para implementar en este sistema debido a que ofrecen una estructura definida y consistente, lo cual es esencial para mantener la integridad y la coherencia de los datos en un sistema con múltiples entidades interrelacionadas. Esto permite realizar consultas de manera más acertada, lo cual es clave para la aplicación de analíticas y métricas tanto para los usuarios como para los desarrolladores, permitiendo tener una idea de qué tipo de usuarios están usando el aplicativo y así aplicar estrategias de negocio.

D. ¿Qué protocolos de seguridad plantea y por qué?

Se plantean diferentes protocolos de seguridad:

- Usar aplicativos de análisis de código como SonarQube para analizar y corregir vulnerabilidades, bugs y code smells en el código.
- Desplegar la aplicación usando alguna cadena de computación en la nube como AWS o Azure.
- Tener respaldos de los datos, usando una base de datos basada en la nube con alguno de los servicios mencionados.
- Utilizar OAuth 2.0 para la autenticación de usuarios.
- Utilizar algoritmos de cifrado para las contraseñas.

- Utilizar consultas parametrizadas para evitar inyección de código SQL.
- Mantener logs de auditoría, especialmente para las operaciones más importantes o críticas.

E. ¿Qué elemento de integración y middleware propone para la solución y por qué?

Se pueden implementar varios servicios de middleware:

- Middleware de mensajería entre usuarios de la aplicación para gestionar la comunicación.
- Middleware de base de datos para la persistencia de datos.
- Middleware de servicios web debido a la arquitectura.
- Middleware de gestión de transacciones para el microservicio de venta de entradas.

F. ¿Cuáles servicios plantea y qué tecnología utilizaría?

Los servicios planteados serán los mismos que los microservicios. Estos serían:

- Servicio de creación de torneos.
- Servicio de venta de entradas.
- Servicio de gestión de juegos.
- Servicio de gestión de vistas virtuales.
- Servicio de autenticación y gestión de usuarios.

Tecnologías:

- Springboot para el backend,
- SQL Server para la base de datos,
- Hibernate para el ORM,
- RESTful para la API,
- RabbitMQ para el servicio de mensajería.