El propósito de este documento es especificar las funcionalidades de la aplicación "Canchas", así como las tecnologías y arquitecturas que se utilizarán para implementarlas. El objetivo es proporcionar una visión clara que permita seleccionar las tecnologías más adecuadas para el desarrollo del proyecto. Considerando las dimensiones del proyecto, se espera un alto volumen de peticiones y usuarios simultáneos. Por lo tanto, se busca asegurar una alta disponibilidad y optimización de los servicios.

**Listado de funcionalidades:**

1. Administración de usuarios.
   1. Registro y autenticación de usuarios.
   2. Administración de perfiles.
   3. Automación y roles.
2. Conexión entre jugadores.
   1. Conexión entre jugadores para organizar partidos.
   2. Búsqueda de jugadores y equipos.
   3. Gestión de invitaciones y aceptación de partidos.
3. Gestión de canchas.
   1. Reserva de canchas.
   2. Gestión de disponibilidad.
   3. Pagos y confirmación de reservas.
4. Funcionalidad de notificaciones.
   1. Envió de notificaciones push y correos electrónicos.
   2. Alerta de reservas, partidos y apuestas.
   3. Notificaciones de eventos.
5. Funcionalidad de apuestas.
   1. Gestión de apuestas entre partidos.
   2. Calculo y pago de premios.
   3. Historial de apuestas y transacciones.
6. Funcionalidad de chat.
   1. Búsqueda de usuarios.
   2. Envió de mensajes en tiempo real.

**Tecnologías sugeridas.**

Teniendo en cuenta que la finalidad es tener una plataforma administrativa dedicada a las actividades desempeñadas por los administradores de las canchas sintéticas y una plataforma móvil enfocada a los jugadores amateurs organizar y agendar partidos, se relacionan las tecnologías propuestas para llevar el desarrollo.

|  |  |  |
| --- | --- | --- |
| **Plataforma** | **Tecnología** | **Version** |
| Web | Angular |  |
| Móvil | Flutter |  |
| Backend | Spring boot |  |
| Motor BD | PostgresSQL |  |

**Descripción de arquitectura.**

En la fase inicial del proyecto se utilizará una arquitectura que facilite la agilidad en el proceso de desarrollo y que no conlleve un gran esfuerzo para realizar procesos de despliegue, adicional es necesario mantener un bajo acoplamiento entre los compontes para aumentar la flexibilidad del aplicativo, para esto se plantea utilizar la arquitectura monolito modular con el objetivo de en algún punto futuro realizar la migración hacia microservicios.

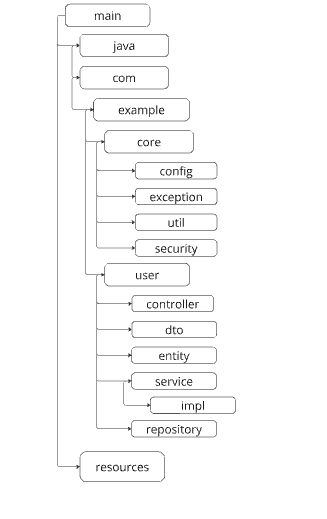
**Monolito modular:**

A continuación, se relacionan los módulos por los que estará conformado la aplicación, estos se definen en base a las funcionalidades anteriormente mencionadas.

1. User.
2. Player connection.
3. Field Management.
4. Notification.
5. Betting.

**Distribución de carpetas.**

A continuación, se relaciona un ejemplo de la distribución de capetas para un servicio con la tecnología spring boot.



**Descripción**

* **config/**

Nombre archivo: AppConfig.java

Configuración general de la aplicación, como beans y propiedades específicas.

* **controller/**

Nombre archivo: UserController.java

Controladores REST que manejan las solicitudes HTTP relacionadas con usuarios.

* **dto/**

Nombre archivo: UserDTO.java

Transferencia de datos

* **exception/**

Nombre archivo: GlobalExceptionHandler.java

Maneja todas las excepciones globales de la aplicación

* **entity/**

Nombre archivo: User.java

Clases que representan las entidades de la base de datos.

* **repository/**

Nombre archivo: UserRepository.java

Interfaces que extienden JpaRepository o CrudRepository para interactuar con la base de datos.

* **service/**

Nombre archivo: UserService.java

Clases abstractas que definen las funcionalidades.

* **service/impl/**

Nombre archivo: UserServiceImpl.java

Clases que extiende de UserService y contienen la lógica de negocio.

* **security/**

Nombre archivo: SecurityConfig.java

Configuración de seguridad, incluyendo reglas de acceso y protección de rutas.

* **util/J**wtUtil.java

**Costos**

2024-05-08

A continuación, se muestra cotización realizada en los principales prestadores de servicio nube, teniendo en cuenta la arquitectura, tecnologías y servicios anteriormente planteados.

**AWS**

|  |  |  |
| --- | --- | --- |
| **Descripción** | **Valor USD** | **Datos adicionales** |
| Amazon Virtual Private Cloud (VPC) | 73.90 | 10 GB mes |
| Amazon EC2 | 20 | 2 CPU, 4GB RAM, 25GB SSD |
| Total | 93.9 |  |