# Proyecto Jogo

Resumen

El proyecto consiste en el desarrollo de una plataforma web destinada a la gestión integral de torneos de futsal. La plataforma permitirá registrar, organizar y controlar múltiples torneos de forma autónoma, teniendo la experiencia tanto para los administradores como para los participantes. La solución está diseñada para estar presente en varios torneos con la expectativa de controlar competiciones máximo de 24 equipos.

Puntos clave:

1. Gestión de Torneos y Equipos:
2. Capacidad Máxima: La plataforma soporta hasta 24 equipos por torneo y 12 jugadores por equipo.
3. Autonomía Organizativa: El sistema organiza automáticamente los partidos de cada torneo, evitando conflictos de horarios si un equipo participa en múltiples torneos.
4. Control Administrativo: Los administradores tendrán acceso a herramientas para gestionar equipos, registrar resultados (goles y tarjetas), y controlar el estado de los pagos de inscripción.
5. Sistema de Clasificación y Playoffs:
6. Clasificación Automática: La plataforma calcula automáticamente la clasificación de los equipos en base a los resultados de los partidos.
7. Organización de Playoffs: Las fases de playoffs se generan automáticamente, basadas en la clasificación obtenida durante la fase regular del torneo.
8. Historial y Consulta de Torneos:
9. Historial de Torneos: Los usuarios pueden consultar el historial de torneos anteriores, incluyendo resultados, campeones, y estadísticas detalladas.
10. Notificaciones y Recordatorios:
11. Sistema de Notificaciones: Los usuarios recibirán notificaciones automáticas sobre próximos partidos, plazos de inscripción, y otros eventos importantes.

Diseño detallado del Sistema de Gestión Jogo.

1. Arquitectura del Proyecto
2. Front-end:

Tecnologías: Angular

Diseño Responsivo:

* Usar frameworks como Bootstrap o materializes para asegurar que la plataforma funcione bien en dispositivos móviles y de escritorio.

Interfaz de Usuario (UI):

* Dashboard para administradores con acceso a la gestión de torneos, equipos, jugadores, y configuración de reglas del torneo.
* Formularios para la creación y edición de torneos, equipos, y jugadores.
* Vista de calendario para visualizar los partidos programados.
* Tablas para mostrar clasificaciones, estadísticas de jugadores, y registros de partidos.

1. Back-end:

Tecnologías: Node.js con Express o Spring Boot para gestionar microservicios.

Base de Datos: PostgreSQL con el fin de almacenar información sobre torneos, equipos, jugadores, partidos, y estadísticas.

APIs REST: Exponer APIs para que el front-end pueda interactuar con los datos (gestión de torneos, equipos, jugadores, y programación de partidos).

Autenticación y Autorización: Implementar roles (administrador, usuario) usando JWT

1. Funcionalidades Específicas

Gestión de Torneos:

* Creación y configuración de torneos (nombre, fechas, equipos).
* Inscripción de equipos en un torneo, asegurando que el número no exceda los 24 equipos.
* Automatización de la programación de partidos, evitando conflictos si un equipo participa en varios torneos.

Gestión de Equipos y Jugadores:

* Registro y edición de equipos.
* Asignación de jugadores a los equipos, con un máximo de 12 jugadores por equipo.

Organización de Partidos:

* Algoritmo para programar partidos evitando solapamientos para equipos en múltiples torneos.
* Registro de resultados (goles, tarjetas) y actualización automática de las estadísticas y clasificaciones.

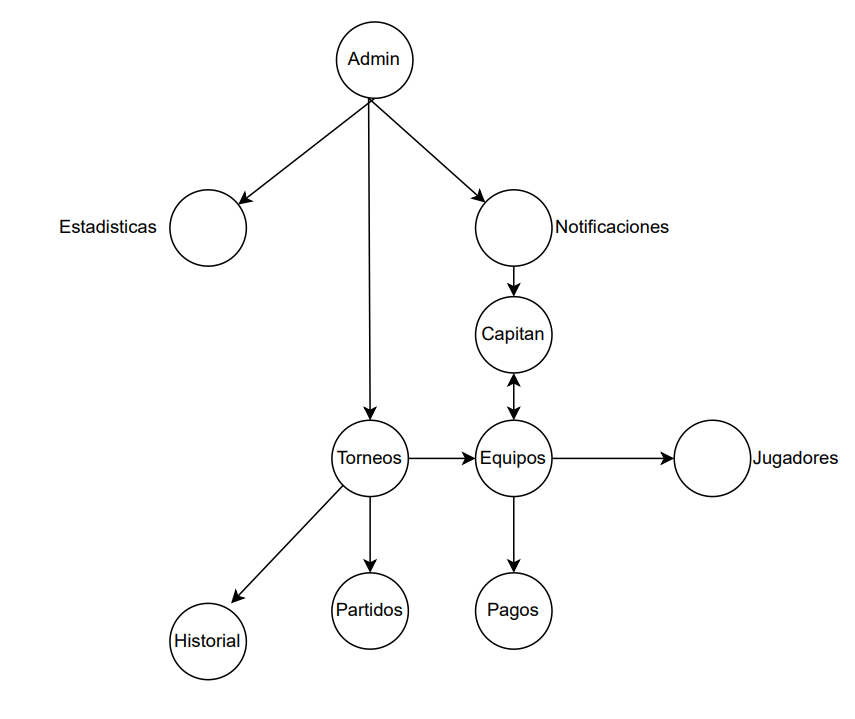
Gestión Financiera:

* Registro de pagos de inscripción.
* Monitoreo del estado de los pagos (pendientes, completados).
* Notificaciones a los administradores sobre pagos pendientes.

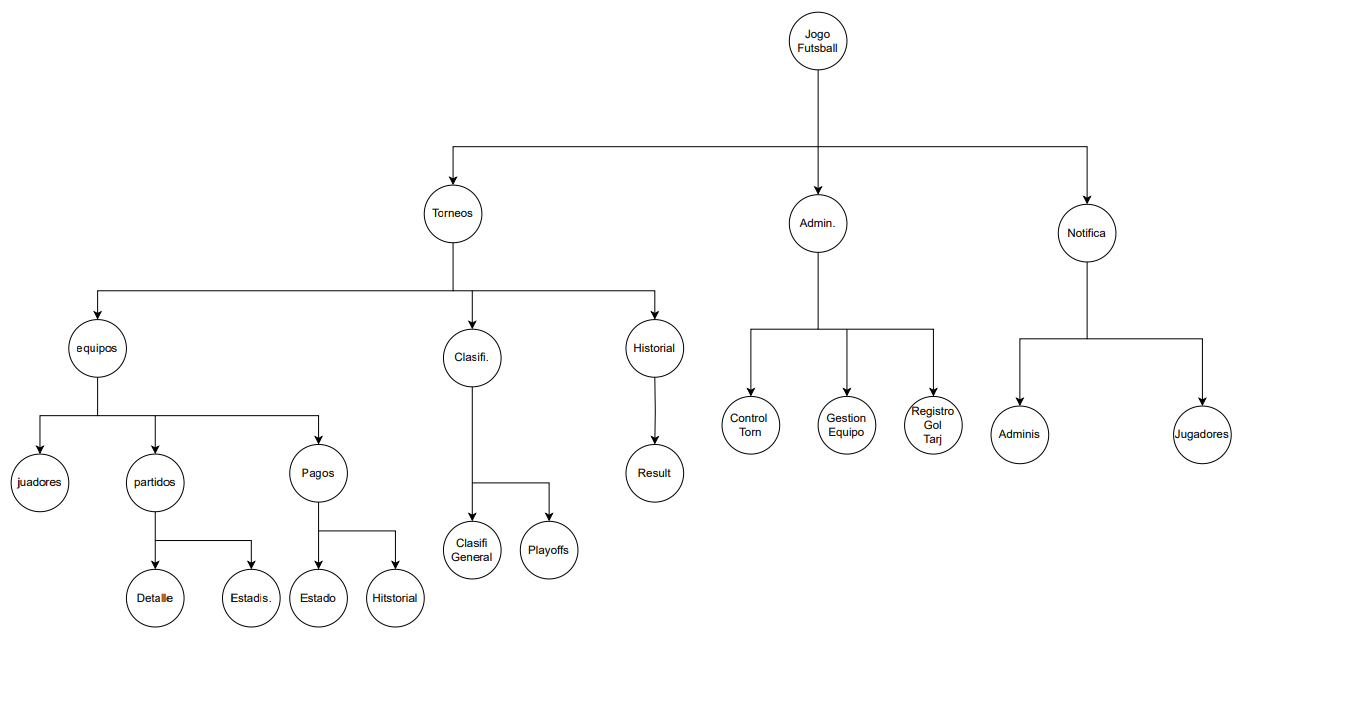
Panel Administrativo:

* Dashboard con resúmenes de torneos activos, equipos inscritos, pagos pendientes, y próximos partidos.
* Herramientas para enviar notificaciones a los equipos o jugadores.
* Reportes descargables en formato PDF o Excel con estadísticas de los torneos.

Diagrama de grafos que representen las interacciones y relaciones complejas en la gestión ambiental.



Diseño de la estructura de árbol para organizar la información



Uso de patrones de diseño creacionales (Singleton, Builder, u otros) en la estructura del sistema.

### 1. Singleton Pattern

1. **Gestión de Conexiones a la Base de Datos**: Dado que solo se debe crear una instancia por todo el proyecto.
2. **Manejador de Configuraciones**: Si tienes un objeto que maneja las configuraciones globales de la plataforma (como las configuraciones de los torneos, límites de equipos, etc.), este objeto podría ser un Singleton. Esto asegura que cualquier cambio en la configuración se refleje en toda la aplicación sin necesidad de múltiples instancias.

### 2. Factory Pattern

1. **Creación de Partidos en Diferentes Fases del Torneo**: Dependiendo de la fase del torneo (grupos, eliminatorias, finales), los partidos pueden tener reglas y configuraciones diferentes. Puedes usar una Factory para crear instancias de partidos específicos para cada fase, asegurando que cada tipo de partido cumpla con los requisitos de la fase correspondiente.
2. **Gestión de Notificaciones**: Al usar una notificación SMS se puede aplicar este tipo de patron.

### 3. Prototype Pattern

1. **Duplicación de Torneos**: Al momento de crear varios torneos con especificaciones parecidas, se puede aplicar este patron

Arquitectura

**Microservicios Propuestos:**

1. Servicio de Autenticación:
2. Gestión de usuarios y autenticación JWT.
3. Registro y recuperación de contraseñas.
4. Autorización de roles (jugadores, administradores).
5. Servicio de Gestión de Torneos:
6. CRUD para torneos: Creación, edición, eliminación de torneos.
7. Gestión de equipos (Registra equipos al torneo) y jugadores en torneos.
8. Creación automática de calendario de partidos.
9. Servicio de Gestión de Equipos:
10. CRUD de equipos.
11. Gestión de perfiles de jugadores y estadísticas.
12. Servicio de Pagos:
13. **Proveedores de Pago:** Integrar **Stripe** o **PayPal** para gestionar pagos en línea.
14. **Funcionalidades:**
15. Procesar pagos para inscripción a torneos, registro de jugadores, o cuotas mensuales.
16. Manejar devoluciones y reembolsos si los torneos se cancelan o reprograman.
17. Confirmación automática de pagos y actualización de estados de inscripción.
18. **Base de datos:** Registro de transacciones (ID de transacción, monto, fecha de pago, estado del pago).
19. **Webhooks:** Para recibir notificaciones en tiempo real de los pagos completados por los jugadores/equipos y actualizar el sistema.
20. Servicio de Notificaciones:
21. Recordatorios de partidos, cambios en el calendario.
22. Notificaciones de pagos exitosos o pendientes.