[Diego Alejandro Cortes Grimaldo](mailto:diacortesgri@unal.edu.co)

Super usuario

Usuario - Contraseña

dcort - Grimaldo759

**Proyecto Final**

**Descripción**

Gestionar un equipo de fútbol, especialmente en el ámbito amateur y semiprofesional, no es tarea fácil. Muchas veces, las herramientas disponibles son insuficientes y la organización se vuelve un reto: entrenamientos sin control, partidos desordenados y dificultades para hacer seguimiento al desempeño de los jugadores.

Para solucionar este problema, hemos diseñado un sistema web que facilitará la administración de un equipo de fútbol, permitiendo que cada persona involucrada tenga acceso a la información clave de manera sencilla y eficiente.

#### **Usuarios del sistema:**

1. **Futbolistas:** Pueden acceder a su perfil, consultar sus estadísticas de rendimiento, ver los entrenamientos programados y recibir información sobre partidos en los que participarán.
2. **Técnicos:** Se encargan de planificar y registrar entrenamientos, analizar el desempeño de los futbolistas y consultar estadísticas para mejorar el rendimiento del equipo.
3. **Organizadores de Liga:** Gestionan la programación de partidos, seleccionan equipos, establecen fechas y registran los resultados de los encuentros.
4. **Administradores:** Son responsables de la gestión de usuarios dentro del sistema, asegurando el acceso adecuado a cada perfil y manteniendo actualizada la información de los equipos y sus integrantes.

En conclusión, este sistema busca mejorar la organización de los equipos y facilitar la labor tanto de los jugadores como de quienes están detrás de la gestión, permitiendo que todos puedan enfocarse en lo más importante: jugar y disfrutar del fútbol.

## **Historias de Usuario**

### **Futbolista**

1. **Consultar perfil:** Como futbolista, quiero acceder a mi perfil para ver mi información personal y mis estadísticas.
2. **Ver estadísticas de rendimiento:** Como futbolista, quiero consultar mis estadísticas de rendimiento para analizar mi desempeño en los partidos y entrenamientos.
3. **Consultar entrenamientos programados:** Como futbolista, quiero ver los entrenamientos programados para saber cuándo debo asistir.
4. **Registrarse en un entrenamiento**: Como futbolista, quiero inscribirme en un entrenamiento para poder mejorar mis habilidades.
5. **Ver partidos programados:** Como futbolista, quiero consultar los partidos programados para conocer el calendario de juego de mi equipo.
6. **Consultar detalles de un partido:** Como futbolista, quiero ver los detalles de un partido para conocer la alineación, ubicación y hora.

### **Técnico**

1. **Planificar entrenamientos:** Como técnico, quiero programar entrenamientos para preparar mejor a mis jugadores.
2. **Registrar entrenamientos:** Como técnico, quiero registrar los entrenamientos realizados para llevar un control de las sesiones.
3. **Consultar entrenamientos programados:** Como técnico, quiero ver los entrenamientos agendados para organizar mis sesiones.
4. **Asignar futbolistas a entrenamientos:** Como técnico, quiero asignar futbolistas a entrenamientos específicos para mejorar sus habilidades según sus necesidades.
5. **Analizar estadísticas de futbolistas:** Como técnico, quiero consultar las estadísticas de los jugadores para evaluar su rendimiento y tomar decisiones tácticas.
6. **Registrar observaciones sobre el rendimiento:** Como técnico, quiero registrar observaciones sobre mis jugadores para hacer seguimiento de su progreso.

### **Organizador de Liga**

1. **Programar partidos:** Como organizador, quiero programar partidos para definir el calendario de la liga.
2. **Modificar programación de partidos:** Como organizador, quiero cambiar la programación de un partido en caso de imprevistos.
3. **Asignar equipos a un partido:** Como organizador, quiero asignar equipos a un partido para confirmar la participación de los clubes.
4. **Registrar resultados de partidos:** Como organizador, quiero registrar los resultados de los partidos para actualizar la clasificación de la liga.
5. **Consultar historial de partidos:** Como organizador, quiero ver el historial de partidos para analizar el desarrollo de la liga.
6. **Gestionar horarios y ubicaciones:** Como organizador, quiero gestionar los horarios y ubicaciones de los partidos para garantizar el uso adecuado de las canchas.

### **Administrador**

1. **Registrar nuevos usuarios**: Como administrador, quiero registrar nuevos usuarios para que puedan acceder al sistema.
2. **Modificar información de usuarios:** Como administrador, quiero modificar la información de los usuarios para mantener sus datos actualizados.
3. **Eliminar usuarios:** Como administrador, quiero eliminar usuarios que ya no forman parte del sistema para mantener la base de datos organizada.
4. **Gestionar equipos:** Como administrador, quiero gestionar equipos para mantener el orden en la organización del sistema.
5. **Asignar roles a usuarios:**  Como administrador, quiero asignar roles a los usuarios para definir sus permisos en la plataforma.
6. **Restaurar accesos y contraseñas:** Como administrador, quiero restablecer contraseñas de los usuarios para ayudar en caso de problemas de acceso.

### **Requisitos Funcionales**

1. El sistema debe permitir a los futbolistas consultar su perfil y estadísticas personales.
2. El sistema debe permitir a los técnicos planificar y registrar entrenamientos.
3. El sistema debe permitir al organizador de liga programar y modificar partidos.
4. El sistema debe gestionar la información de los equipos y jugadores.
5. El administrador debe poder registrar, modificar y eliminar usuarios.
6. El sistema debe permitir la asignación de roles a los usuarios.
7. El sistema debe permitir la gestión de horarios y ubicaciones de los partidos.
8. Los futbolistas deben poder registrarse en entrenamientos y consultar detalles de partidos.
9. El sistema debe proporcionar estadísticas detalladas para técnicos y futbolistas.
10. Los organizadores deben poder registrar los resultados de los partidos.

### **Requisitos No Funcionales**

1. El sistema debe ser accesible desde dispositivos móviles y computadoras.
2. La interfaz debe ser intuitiva y fácil de usar.
3. Los datos de los usuarios deben estar protegidos mediante autenticación segura.
4. La plataforma debe responder en menos de 2 segundos para cualquier consulta.
5. El sistema debe almacenar un historial de partidos y entrenamientos durante al menos un año.
6. Debe haber un sistema de respaldo de datos automático cada 24 horas.
7. El sistema debe estar disponible al menos el 99% del tiempo.
8. El acceso a las funcionalidades debe estar limitado según los roles asignados.

**Casos de Uso**

### **Futbolista**

1. **Consultar perfil**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista puede acceder a su perfil personal para ver su información y estadísticas.
   * **Flujo básico**:
     + El futbolista inicia sesión en el sistema.
     + El sistema muestra su panel de control.
     + El futbolista selecciona la opción "Perfil".
     + El sistema muestra la información personal y las estadísticas del futbolista.
   * **Flujo alternativo**:
     + Si el futbolista no está autenticado, el sistema redirige al usuario a la página de inicio de sesión.
2. **Ver estadísticas de rendimiento**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista consulta sus estadísticas para analizar su rendimiento en partidos y entrenamientos.
   * **Flujo básico**:
     + El futbolista selecciona "Estadísticas".
     + El sistema muestra las estadísticas de rendimiento del futbolista, incluyendo goles, asistencias, minutos jugados, etc.
   * **Flujo alternativo**:
     + Si no existen estadísticas registradas, el sistema muestra un mensaje informando al usuario que aún no tiene estadísticas.
3. **Consultar entrenamientos programados**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista consulta los entrenamientos programados.
   * **Flujo básico**:
     + El futbolista selecciona "Entrenamientos programados".
     + El sistema muestra una lista de los entrenamientos programados con fecha y hora.
   * **Flujo alternativo**:
     + Si no hay entrenamientos programados, el sistema muestra un mensaje indicando que no hay entrenamientos para las fechas seleccionadas.
4. **Registrarse en un entrenamiento**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista se inscribe en un entrenamiento específico.
   * **Flujo básico**:
     + El futbolista selecciona un entrenamiento programado.
     + El sistema muestra los detalles del entrenamiento.
     + El futbolista hace clic en "Registrarse".
     + El sistema confirma la inscripción y agrega el entrenamiento a su calendario.
   * **Flujo alternativo**:
     + Si el futbolista ya está registrado en el entrenamiento, el sistema muestra un mensaje indicando que ya está inscrito.
5. **Ver partidos programados**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista consulta los partidos programados de su equipo.
   * **Flujo básico**:
     + El futbolista selecciona "Partidos programados".
     + El sistema muestra los partidos próximos con detalles de fecha, hora y ubicación.
   * **Flujo alternativo**:
     + Si no hay partidos programados, el sistema muestra un mensaje indicando que no hay partidos programados.
6. **Consultar detalles de un partido**
   * **Actor**: Futbolista
   * **Descripción**: El futbolista consulta los detalles de un partido específico.
   * **Flujo básico**:
     + El futbolista selecciona un partido programado.
     + El sistema muestra detalles del partido, como alineación, hora, ubicación, etc.
   * **Flujo alternativo**:
     + Si el partido no tiene detalles disponibles, el sistema muestra un mensaje indicando que los detalles aún no han sido cargados.

### **Técnico**

1. **Planificar entrenamientos**
   * **Actor**: Técnico
   * **Descripción**: El técnico programa entrenamientos para el equipo.
   * **Flujo básico**:
     + El técnico inicia sesión en el sistema.
     + El técnico selecciona "Planificar entrenamiento".
     + El sistema muestra un formulario para ingresar detalles del entrenamiento.
     + El técnico completa el formulario y confirma.
     + El sistema guarda el entrenamiento en el calendario.
   * **Flujo alternativo**:
     + Si el técnico no completa todos los campos obligatorios, el sistema muestra un mensaje de error indicando los campos faltantes.
2. **Registrar entrenamientos**
   * **Actor**: Técnico
   * **Descripción**: El técnico registra entrenamientos ya realizados.
   * **Flujo básico**:
     + El técnico selecciona "Registrar entrenamiento".
     + El sistema muestra un formulario para ingresar detalles del entrenamiento.
     + El técnico ingresa los datos y confirma.
     + El sistema guarda el registro de entrenamiento.
   * **Flujo alternativo**:
     + Si los datos ingresados son inválidos (por ejemplo, fecha incorrecta), el sistema muestra un mensaje de error y permite corregir la información.
3. **Consultar entrenamientos programados**
   * **Actor**: Técnico
   * **Descripción**: El técnico consulta los entrenamientos programados.
   * **Flujo básico**:
     + El técnico selecciona "Entrenamientos programados".
     + El sistema muestra una lista de entrenamientos programados.
   * **Flujo alternativo**:
     + Si no hay entrenamientos programados, el sistema muestra un mensaje indicando que no hay entrenamientos programados para las fechas solicitadas.
4. **Asignar futbolistas a entrenamientos**
   * **Actor**: Técnico
   * **Descripción**: El técnico asigna futbolistas a entrenamientos específicos.
   * **Flujo básico**:
     + El técnico selecciona un entrenamiento.
     + El sistema muestra una lista de futbolistas disponibles.
     + El técnico selecciona los futbolistas a asignar.
     + El sistema asigna los futbolistas y actualiza el registro.
   * **Flujo alternativo**:
     + Si un futbolista ya está asignado a un entrenamiento, el sistema muestra un mensaje indicando que ya está registrado en ese entrenamiento.
5. **Analizar estadísticas de futbolistas**
   * **Actor**: Técnico
   * **Descripción**: El técnico consulta las estadísticas de los jugadores.
   * **Flujo básico**:
     + El técnico selecciona "Estadísticas de futbolistas".
     + El sistema muestra las estadísticas detalladas de los jugadores.
   * **Flujo alternativo**:
     + Si no existen estadísticas registradas para algún jugador, el sistema muestra un mensaje indicando que no hay estadísticas disponibles para ese jugador.
6. **Registrar observaciones sobre el rendimiento**
   * **Actor**: Técnico
   * **Descripción**: El técnico registra observaciones sobre el rendimiento de los futbolistas.
   * **Flujo básico**:
     + El técnico selecciona un futbolista.
     + El sistema muestra los detalles del futbolista.
     + El técnico agrega las observaciones.
     + El sistema guarda las observaciones.
   * **Flujo alternativo**:
     + Si el técnico no agrega ningún texto en las observaciones, el sistema muestra un mensaje de advertencia y solicita ingresar al menos un comentario.

### **Organizador de Liga**

1. **Programar partidos**
   * **Actor**: Organizador
   * **Descripción**: El organizador programa los partidos de la liga.
   * **Flujo básico**:
     + El organizador selecciona "Programar partido".
     + El sistema muestra un formulario para ingresar detalles del partido.
     + El organizador ingresa los detalles y confirma.
     + El sistema guarda el partido en el calendario.
   * **Flujo alternativo**:
     + Si el organizador ingresa una fecha u hora de un partido que ya está programado, el sistema muestra un mensaje informando sobre el conflicto y permite reprogramar.
2. **Modificar programación de partidos**
   * **Actor**: Organizador
   * **Descripción**: El organizador modifica la programación de partidos.
   * **Flujo básico**:
     + El organizador selecciona "Modificar partido".
     + El sistema muestra los partidos programados.
     + El organizador selecciona el partido a modificar.
     + El organizador ajusta los detalles y confirma.
     + El sistema guarda los cambios.
   * **Flujo alternativo**:
     + Si no hay cambios realizados en la programación, el sistema muestra un mensaje indicando que no se han hecho modificaciones.
3. **Asignar equipos a un partido**
   * **Actor**: Organizador
   * **Descripción**: El organizador asigna equipos a los partidos programados.
   * **Flujo básico**:
     + El organizador selecciona un partido.
     + El sistema muestra los equipos disponibles.
     + El organizador asigna los equipos.
     + El sistema actualiza el partido con los equipos asignados.
   * **Flujo alternativo**:
     + Si un equipo ya está asignado a otro partido en la misma fecha, el sistema muestra un mensaje de advertencia y permite reprogramar.
4. **Registrar resultados de partidos**
   * **Actor**: Organizador
   * **Descripción**: El organizador registra los resultados de los partidos.
   * **Flujo básico**:
     + El organizador selecciona "Registrar resultados".
     + El sistema muestra los partidos concluidos.
     + El organizador ingresa los resultados y confirma.
     + El sistema guarda los resultados y actualiza la clasificación.
   * **Flujo alternativo**:
     + Si los resultados ingresados son incorrectos (por ejemplo, si los goles no corresponden a los equipos), el sistema muestra un mensaje de error y permite corregir los resultados.
5. **Consultar historial de partidos**
   * **Actor**: Organizador
   * **Descripción**: El organizador consulta el historial de partidos de la liga.
   * **Flujo básico**:
     + El organizador selecciona "Historial de partidos".
     + El sistema muestra los partidos pasados con detalles y resultados.
   * **Flujo alternativo**:
     + Si no hay partidos registrados en el historial, el sistema muestra un mensaje informando que no se han jugado partidos aún.

### **Administrador**

1. **Registrar nuevos usuarios**
   * **Actor**: Administrador
   * **Descripción**: El administrador registra nuevos usuarios en el sistema.
   * **Flujo básico**:
     + El administrador selecciona "Registrar usuario".
     + El sistema muestra un formulario de registro.
     + El administrador ingresa los datos del usuario y confirma.
     + El sistema guarda el usuario en la base de datos.
   * **Flujo alternativo**:
     + Si los datos ingresados son inválidos (por ejemplo, un correo incorrecto), el sistema muestra un mensaje de error y permite corregir los datos.
2. **Modificar información de usuarios**
   * **Actor**: Administrador
   * **Descripción**: El administrador modifica la información de los usuarios registrados.
   * **Flujo básico**:
     + El administrador selecciona "Modificar usuario".
     + El sistema muestra una lista de usuarios.
     + El administrador selecciona un usuario y modifica sus datos.
     + El sistema guarda los cambios.
   * **Flujo alternativo**:
     + Si no se realiza ningún cambio en los datos, el sistema muestra un mensaje informando que no se han hecho modificaciones.
3. **Eliminar usuarios**
   * **Actor**: Administrador
   * **Descripción**: El administrador elimina usuarios del sistema.
   * **Flujo básico**:
     + El administrador selecciona "Eliminar usuario".
     + El sistema muestra una lista de usuarios registrados.
     + El administrador selecciona el usuario a eliminar.
     + El sistema confirma la eliminación y elimina al usuario.
   * **Flujo alternativo**:
     + Si el usuario a eliminar tiene datos importantes asociados, el sistema muestra un mensaje de advertencia y solicita confirmar la eliminación.
4. **Gestionar equipos**
   * **Actor:** Administrador
   * **Descripción:** El administrador gestiona los equipos en el sistema (crear, modificar o eliminar equipos).
   * **Flujo básico:**
     + El administrador selecciona "Gestionar equipos".
     + El sistema muestra una lista de equipos registrados.
     + El administrador puede seleccionar un equipo y realizar acciones como modificar o eliminar.
     + El administrador puede crear un nuevo equipo ingresando los datos correspondientes.
     + El sistema guarda los cambios o crea el nuevo equipo.
   * **Flujo alternativo:**
     + Si el administrador intenta crear un equipo con datos incompletos o inválidos, el sistema muestra un mensaje de error y permite corregir la información.
5. **Asignar roles a usuarios**
   * **Actor:** Administrador
   * **Descripción:** El administrador asigna roles a los usuarios del sistema para definir sus permisos.
   * **Flujo básico:**
     + El administrador selecciona "Asignar roles".
     + El sistema muestra una lista de usuarios registrados.
     + El administrador selecciona un usuario y asigna un rol (futbolista, técnico, organizador, etc.).
     + El sistema actualiza los permisos del usuario según el rol asignado.
   * **Flujo alternativo:**
     + Si el administrador intenta asignar un rol incompatible con el tipo de usuario (por ejemplo, asignar un rol de "futbolista" a un "técnico"), el sistema muestra un mensaje de error y solicita seleccionar un rol adecuado.
6. **Restaurar accesos y contraseñas**
   * **Actor:** Administrador
   * **Descripción:** El administrador restaura accesos y contraseñas para los usuarios que lo necesiten.
   * **Flujo básico:**
     + **El administrador selecciona "Restaurar acceso".**
     + **El sistema solicita la identificación del usuario.**
     + **El administrador ingresa la información del usuario y restablece la contraseña.**
     + **El sistema envía la nueva contraseña al correo del usuario o permite configurar una nueva.**
   * **Flujo alternativo:**
     + **Si el usuario no está registrado o la información ingresada es incorrecta, el sistema muestra un mensaje de error y solicita verificar los datos.**

**Diagrama Casos de uso (Anexo el diagrama)**

**Código en PlantUML**

@startuml

left to right direction

actor Futbolista

actor Tecnico

actor "Organizador de Liga" as Organizador

actor Administrador

rectangle Sistema {

Futbolista -- (Consultar perfil)

Futbolista -- (Ver estadísticas de rendimiento)

Futbolista -- (Consultar entrenamientos programados)

Futbolista -- (Registrarse en un entrenamiento)

Futbolista -- (Ver partidos programados)

Futbolista -- (Consultar detalles de un partido)

Tecnico -- (Planificar entrenamientos)

Tecnico -- (Registrar entrenamientos)

Tecnico -- (Consultar entrenamientos programados)

Tecnico -- (Asignar futbolistas a entrenamientos)

Tecnico -- (Analizar estadísticas de futbolistas)

Tecnico -- (Registrar observaciones sobre el rendimiento)

Organizador -- (Programar partidos)

Organizador -- (Modificar programación de partidos)

Organizador -- (Asignar equipos a un partido)

Organizador -- (Registrar resultados de partidos)

Organizador -- (Consultar historial de partidos)

Organizador -- (Gestionar horarios y ubicaciones)

Administrador -- (Registrar nuevos usuarios)

Administrador -- (Modificar información de usuarios)

Administrador -- (Eliminar usuarios)

Administrador -- (Gestionar equipos)

Administrador -- (Asignar roles a usuarios)

Administrador -- (Restaurar accesos y contraseñas)

}

@enduml

**Modelo del Dominio (Anexo el diagrama)**

**Código en PlantUML**

@startuml

' Definición de las clases/entidades

class Persona {

}

class Futbolista extends Persona {

}

class Tecnico extends Persona {

}

class OrganizadorLiga extends Persona {

}

class Administrador extends Persona {

}

class Equipo {

}

class Entrenamiento {

}

class Partido {

}

class Liga {

}

class Estadistica {

}

Futbolista --\* Estadistica : registra (has-a)

Futbolista \*-- Equipo : pertenece (has-a)

Tecnico --\* Equipo : dirige (has-a)

Equipo --\* Partido : participa (has-a)

Liga --\* Equipo : tiene (has-a)

Entrenamiento --\* Futbolista : asisten (has-a)

Tecnico --\* Entrenamiento : supervisa (has-a)

Partido \*-- Equipo : enfrenta (has-a)

Liga --\* Partido : organiza (has-a)

@enduml

**Diagrama de Robustez (Anexo el diagrama)**

@startuml

actor Futbolista

actor Tecnico

actor Organizador

actor Administrador

entity "Futbolista" as FutbolistaEntity

entity "Equipo" as Equipo

entity "Entrenamiento" as Entrenamiento

entity "Partido" as Partido

entity "Liga" as Liga

entity "Estadística" as Estadistica

boundary "Interfaz Usuario Futbolista" as UI\_Futbolista

boundary "Interfaz Usuario Técnico" as UI\_Tecnico

boundary "Interfaz Usuario Organizador" as UI\_Organizador

boundary "Interfaz Usuario Administrador" as UI\_Administrador

control "Controlador Entrenamiento" as CtrlEntrenamiento

control "Controlador Partido" as CtrlPartido

control "Controlador Estadística" as CtrlEstadistica

control "Controlador Usuario" as CtrlUsuario

Futbolista --> UI\_Futbolista : Consultar perfil

UI\_Futbolista --> CtrlUsuario : Obtener datos del futbolista

CtrlUsuario --> FutbolistaEntity : Consultar datos

Futbolista --> UI\_Futbolista : Ver estadísticas de rendimiento

UI\_Futbolista --> CtrlEstadistica : Consultar estadísticas

CtrlEstadistica --> Estadistica : Obtener datos de rendimiento

Tecnico --> UI\_Tecnico : Planificar entrenamientos

UI\_Tecnico --> CtrlEntrenamiento : Crear nuevo entrenamiento

CtrlEntrenamiento --> Entrenamiento : Registrar sesión

Organizador --> UI\_Organizador : Programar partidos

UI\_Organizador --> CtrlPartido : Crear partido

CtrlPartido --> Partido : Registrar datos del partido

Administrador --> UI\_Administrador : Registrar nuevos usuarios

UI\_Administrador --> CtrlUsuario : Crear cuenta de usuario

CtrlUsuario --> FutbolistaEntity : Guardar datos

@enduml

### **Diagrama de Secuencia** **(Anexo el diagrama)**

@startuml

actor Futbolista

actor Tecnico

actor OrganizadorLiga

actor Administrador

boundary PantallaPerfil

boundary PantallaEntrenamiento

boundary PantallaPartido

boundary PantallaGestionUsuarios

control ControladorPerfil

control ControladorEntrenamiento

control ControladorPartido

control ControladorGestionUsuarios

entity Estadisticas

entity Entrenamiento

entity Partido

entity Equipos

entity Usuarios

Futbolista -> PantallaPerfil : Accede

PantallaPerfil -> ControladorPerfil : Solicita información

ControladorPerfil -> Estadisticas : Recupera datos

ControladorPerfil -> PantallaPerfil : Muestra información

Tecnico -> PantallaEntrenamiento : Ingresa detalles

PantallaEntrenamiento -> ControladorEntrenamiento : Envía datos

ControladorEntrenamiento -> Entrenamiento : Guarda información

ControladorEntrenamiento -> PantallaEntrenamiento : Confirma registro

OrganizadorLiga -> PantallaPartido : Accede

PantallaPartido -> ControladorPartido : Solicita programación

ControladorPartido -> Equipos : Verifica disponibilidad

ControladorPartido -> Partido : Guarda programación

ControladorPartido -> PantallaPartido : Muestra confirmación

Futbolista -> PantallaPartido : Consulta información

PantallaPartido -> ControladorPartido : Envía solicitud

ControladorPartido -> Partido : Obtiene detalles

ControladorPartido -> PantallaPartido : Retorna información

Tecnico -> PantallaEntrenamiento : Consulta sesiones

PantallaEntrenamiento -> ControladorEntrenamiento : Envía solicitud

ControladorEntrenamiento -> Entrenamiento : Recupera datos

ControladorEntrenamiento -> PantallaEntrenamiento : Muestra información

Administrador -> PantallaGestionUsuarios : Accede

PantallaGestionUsuarios -> ControladorGestionUsuarios : Solicita gestión de usuarios

ControladorGestionUsuarios -> Usuarios : Agrega/Modifica usuarios

ControladorGestionUsuarios -> PantallaGestionUsuarios : Confirma cambios

@enduml

**Diagrama de clases (Anexo el diagrama)**

@startuml

class Persona {

- id: int

- nombre: string

- direccion: string

+ obtenerInformacion(): string

}

class Futbolista extends Persona {

+ consultarEstadisticas(): Estadistica

}

class Tecnico extends Persona {

+ planificarEntrenamiento(): void

+ analizarEstadisticas(): void

}

class OrganizadorLiga extends Persona {

+ programarPartidos(): void

+ modificarProgramacion(): void

}

class Administrador extends Persona {

+ registrarUsuario(): void

+ modificarUsuario(): void

+ eliminarUsuario(): void

}

class Equipo {

- id: int

- nombre: string

+ agregarJugador(futbolista: Futbolista): void

}

class Entrenamiento {

- id: int

- fecha: Date

- duracion: int

+ registrarAsistencia(futbolista: Futbolista): void

}

class Partido {

- id: int

- fecha: Date

- resultado: string

+ registrarResultado(resultado: string): void

}

class Liga {

- id: int

- nombre: string

+ agregarEquipo(equipo: Equipo): void

}

class Estadistica {

- id: int

- goles: int

- asistencias: int

- tarjetas: int

+ calcularRendimiento(): double

}

Futbolista --\* Estadistica : "1" registra "\*"

Futbolista \*-- Equipo : "\*" pertenece "1"

Tecnico --\* Equipo : "1" dirige "\*"

Equipo --\* Partido : "\*" participa "2"

Liga --\* Equipo : "\*" tiene "\*"

Entrenamiento --\* Futbolista : "\*" asisten "\*"

Tecnico --\* Entrenamiento : "1" supervisa "\*"

Partido \*-- Equipo : "2" enfrenta "1"

Liga --\* Partido : "\*" organiza "\*"

@enduml