

PPIII: Diseño y Arquitectura de Sistemas



FixtureAR

Profesora: Patricia Litovicius

Estudiantes (Grupo 5):

- **Diextra Castillo, Christian**
- **Nadalin, Maximiliano**
- **Rivoldini, Maria Victoria**
- **Airoldi, Rafael Maria**

Índice

1. Descripción del Proyecto
2. Objetivos
3. Alcance
4. Análisis de Usuarios y Experiencia de Usuario (UX)
5. Casos de Uso (UML)
6. Especificación de Requerimientos
7. Recursos Disponibles
8. Plan de Trabajo
9. Gestión de Riesgos
10. Diseño del Sistema
11. Conclusiones

1. Descripción del Proyecto

1.1 Contexto institucional y problema detectado

FixtureAR es una aplicación web desarrollada con el objetivo de **simplificar y modernizar la organización de torneos en ámbitos educativos y deportivos**. Observamos que, en la mayoría de los casos, estas actividades se gestionan mediante planillas improvisadas, mensajes informales y procesos manuales que suelen generar errores y dificultan la comunicación entre los participantes.

A partir de ese diagnóstico, definimos **una solución que permite crear torneos de manera guiada, generar automáticamente el fixture según la cantidad de equipos y publicar toda la información mediante un enlace accesible desde cualquier dispositivo**. De esta manera, docentes, entrenadores, jugadores y familias pueden mantenerse informados de forma clara, unificada y sin necesidad de instalar aplicaciones adicionales.

1.2 Tipo de sistema cliente-servidor propuesto

Para responder a esa necesidad proponemos una **aplicación web cliente-servidor**.

- Del lado del **cliente**, los usuarios acceden desde un navegador (en computadora o dispositivo móvil) a una interfaz web donde pueden crear torneos, visualizar fixtures o consultar información según su perfil.
- Del lado del **servidor**, se concentra la lógica de negocio: creación y almacenamiento de torneos, generación automática del fixture y exposición de la información a través de enlaces públicos.
- Una **base de datos** central almacena equipos, partidos y demás entidades necesarias para el funcionamiento del sistema.

Elegimos este tipo de arquitectura porque se adapta bien al entorno educativo, no requiere instalación en los dispositivos y permite que el sistema crezca en funcionalidad sin afectar la forma en que los usuarios acceden.

1.3 Actores clave del sistema

En FixtureAR identificamos tres actores principales:

- **Organizador (Manager)**: docentes, coordinadores o entrenadores responsables de crear y administrar torneos, configurar parámetros básicos y publicar el fixture.
- **Jugador (Participante)**: personas que forman parte de los equipos y consultan sus próximos partidos o la estructura del torneo. En algunos casos también pueden crear partidos rápidos.

- **Invitado (Público):** familiares, estudiantes y miembros de la comunidad que sólo necesitan consultar la información del torneo a través de un enlace público, sin registrarse.

Cada uno de estos actores tiene necesidades distintas y niveles de interacción diferentes con el sistema, lo que se refleja tanto en el diseño de la interfaz como en los casos de uso.

1.4 Análisis FODA del proyecto

A partir del análisis del contexto y la propuesta de solución, elaboramos el siguiente FODA del sistema FixtureAR:

Fortalezas

- Interfaz web accesible desde cualquier dispositivo con navegador.
- Generación automática del fixture, reduciendo errores manuales.
- Publicación mediante enlace público, sin necesidad de que el público se registre.
- Enfoque específico en el entorno educativo y deportivo amateur.

Oportunidades

- Posibilidad de adopción en colegios, clubes y ligas locales que hoy no cuentan con herramientas similares.
- Escalabilidad hacia versiones futuras con estadísticas, notificaciones y otras funcionalidades avanzadas.
- Potencial para ofrecer planes institucionales o comerciales en un futuro.

Debilidades

- Alcance inicial acotado en cuanto a funcionalidades (MVP).
- Dependencia de conexión a internet para acceder al fixture.
- Necesidad de una mínima familiaridad con herramientas digitales por parte de los organizadores.

Amenazas

- Uso arraigado de planillas tradicionales y resistencia al cambio en algunos entornos.
- Existencia de aplicaciones externas que pueden cubrir parte de la misma necesidad.
- Posibles limitaciones de recursos para mantener y evolucionar la plataforma a largo plazo.

2. Objetivos

El desarrollo de FixtureAR se orienta a brindar una herramienta accesible y eficiente para la organización de torneos en contextos educativos y deportivos. **Nuestro objetivo general es ofrecer un sistema que permita crear, gestionar y visualizar fixtures de manera simple, ordenada y desde cualquier dispositivo.**

A partir de este propósito, establecimos los siguientes objetivos específicos:

- **Simplificar la creación de torneos**, ofreciendo un proceso guiado que facilite la configuración inicial sin necesidad de conocimientos técnicos.
- **Generar automáticamente el fixture** en función de la cantidad de equipos y la modalidad elegida, reduciendo errores y tiempos de preparación.
- **Permitir la visualización pública del torneo** mediante un enlace accesible sin registro, para que jugadores, familias y asistentes puedan consultar la información de manera rápida y clara.
- **Incorporar la posibilidad de crear partidos rápidos**, una funcionalidad pensada para encuentros informales o de práctica que no requieren la estructura de un torneo completo.
- **Garantizar una experiencia de uso intuitiva**, con una interfaz limpia, ordenada y consistente que facilite la adopción del sistema en el ámbito escolar y comunitario.

En esta versión del proyecto nos propusimos concentrarnos en estas funcionalidades esenciales, con la intención de construir una base sólida sobre la cual puedan incorporarse futuras mejoras y ampliaciones.

3. Alcance

Al definir el alcance del proyecto buscamos encontrar un equilibrio entre las necesidades reales que detectamos y el tiempo y complejidad razonables para esta instancia de la materia. Por ese motivo, optamos por concentrarnos en una primera versión funcional (MVP) que resuelva el problema principal: organizar y comunicar torneos de manera clara y accesible.

3.1 Alcance de la versión actual (MVP)

En esta versión de FixtureAR nos enfocamos en las siguientes funcionalidades:

- **Creación de torneos con una cantidad acotada de equipos:** el sistema permite configurar torneos con hasta 8 equipos, lo que resulta suficiente para el tipo de competencias que suelen darse en colegios y clubes de pequeña escala.
- **Generación automática del fixture:** una vez definidos los datos básicos del torneo (equipos y modalidad), el sistema genera el calendario de partidos de forma automática, evitando la construcción manual de cruces.
- **Edición de datos básicos:** el organizador puede ajustar información relevante del torneo, como nombre, fechas, horarios y lugar, sin necesidad de rehacer todo el proceso.
- **Publicación mediante enlace público:** el fixture puede consultarse a través de un enlace accesible desde cualquier navegador, sin requerir registro para el público general.
- **Visualización según distintos perfiles:** tanto jugadores como invitados pueden acceder a la información del torneo, consultar próximos partidos y comprender la estructura general de la competencia.
- **Creación de partidos rápidos:** además del formato de torneo, contemplamos la posibilidad de generar encuentros puntuales o informales, pensados para prácticas o amistosos que no requieren una estructura completa.

Estas funcionalidades conforman el núcleo de la propuesta y responden directamente al problema detectado: la falta de una herramienta simple y centralizada para organizar y comunicar fixtures.

3.2 Funcionalidades fuera de alcance en esta etapa

Al mismo tiempo, identificamos una serie de características que consideramos valiosas, pero que decidimos dejar fuera de esta primera versión para no comprometer la calidad del diseño ni la claridad del entregable:

- **Actualización de resultados en tiempo real y tabla de posiciones dinámica:** si bien es una mejora natural para futuras iteraciones, en esta etapa el foco está puesto en la organización y comunicación del fixture, no en la gestión integral de estadísticas.
- **Gestión de inscripciones y asistencia:** funcionalidades como registro de jugadores, confirmación de presencia o control de cupos se consideran una posible extensión del sistema, pero escapan al alcance del MVP.
- **Estadísticas detalladas por jugador o equipo:** métricas de rendimiento, historial de partidos y comparativas quedarían reservadas para una versión más avanzada.
- **Notificaciones automáticas (correo, mensajería, etc.):** el envío automático de recordatorios o avisos de cambios es una mejora interesante, pero requiere infraestructura y lógica adicional que exceden el objetivo de esta etapa.

Al dejar explícitas estas exclusiones, buscamos mostrar que el proyecto se limita por una decisión consciente de acotar el alcance para garantizar un diseño consistente y viable.

3.3 Posible estrategia de monetización futura

Si bien el proyecto se plantea inicialmente para el contexto educativo y comunitario, consideramos que FixtureAR podría evolucionar hacia un modelo de uso comercial en versiones futuras. En ese escenario, la incorporación de integraciones con pasarelas de pago u otros sistemas externos podría habilitar distintos esquemas de monetización, por ejemplo:

- Cobro de una **suscripción institucional** para colegios, clubes o ligas que quieran gestionar varios torneos en simultáneo.
- Venta de **planes ampliados** con acceso a funcionalidades avanzadas, como estadísticas detalladas, notificaciones automáticas o reportes personalizados.
- Integración con servicios externos que permitan **cobrar inscripciones** o cuotas vinculadas a torneos específicos.

Estas posibilidades no forman parte del alcance del MVP ni del desarrollo técnico de esta entrega, pero se incluyen como una línea de proyección futura. La intención es dejar documentado que el diseño conceptual del sistema admite una extensión natural hacia modelos de negocio sostenibles, sin alterar la esencia de la propuesta original.

4. Análisis de Usuarios y Experiencia de Usuario (UX)

4.1 Perfiles de usuarios

A partir del contexto de uso del sistema identificamos tres perfiles principales:

- **Organizador (Manager):** docentes, coordinadores o entrenadores que crean y administran torneos. Son quienes configuran los parámetros del torneo, generan el fixture y lo comparten con la comunidad.
- **Jugador (Participante):** estudiantes o jugadores que consultan sus partidos, resultados y, en algunos casos, crean partidos rápidos para encuentros informales.
- **Invitado (Público):** familiares, compañeros y otras personas interesadas que acceden al fixture únicamente para visualizar información, sin necesidad de registrarse.

Estos perfiles nos permitieron orientar la interfaz y las funcionalidades según el tipo de interacción esperado para cada uno.

4.2 Personas y escenarios de uso

Para profundizar en las necesidades reales de estos perfiles, construimos tres **personas** basadas en situaciones típicas del entorno educativo y deportivo. Estas personas guiaron muchas de las decisiones de diseño tomadas durante el desarrollo del prototipo:

- **Marcelo Maidana – Organizador**

Profesor de Educación Física y coordinador en un club barrial. Organiza torneos utilizando planillas y grupos de mensajería, lo que le exige tiempo extra y lo obliga a corregir errores manuales con frecuencia.

Escenario: Marcelo utiliza FixtureAR para crear un torneo, cargar los equipos y generar el fixture automático. Luego comparte el enlace con los jugadores y las familias, evitando confusiones por versiones distintas del calendario.

- **Matías Fernández – Jugador**

Estudiante de 12 años que participa en los torneos de la escuela. Usa principalmente el celular y valora poder consultar horarios de forma rápida y visual.

Escenario: Matías entra a FixtureAR desde un enlace, filtra por su equipo y consulta el próximo partido sin depender de mensajes en el grupo ni de explicaciones adicionales.

- **Florencia Rodríguez – Invitada**

Madre de dos jugadores, trabaja en horario administrativo y organiza la agenda familiar. Prefiere interfaces ordenadas y claras.

Escenario: Florencia accede al fixture desde el enlace enviado por la escuela, revisa los partidos del fin de semana y planifica traslados sin necesidad de registrarse ni navegar por menús complejos.

Estas personas nos ayudaron a evaluar si las pantallas del prototipo respondían a problemas concretos y no solo a una idea abstracta de “usuario final”.

4.3 Wireframes, mockups y prototipo

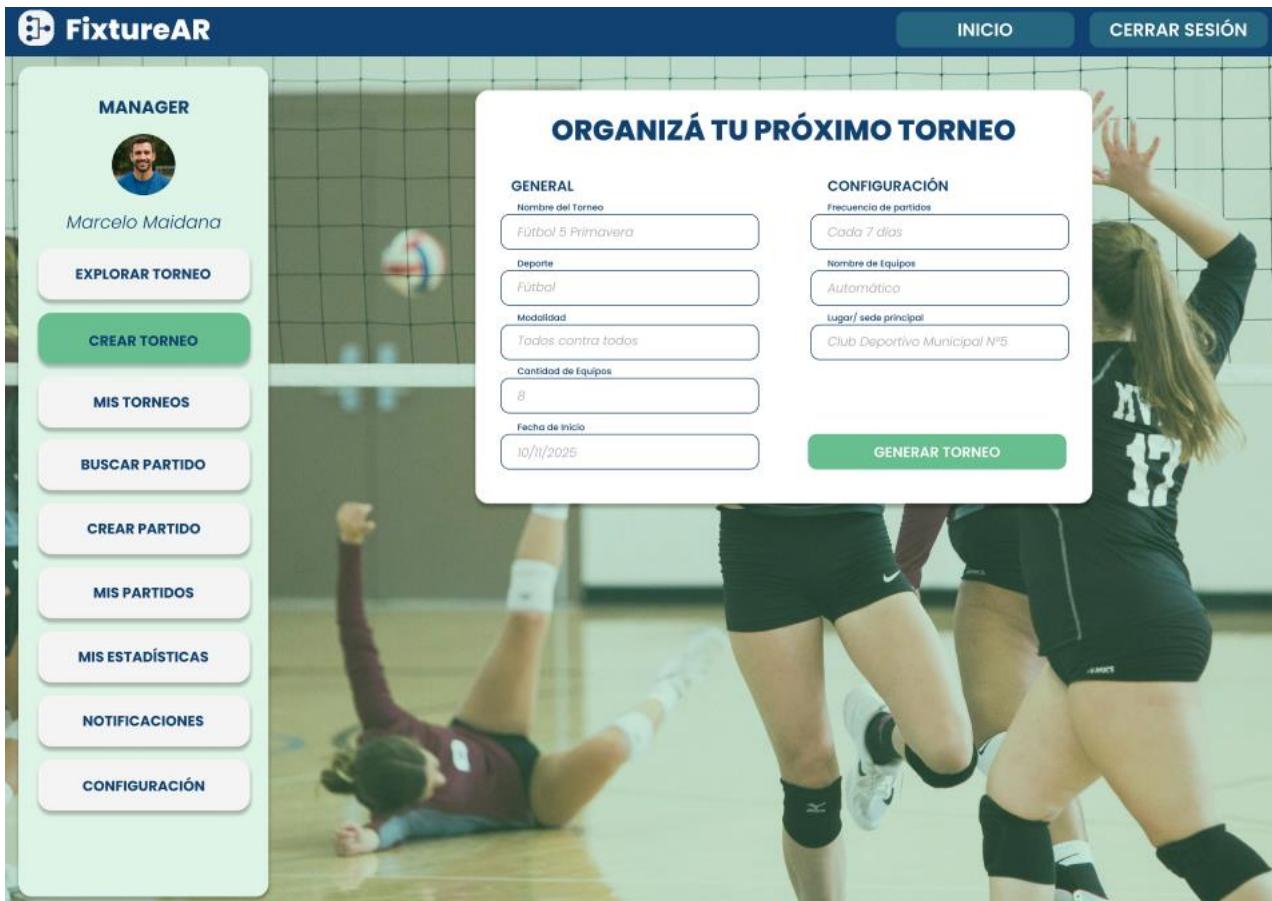
En función de los perfiles y escenarios definidos, desarrollamos una serie de **wireframes** y **mockups** que luego se transformaron en un prototipo navegable en Figma.

Entre las pantallas principales se destacan:

- La **pantalla de inicio**, donde se presentan las acciones clave del sistema: crear un torneo, iniciar sesión, registrarse y ver una demo.
- La **pantalla de configuración de torneo** para el Organizador, con campos agrupados por bloques lógicos (datos generales, modalidad, cantidad de equipos, fechas, sede).
- La **pantalla de creación de partida rápida** para el Jugador, con un formulario simplificado pensado para encuentros informales.
- Las **pantallas de confirmación**, que informan con mensajes claros cuando un torneo o un partido se crean correctamente y ofrecen el enlace listo para compartir.



En la **Figura 4.1** se muestra la pantalla inicial de FixtureAR, donde se presentan las acciones principales: crear un torneo, iniciar sesión, registrarse y ver la demo. La intención fue que cualquier usuario entienda, en pocos segundos, qué ofrece la herramienta y cómo empezar.



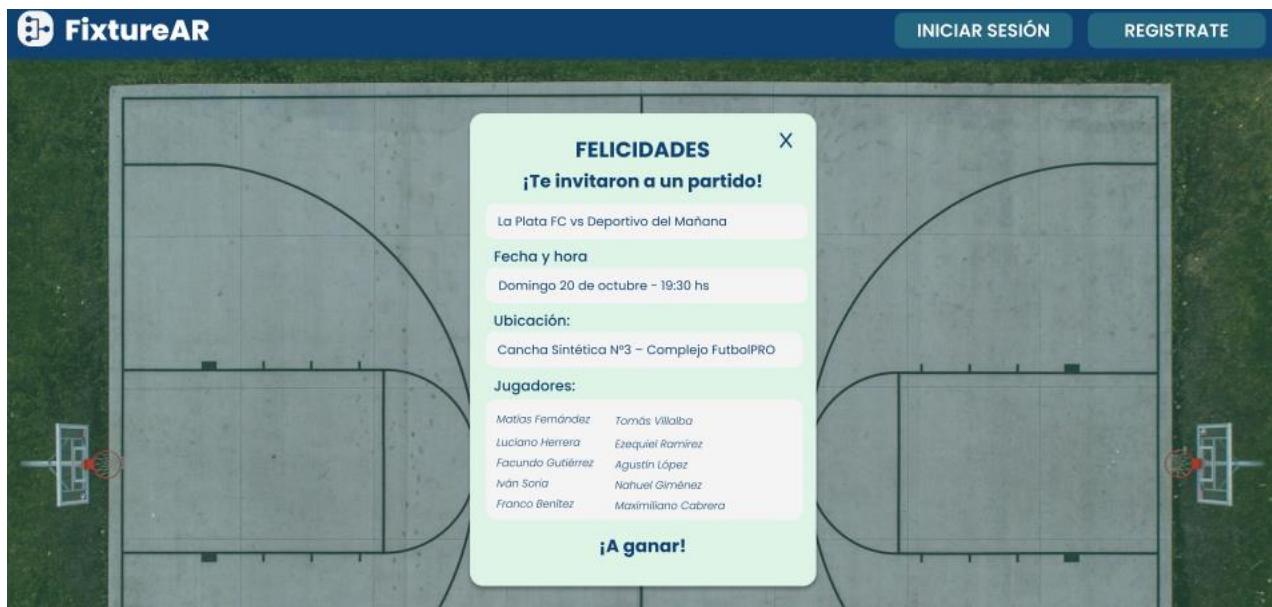
En la **Figura 4.2** se observa la pantalla de configuración de torneo del modo Manager. Allí se concentran los campos esenciales (nombre del torneo, deporte, modalidad, cantidad de equipos, fecha de inicio y lugar), organizados en secciones claras. Esta vista resume el flujo principal del organizador y materializa uno de los objetivos centrales del sistema: simplificar la creación de torneos.



La **Figura 4.3** corresponde a la creación de una partida rápida desde el modo Jugador. A diferencia de la configuración de torneo, este formulario es más acotado, pensado para encuentros informales. El diseño mantiene la misma lógica visual, pero prioriza la velocidad de carga y la simplicidad.



En la **Figura 4.4** se muestra la pantalla de confirmación de partido creado, donde se presenta el mensaje de éxito junto con el enlace listo para copiar y compartir. Esta instancia cierra el flujo del usuario y refuerza la idea de que el sistema no solo organiza la información, sino que también facilita su difusión.



En la **Figura 4.5** se muestra la pantalla de invitación a un partido. Representa un caso particular dentro del flujo general del sistema: cómo recibe e interpreta la información una persona que no es usuaria activa de FixtureAR, sino que accede únicamente a través de un enlace compartido.

El objetivo de estas pantallas fue asegurar que cada usuario pueda completar sus tareas principales con la menor fricción posible, tanto en computadora como en dispositivos móviles.

4.4 Principios de accesibilidad y diseño inclusivo

Durante la elaboración del prototipo aplicamos principios de diseño centrado en el usuario que nos ayudaron a mantener coherencia visual y claridad funcional.

Desde la perspectiva de la **Gestalt**, tuvimos en cuenta:

- **Proximidad:** agrupamos campos relacionados en secciones (por ejemplo, “Datos generales” y “Configuración del torneo”) para facilitar la lectura y la asociación mental.
- **Similitud:** utilizamos estilos consistentes en botones, campos de texto e íconos para que el usuario pueda reconocer rápidamente elementos de la misma categoría.
- **Figura y fondo:** definimos jerarquías visuales claras, destacando contenido principal sobre fondos neutros y minimizando distracciones.
- **Continuidad:** organizamos los flujos de forma que las acciones principales (crear torneo, ver demo, crear partido rápido) se perciban como un recorrido natural y previsible.

Además, consideramos varias de las **heurísticas de Nielsen**:

- **Visibilidad del estado del sistema:** incluimos mensajes de confirmación visibles (“Torneo creado con éxito”, “Partido creado con éxito”) que informan al usuario sobre el resultado de sus acciones.
- **Consistencia y estándares:** mantuvimos un estilo visual uniforme en todas las pantallas, reutilizando patrones de botones, formularios y navegación.
- **Prevención de errores:** los formularios guían al usuario paso a paso y limitan la cantidad de información cargada en cada pantalla para reducir errores.
- **Reconocer antes que recordar:** las acciones se expresan con textos claros y directos (“Crear torneo”, “Ver demo”), para que el usuario no tenga que memorizar rutas o términos complejos.
- **Diseño estético y minimalista:** optamos por una interfaz limpia, con una cantidad acotada de elementos en pantalla, priorizando lo funcional sobre lo decorativo.

Estas decisiones buscan que FixtureAR resulte comprensible incluso para usuarios que no están acostumbrados a utilizar sistemas específicos para la gestión de torneos.

Además de los principios de diseño mencionados, elaboramos una matriz de trazabilidad que relaciona a cada tipo de usuario con los casos de uso y las pantallas del prototipo. Esta matriz nos permitió verificar que cada vista del sistema respondiera a una necesidad real identificada en los perfiles y escenarios de uso, evitando funcionalidades aisladas o inconsistentes.

ANEXO 1: Funcionalidades por perfil

Para reforzar la relación entre los perfiles de usuario y las funcionalidades, elaboramos un cuadro que resume qué puede hacer cada tipo de usuario dentro del sistema.

Funcionalidad	Invitado	Jugador	Manager
Ver información de torneos públicos	Sí	Sí	Sí
Participar en partidos rápidos	Sí (por invitación)	Sí	Sí
Guardar estadísticas personales	X	Sí	Sí
Crear partidos rápidos	X	Sí	Sí
Buscar partidos en comunidades	X	Sí	Sí
Participar en torneos de comunidad	X	Sí	Sí
Crear torneos	X	X	Sí
Generar fixture automático	X	X	Sí
Cargar resultados y estadísticas de torneo	X	X	Sí
Gestionar convocatorias y asistencia	X	X	Sí
Recibir recordatorios automáticos	X	Sí	Sí
Acceder al historial completo de torneos	X	X	Sí
Compartir torneos por enlace	Sí	Sí	Sí
Crear comunidades y organizar múltiples torneos	X	X	Sí

En este cuadro se distinguen claramente las capacidades del Invitado, del Jugador y del Manager: desde visualizar torneos públicos y participar en partidos rápidos, hasta crear torneos, gestionar comunidades y acceder a estadísticas. De esta manera, se hace explícito qué rol cubre cada perfil dentro del sistema.

ANEXO 2: Matriz de trazabilidad de casos de uso

En este anexo se presenta la matriz de trazabilidad que vincula a los actores del sistema con los casos de uso definidos y las pantallas del prototipo de FixtureAR. El objetivo de esta matriz es asegurar la coherencia entre el análisis funcional, el diseño de la experiencia de usuario y las vistas concretas del sistema.

Para cada combinación de actor y caso de uso se indica la actividad principal que realiza el sistema y la pantalla asociada en la interfaz. Esto nos permite verificar que:

- cada funcionalidad tenga una representación clara en el prototipo,
- cada pantalla responda a una necesidad real de uno o más tipos de usuario,
- no existan vistas “huérfanas” sin relación con los casos de uso planteados.

Actor	Caso de Uso	Actividad del Sistema	Pantalla UX Asociada
Invitado	Visualizar torneo público	Mostrar información básica del torneo (fixture, resultados, fecha, etc.)	Ver Torneo
Invitado	Participar en partido rápido	Ver detalle del evento recibido por link. No guarda estadística ni historial	Pantalla de Partido Rápido
Jugador	Registrarse	Completar formulario, validar datos, almacenar usuario y redirigir al modo elegido	Registrarse
Jugador	Iniciar sesión	Validar credenciales y permitir acceso según el perfil	Iniciar sesión
Jugador	Crear partido rápido	Completar formulario, generar link compatible y guardar estadísticas personales	Crear Partido Rápido
Jugador	Unirse a torneo comunitario	Buscar torneos disponibles, solicitar participación, visualizar información propia	Ver Comunidad
Manager	Crear torneo	Completar configuración, validar datos, generar fixture automáticamente	Crear Torneo
Manager	Administrar torneo	Cargar resultados, editar datos, visualizar estadísticas generales	Gestión de Torneo
Manager	Gestionar comunidad	Crear espacios, invitar jugadores, asignar roles, organizar múltiples torneos	Comunidad / Dashboard

Esta matriz sirve como herramienta de validación del diseño, ya que muestra de forma explícita cómo se conectan los usuarios, los flujos del sistema y las pantallas que conforman la experiencia final.

5. Casos de Uso (UML)

Para estructurar las funcionalidades del sistema desde la perspectiva de los distintos usuarios, elaboramos un conjunto de diagramas de casos de uso que representan las acciones principales que cada actor puede realizar dentro de FixtureAR. Estos diagramas funcionan como una síntesis visual del comportamiento esperado del sistema y nos permiten verificar que las interacciones diseñadas sean coherentes con los perfiles y escenarios planteados en el análisis de usuarios.

DIAGRAMA 1 – ORGANIZADOR (MANAGER)

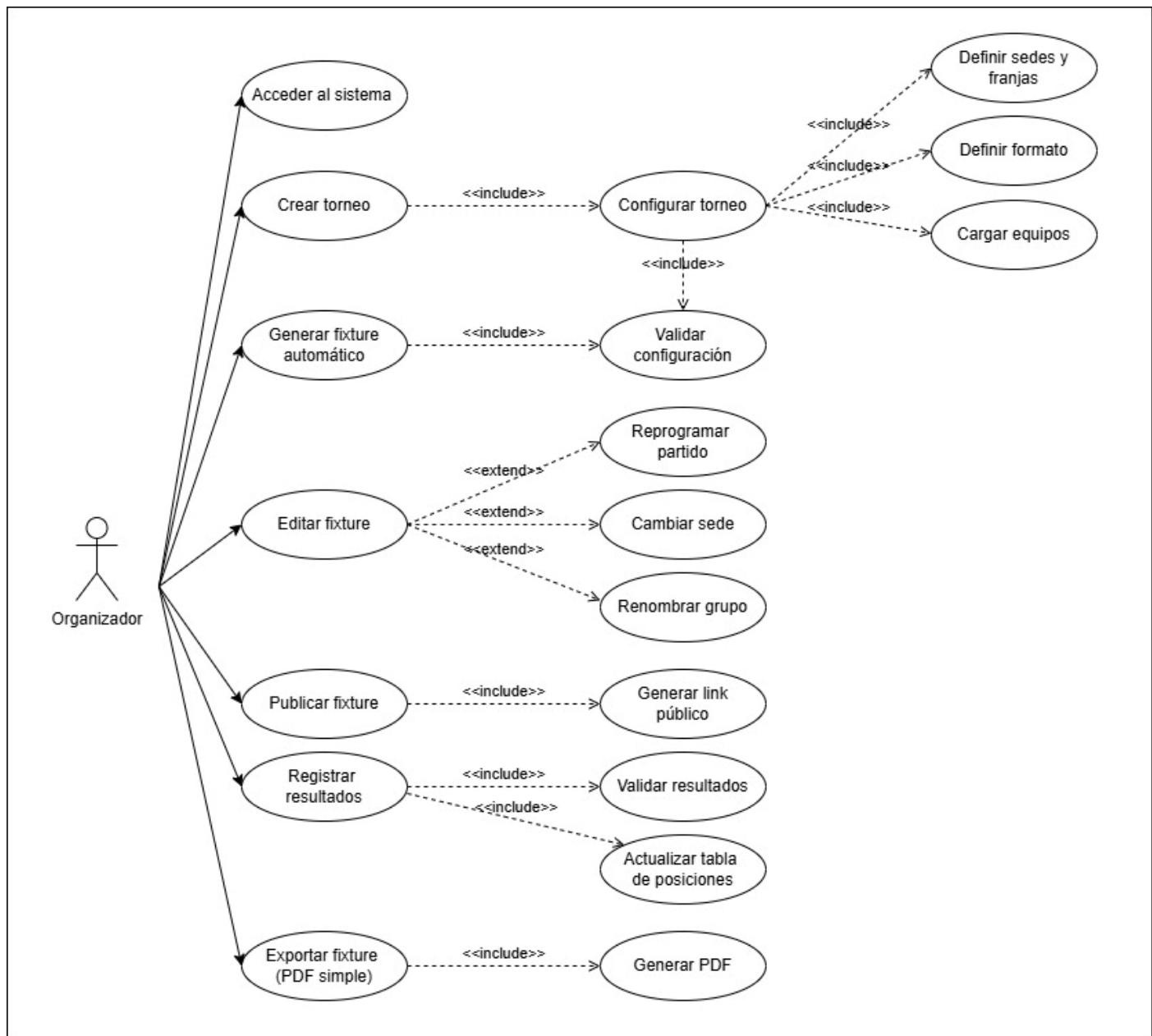


Figura 5.1 – Casos de uso del actor Organizador en FixtureAR.

El Organizador es el actor más completo dentro del sistema, y por eso su diagrama concentra la mayor cantidad de acciones. Este actor puede crear torneos, cargar equipos, configurar aspectos generales, generar el fixture automático y publicar la información mediante un enlace público. Además, administra comunidades y gestiona múltiples torneos. Su rol es clave para asegurar que la información publicada en FixtureAR sea correcta, actualizada y clara para el resto de los usuarios.

DIAGRAMA 2 – JUGADOR (PARTICIPANTE)

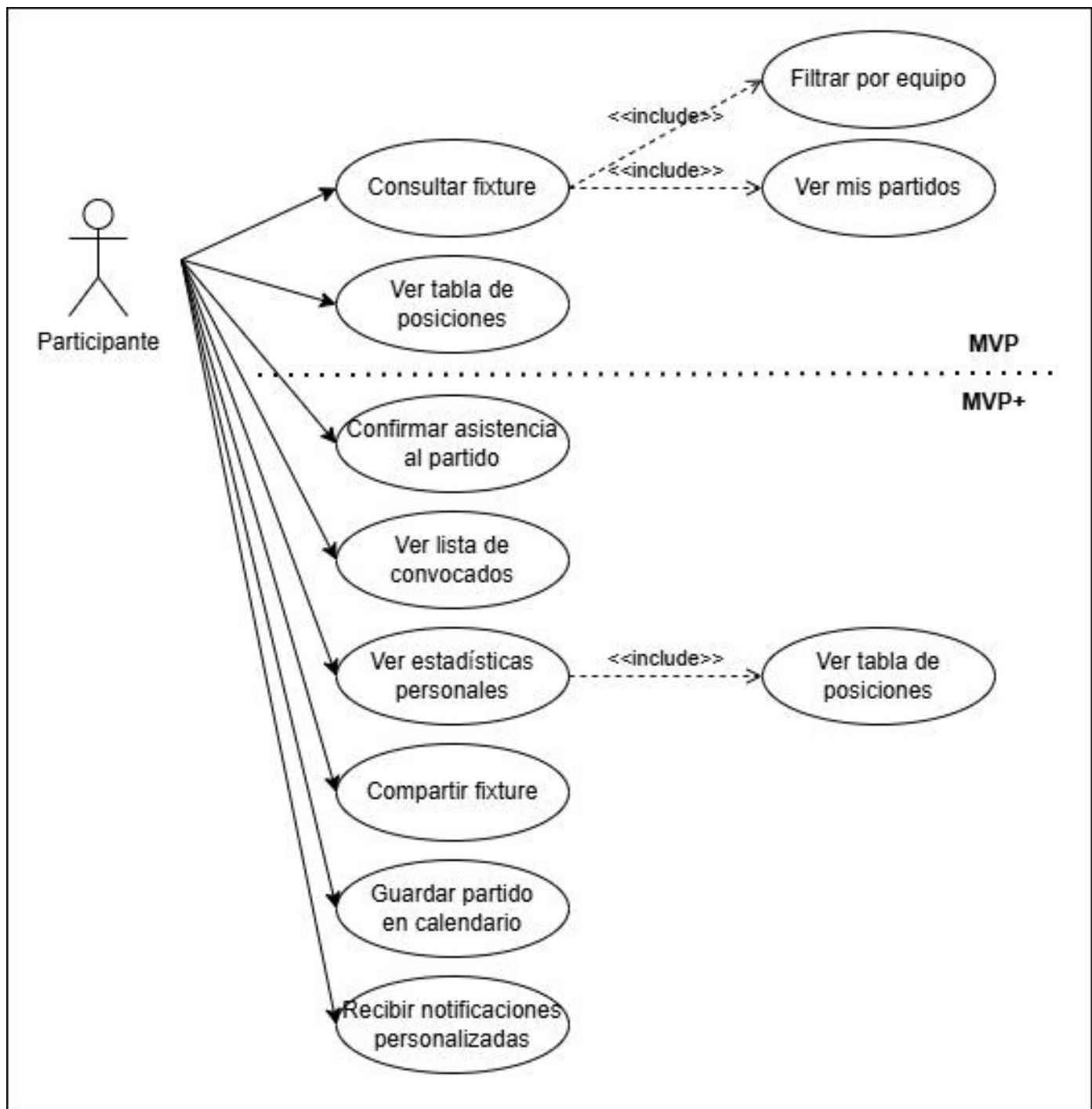


Figura 5.2 – Casos de uso del actor Jugador en FixtureAR.

En este diagrama se muestran las acciones relacionadas con los usuarios que participan activamente de los torneos o encuentros. El Jugador puede registrarse, iniciar sesión, consultar su información personal, visualizar el fixture completo y crear partidos rápidos. Su rol implica una interacción más profunda que la del Invitado, ya que además de consumir información también puede generar nuevos eventos dentro del sistema.

DIAGRAMA 3 – INVITADO (PÚBLICO)

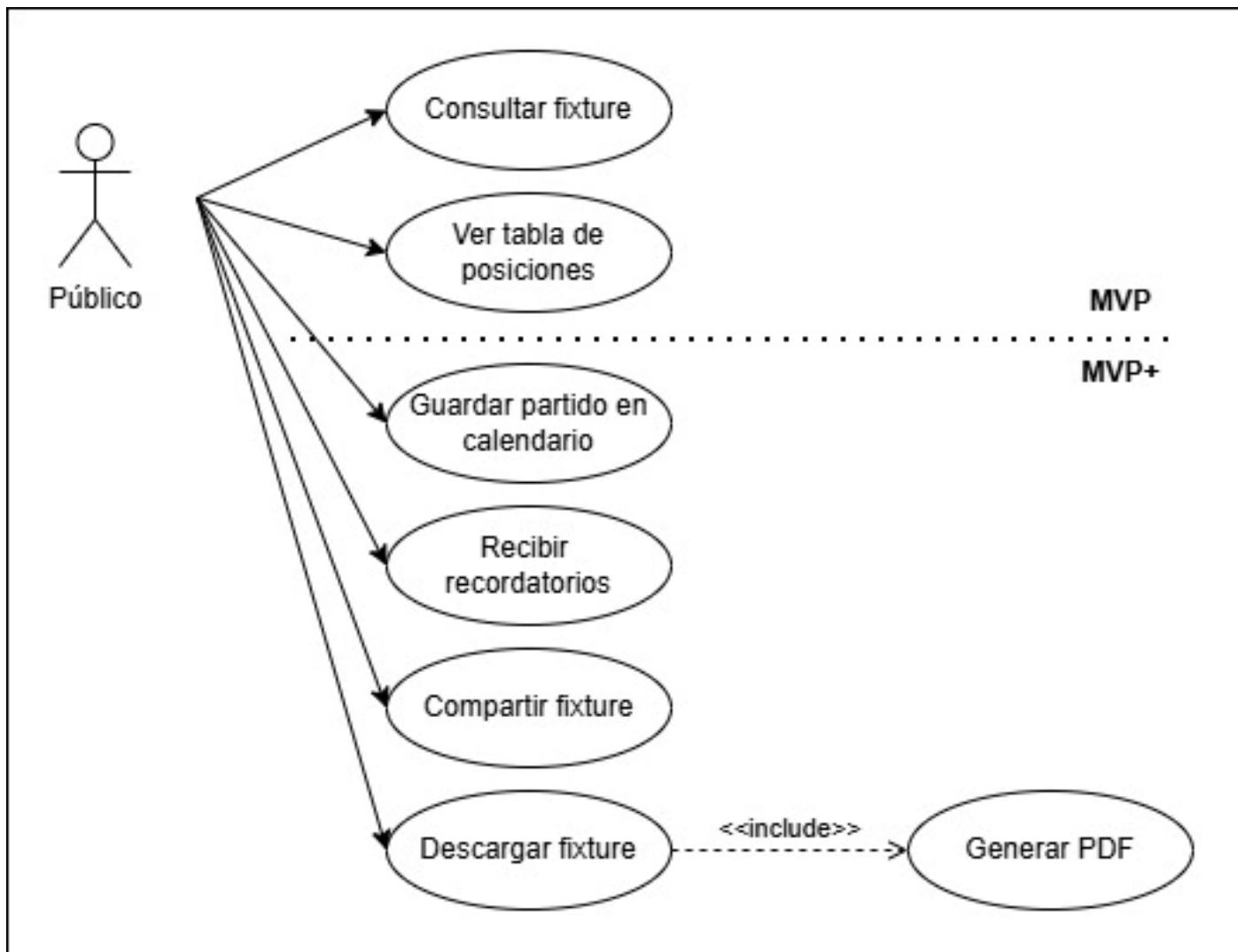


Figura 5.3 – Casos de uso del actor Invitado en FixtureAR.

Este diagrama representa las acciones disponibles para una persona que accede al sistema sin autenticarse. El Invitado puede visualizar torneos públicos, consultar partidos desde un enlace compartido y acceder a invitaciones de partidos rápidos sin necesidad de registrarse ni crear una cuenta. Su interacción es exclusivamente de consulta, acorde al objetivo de que el sistema sea accesible incluso para quienes lo utilizan de manera ocasional o puntual.

6. Especificación de Requerimientos

Los requerimientos del sistema FixtureAR surgieron a partir del análisis del problema, los perfiles de usuario y el alcance definido para la primera versión del proyecto. En esta sección se presentan, por un lado, los requerimientos funcionales que describen qué debe hacer el sistema y, por otro, los requerimientos no funcionales y de arquitectura que condicionan cómo debe hacerlo y bajo qué criterios de calidad.

6.1 Requerimientos funcionales

Los siguientes requerimientos funcionales corresponden al alcance del MVP definido para FixtureAR:

- El sistema debe permitir **crear torneos** mediante un proceso guiado, solicitando los datos básicos del evento (nombre, deporte, modalidad, cantidad de equipos, fechas, sede).
- El sistema debe **generar automáticamente el fixture** del torneo en función de la cantidad de equipos y la modalidad seleccionada.
- El sistema debe permitir **editar datos básicos del torneo** (por ejemplo, nombre, fechas, horarios y lugar) sin necesidad de crear un torneo nuevo.
- El sistema debe **publicar el torneo mediante un enlace accesible**, de modo que cualquier persona pueda visualizar la información sin registrarse.
- El sistema debe permitir que los usuarios **visualicen la información del torneo** (partidos, fechas, equipos participantes) desde distintos dispositivos.
- El sistema debe permitir a los Jugadores **registrarse y autenticarse** para acceder a funcionalidades específicas asociadas a su perfil.
- El sistema debe permitir **crear partidos rápidos**, es decir, encuentros puntuales que no forman parte de un torneo estructurado, generando un enlace para compartir la información.
- El sistema debe mostrar **mensajes de confirmación claros** cuando se crea un torneo o un partido rápido, indicando el resultado de la acción y el enlace correspondiente.
- El sistema debe permitir **compartir enlaces de torneos y partidos** con otros usuarios por fuera de la plataforma (por ejemplo, a través de mensajería externa).

6.2 Requerimientos no funcionales

Además de lo que el sistema hace, definimos una serie de criterios de calidad que consideramos importantes para el contexto educativo y deportivo donde se utilizará FixtureAR:

- El sistema debe ofrecer una **interfaz simple e intuitiva**, de modo que pueda ser utilizado por personas con poca experiencia en herramientas tecnológicas.
- El sistema debe ser **accesible desde navegadores web modernos** en computadoras y dispositivos móviles, sin requerir instalación de aplicaciones adicionales.
- El sistema debe mantener **tiempos de respuesta razonables** en las operaciones principales (creación de torneos, generación de fixture, visualización de información).
- El sistema debe implementar mecanismos básicos de **seguridad y autenticación** para proteger las cuentas de los usuarios que se registran (por ejemplo, Jugadores y Managers).
- El sistema debe ser diseñado con criterios de **escalabilidad**, de manera que pueda incorporar nuevas funcionalidades en versiones futuras sin requerir una reescritura completa.
- El sistema debe favorecer la **mantenibilidad**, utilizando una estructura clara de módulos y responsabilidades bien definidas.
- El sistema debe priorizar la **consistencia visual y de interacción**, manteniendo patrones de diseño homogéneos en las distintas pantallas para reducir la curva de aprendizaje.

6.3 Requisitos de arquitectura

En cuanto a la arquitectura, FixtureAR se apoya en un modelo simple pero lo suficientemente flexible como para acompañar la evolución del sistema:

- El sistema debe implementar una **arquitectura cliente-servidor**, en la que la interfaz de usuario se ejecute en el navegador y la lógica de negocio se procese en un servidor central.
- La solución debe organizarse en **capas diferenciadas** (presentación, lógica de negocio y acceso a datos), para separar responsabilidades y facilitar el mantenimiento.
- El sistema debe contar con una **base de datos centralizada** donde se almacenen de forma persistente los torneos, equipos, partidos, usuarios y demás entidades relevantes.
- La arquitectura debe permitir, en el futuro, la **integración con servicios externos** (por ejemplo, APIs o servicios en la nube) sin modificar la lógica principal del sistema.
- El diseño debe contemplar mecanismos básicos de **respaldo y recuperación de datos**, acordes a la criticidad del contexto educativo en el que se utilizará.

ANEXO 3: Historias de Usuario

En este anexo presentamos cinco ejemplos de User Stories derivadas de los requerimientos funcionales de FixtureAR. El objetivo es mostrar cómo podrían gestionarse estas funcionalidades en un entorno de trabajo ágil, utilizando **Scrum** y organizando el backlog a partir de historias de usuario claras y comprobables.

Para redactar las historias utilizamos el criterio **INVEST**, que propone que cada User Story sea:

- **I – Independiente:** pueda planificarse sin depender de demasiadas otras historias.
- **N – Negociable:** abierta a conversación y ajustes con el equipo.
- **V – Valiosa:** aporte un valor concreto a algún tipo de usuario.
- **E – Estimable:** se pueda dimensionar razonablemente.
- **S – Small (Pequeña):** de un tamaño adecuado para completarse dentro de un sprint.
- **T – Testable (Comprobable):** verificable a partir de criterios de aceptación claros.

A continuación, se detallan las cinco historias seleccionadas, dependencias y criterios de aceptación, y finalmente una tabla de trazabilidad que relaciona cada historia con los requerimientos funcionales y los casos de uso correspondientes.

➤ **US-01 – Crear torneo mediante asistente guiado**

Título: Creación guiada de torneos

Descripción:

Como **Organizador**, quiero **crear un torneo mediante un asistente paso a paso**, para **configurar los datos básicos de manera clara y ordenada**.

Dependencias:

- Requerimiento funcional: creación de torneos mediante proceso guiado
- Caso de uso asociado: *Manager – Crear torneo*

Criterios de aceptación:

- El asistente solicita al menos: nombre del torneo, deporte, modalidad, cantidad de equipos, fecha de inicio y sede.
- Si falta un dato obligatorio, el sistema lo informa sin borrar lo ya cargado.
- Al finalizar, el sistema crea el torneo y muestra un mensaje de confirmación.

➤ US-02 – Generación automática de fixture

Título: Generación del fixture

Descripción:

Como **Organizador**, quiero **generar automáticamente el fixture** a partir de la cantidad de equipos y la modalidad configurada, para **evitar confeccionar los cruces manualmente**.

Dependencias:

- Requerimiento funcional: generación automática del fixture
- Caso de uso asociado: *Manager – Generar fixture*

Criterios de aceptación:

- El fixture generado incluye todos los equipos sin omisiones.
- Los partidos quedan asociados al torneo correspondiente.
- Se presenta un resumen del fixture al finalizar el proceso.

➤ US-03 – Publicación del torneo mediante enlace público

Título: Publicación del torneo

Descripción:

Como **Organizador**, quiero **publicar el torneo mediante un enlace accesible**, para **compartir la información con jugadores e invitados sin necesidad de registro**.

Dependencias:

- Requerimiento funcional: publicación del torneo
- Casos de uso asociados:
 - *Manager – Publicar torneo*
 - *Invitado – Visualizar torneo*

Criterios de aceptación:

- Se genera un enlace único para cada torneo.
- Cualquier persona con el enlace puede visualizar datos del torneo sin autenticarse.
- Los partidos se muestran en orden cronológico y con información clara.

➤ US-04 – Creación de partido rápido

Título: Crear partido rápido

Descripción:

Como **Jugador**, quiero **crear un partido rápido ingresando los datos mínimos**, para **organizar encuentros informales sin necesidad de crear un torneo completo**.

Dependencias:

- Requerimiento funcional: creación de partidos rápidos

- Caso de uso asociado: *Jugador – Crear partido rápido*

Criterios de aceptación:

- Se solicita al menos: título del partido, fecha, hora y ubicación.
- El sistema crea el partido y muestra un enlace para compartirlo.
- El enlace permite visualizar los datos del partido en cualquier dispositivo.

➤ US-05 – Registro e inicio de sesión de jugadores

Título: Registro y autenticación

Descripción:

Como **Jugador**, quiero **registrarme e iniciar sesión**, para **acceder a mis partidos y funcionalidades personalizadas**.

Dependencias:

- Requerimiento funcional: registro e inicio de sesión
- Casos de uso asociados: *Jugador – Registrarse, Jugador – Iniciar sesión*

Criterios de aceptación:

- El jugador puede crear una cuenta con nombre, correo y contraseña.
- Si los datos son correctos, se inicia sesión y se accede al panel de usuario.
- Si los datos son incorrectos, el sistema muestra un mensaje de error claro.

Tabla 6.1 – Trazabilidad: Requerimiento – User Story – Caso de uso

Requerimiento funcional	User Story	Caso de uso asociado
Crear torneos mediante un proceso guiado	US-01	Manager – Crear torneo
Generar automáticamente el fixture del torneo	US-02	Manager – Generar fixture
Publicar el torneo mediante un enlace accesible	US-03	Manager – Publicar torneo / Invitado – Visualizar torneo
Crear partidos rápidos para encuentros informales	US-04	Jugador – Crear partido rápido
Registrar jugadores e iniciar sesión	US-05	Jugador – Registrarse / Iniciar sesión

7. Recursos Disponibles

Optamos por una dinámica de trabajo colaborativa en la que **cada integrante asumió las tareas que mejor se alineaban con sus intereses y fortalezas**, pero siempre manteniendo un intercambio constante y tomando las decisiones de manera conjunta.

Aunque algunos roles tuvieron un peso más marcado en determinadas etapas —como diseño, análisis, modelado o aspectos técnicos—, **todo el grupo participó en las discusiones clave**, revisó materiales, propuso mejoras y acompañó la evolución del proyecto. Esta forma de trabajo nos permitió avanzar de manera equilibrada y asegurar coherencia entre las distintas partes del sistema.

7.1 Recursos humanos

Recursos utilizados en este trabajo

El equipo estuvo conformado por cuatro integrantes, cada uno con un rol predominante dentro del proyecto:

- **María Victoria Rivoldini** – *Producto y Diseño*
Lideró la visión del sistema, la definición del producto y el diseño del prototipo. Coordinó la experiencia de usuario, ordenó los flujos, revisó la coherencia funcional y llevó adelante gran parte de la redacción del informe.
- **Christian Diextra Castillo** – *Desarrollo Back-end y Modelado*
Contribuyó a la creación de los diagramas UML, la interpretación técnica del sistema, las User Stories y la revisión general del modelo funcional.
- **Maximiliano Nadalin** – *Desarrollo Front-end / Lógica del Sistema*
Enfocado en cómo se traducen las necesidades del producto en funcionalidades técnicas. Participó en el análisis funcional, los diagramas y la validación de casos de uso.
- **Rafael María Airoldi** – *Infraestructura*
Consideró los aspectos relacionados con entornos de ejecución, servidores y despliegue. También aportó en el análisis funcional y las discusiones sobre arquitectura.

Todos los integrantes participaron en las tareas de análisis y en la toma de decisiones sobre el alcance y la estructura del sistema.

7.2 Recursos materiales y tecnológicos

Herramientas utilizadas en el trabajo práctico

Durante el desarrollo de este proyecto usamos herramientas accesibles que permitieron diseñar, modelar y documentar la solución:

- **Figma** – prototipo, mockups e interfaz
- **draw.io** – diagramas UML
- **Microsoft Word** – documentación
- **OBS Studio** – grabación del video de presentación

Recursos tecnológicos necesarios para un desarrollo real

Para implementar FixtureAR como un sistema operativo se requerirían recursos adicionales, entre ellos:

- **Servidor de aplicación** (Node.js, Python, Java u otro stack)
- **Base de datos** (MySQL, PostgreSQL o MongoDB)
- **Hosting o infraestructura en la nube** (AWS, Azure, GCP)
- **Repositorios de control de versiones** (GitHub, GitLab o Bitbucket)
- **Pipeline de CI/CD** para despliegue continuo
- **Entornos de desarrollo locales** para los miembros del equipo
- **Servicios de autenticación** (OAuth, JWT, Firebase Auth, etc.)
- **Servicios de almacenamiento** para imágenes y adjuntos (S3, Cloud Storage)

Estos recursos permitirían pasar del prototipo a un sistema funcional, escalable y mantenable, alineado con la arquitectura planteada en el punto 6.

8. Plan de Trabajo

El plan de trabajo de FixtureAR organiza las actividades del proyecto a lo largo del tiempo, definiendo fases, tareas, responsables y dependencias. La idea es contar con una hoja de ruta clara que nos permita avanzar de manera ordenada desde el relevamiento inicial hasta la entrega final del sistema y la presentación.

Las tareas se agrupan en cinco grandes fases: **Análisis, Diseño, Desarrollo, Pruebas y Entrega**. Cada fase tiene un conjunto de actividades identificadas con un código (A1, D1, DEV1, etc.), una persona responsable principal y una duración estimada. Este esquema sirve como referencia para coordinar el trabajo en equipo y para visualizar cómo se encadenan las actividades en el tiempo.

Fase	Código	Tarea	Responsable	Inicio	Fin	Duración (días)	Depende de
Análisis	A1	Relevamiento de requerimientos y objetivos	Christian, Rafael	3/1/2026	3/7/2026	7	—
	A2	Identificación de actores y casos de uso	Christian	3/8/2026	3/14/2026	7	A1
	A3	Redacción de historias de usuario y documentación	Rafael	3/15/2026	3/21/2026	7	A2
Diseño	D1	Elaboración de diagramas de casos de uso y clases	Maximiliano	3/22/2026	4/4/2026	14	A2
	D2	Creación de diagramas de secuencia y componentes	Maximiliano, Victoria	4/5/2026	4/11/2026	7	D1
	D3	Diseño del prototipo visual (Figma)	Victoria	4/12/2026	4/18/2026	7	D1
Desarrollo	DEV1	Implementación del backend (generador de fixture, controlador)	Christian	4/22/2026	5/5/2026	14	D2
	DEV2	Desarrollo del frontend (login, vistas públicas)	Rafael	5/6/2026	5/19/2026	14	D3
	DEV3	Conexión con la base de datos (MySQL / HeidiSQL)	Christian	5/20/2026	5/26/2026	7	DEV1, D3
Pruebas	P1	Testeo funcional y de integración	Victoria, Maximiliano	5/27/2026	6/2/2026	7	DEV1, DEV3
	P2	Validación final del sistema y corrección de errores	Todo el grupo	6/3/2026	6/9/2026	7	P1
Entrega	E1	Elaboración del informe final y presentación	Todo el grupo	6/10/2026	6/16/2026	7	P2

8.1 Cronograma general (diagrama de Gantt)

A partir de esta planificación construimos un cronograma tipo diagrama de Gantt, donde se distribuyen las tareas a lo largo de las semanas del proyecto. En él se refleja el encadenamiento entre análisis, diseño, desarrollo, pruebas y entrega, así como los solapamientos necesarios para llegar a tiempo con la entrega final del proyecto.

Tarea	Semanas											
	1	2	3	4	5	6	7	8	9	10	11	12
A1												
A2												
A3												
D1												
D2												
D3												
DEV1												
DEV2												
DEV3												
P1												
P2												
E1												

Figura 8.1 – Diagrama de Gantt del proyecto FixtureAR.

9. Gestión de Riesgos

La gestión de riesgos fue importante tanto para planificar la implementación futura de FixtureAR como para manejar las dificultades reales que surgieron durante el trabajo práctico. A lo largo del proyecto identificamos riesgos técnicos, organizativos y de equipo, algunos propios de un sistema en etapa conceptual y otros asociados a la dinámica real del grupo. El objetivo es dejar documentados los riesgos más relevantes y definir estrategias para reducir su impacto o responder a tiempo si llegaran a ocurrir.

Nº	Riesgo	Severidad	Mitigación
R1	Falta de tiempo para completar todas las entregas (riesgo real vivido por el grupo)	Alta	Definir prioridades claras, concentrarse en el MVP, dividir tareas según fortalezas y mantener reuniones cortas de seguimiento.
R2	Ausencia de roles necesarios para el desarrollo real (QA, DevOps, programadores adicionales)	Media	Incorporar roles rotativos dentro del grupo para cubrir vacíos; definir límites claros del alcance académico para no depender de un perfil inexistente.
R3	Retrasos o bloqueos por falta de definiciones rápidas (decisiones de análisis, prototipo o UML)	Media	Tomar decisiones por consenso rápido; documentar las alternativas descartadas; avanzar con la opción más viable para mantener el ritmo.
R4	Fallas del servidor o indisponibilidad en una futura implementación	Alta	Usar infraestructura en la nube con monitoreo básico; backups automáticos; separación de servicios críticos.
R5	Pérdida o corrupción de datos de torneos, equipos o partidos	Alta	Implementar validaciones de entrada, transacciones para operaciones sensibles y respaldos periódicos.
R6	Vulnerabilidades de seguridad en autenticación o manejo de datos	Media	Aplicar cifrado de contraseñas, sanitización de entradas, y buenas prácticas de autenticación en caso de implementarse.
R7	Sobrecarga del sistema ante un uso mayor al esperado	Media	Diseñar con arquitectura escalable; separar frontend/backend; permitir migrar fácilmente a infraestructura más robusta.

10. Diseño del Sistema

10.1 Arquitectura Cliente-Servidor

La arquitectura propuesta para FixtureAR sigue un modelo **cliente-servidor organizado en tres capas**: frontend, backend y base de datos. Esta estructura está representada en el diagrama de componentes y permite visualizar cómo interactúan los distintos módulos del sistema durante el funcionamiento normal.

En el **Frontend**, que funciona en el navegador del usuario, se encuentran las vistas principales que permiten la interacción directa con FixtureAR. Entre ellas se incluyen:

- **Login**, donde los usuarios que lo requieran pueden autenticarse.
- **Vista de Tablas y Resultados**, que muestra datos procesados por el sistema (por ejemplo, clasificación o resultados).
- **Vista de Fixture**, que permite al usuario consultar los partidos y el cronograma generado.
- Una interfaz general denominada **Aplicación Web**, que agrupa las pantallas mencionadas y actúa como punto de entrada para el usuario (por ejemplo, el Organizador representado en la figura).

Desde estas vistas se envían solicitudes de datos hacia la capa de backend.

El **Backend** está compuesto por tres módulos principales:

- **Generador de Fixture**, responsable de aplicar la lógica que construye los cruces entre equipos.
- **Controlador**, que actúa como intermediario entre el frontend, la lógica del sistema y la base de datos.
- **Seguridad**, módulo encargado de validar accesos, revisar credenciales y garantizar que solo usuarios autorizados ejecuten operaciones específicas.

Estos componentes procesan las solicitudes provenientes del frontend y, cuando es necesario, consultan la información almacenada en la base de datos.

La capa de **Base de Datos** incluye:

- La **interfaz de Base de Datos**,
- Y un conjunto de entidades persistentes: **Usuarios, Torneos, Equipos, Partidos y Resultados**, que conforman el núcleo del modelo de datos del sistema.

La comunicación entre las capas se realiza mediante flujos bien definidos:

- El frontend **envía datos** al backend para procesar solicitudes.
- El backend realiza **consultas a la base de datos** y devuelve los resultados al frontend.
- Esta interacción permite que las vistas se actualicen y muestren la información más reciente.

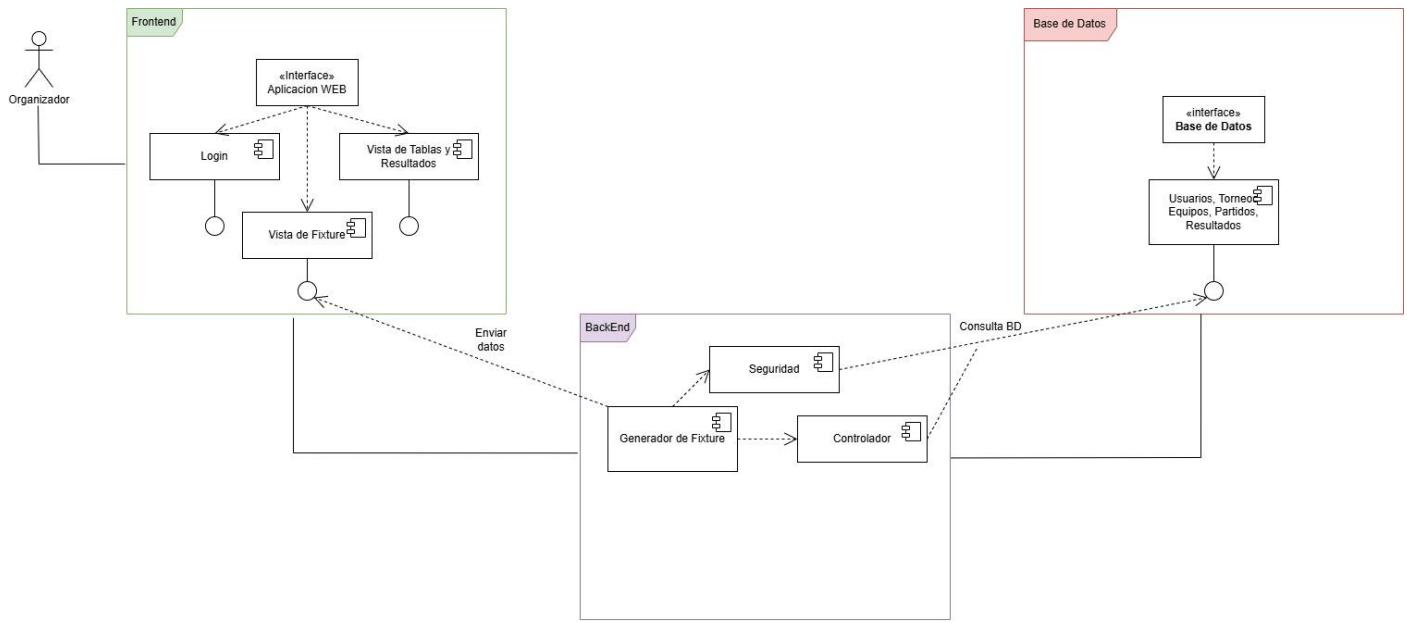


Figura 10.1 – Diagrama de componentes de FixtureAR, que muestra la interacción entre frontend, backend y base de datos.

Justificación de la arquitectura elegida

Elegimos esta arquitectura por varias razones:

- **Adecuación al contexto educativo:** al ser una aplicación web, basta con contar con un navegador y conexión a internet; no hace falta instalar software en cada dispositivo del colegio o club.
- **Centralización de la información:** tener un servidor de aplicación y una base de datos central permite mantener un único “lugar de verdad” para los torneos, evitando versiones dispersas en planillas o archivos compartidos.
- **Escalabilidad y evolución:** separar cliente, servidor y base de datos facilita que el sistema pueda crecer en funcionalidad (por ejemplo, estadísticas, notificaciones o IA en el futuro) sin necesidad de reescribir todo desde cero.
- **Mantenibilidad:** la división en capas (presentación en el cliente, lógica en el servidor y persistencia en la base de datos) ayuda a aislar cambios y a mantener el código más ordenado.
- **Compatibilidad:** este modelo es estándar en aplicaciones web modernas, lo que lo hace compatible con distintas tecnologías de implementación y con prácticas conocidas por equipos de desarrollo.

10.2 Diagramas UML

A) Diagrama de Clases

Clases identificadas y su justificación

En base a los casos de uso y diagramas de actividades, se identificaron las siguientes clases principales para representar la lógica del sistema:

- **Usuario:** clase general que representa a cualquier persona que interactúe con la plataforma.
- **Invitado, Jugador y Manager:** especializaciones de Usuario, con diferentes permisos y funcionalidades según su perfil.
- **Torneo:** entidad central creada por un Manager, que agrupa equipos y partidos.
- **Equipo:** conjunto de jugadores que compiten en un torneo o partido.
- **Partido:** evento deportivo, puede ser parte de un torneo o un partido rápido generado por un jugador.
- **Estadística:** clase que almacena datos de rendimiento de un jugador en partidos o torneos.
- **Notificación:** clase que representa mensajes o recordatorios enviados por el sistema a los usuarios.

Relaciones entre las clases

- Usuario se especializa en Jugador, Manager e Invitado.
- Jugador puede participar en varios Partidos, y generar Estadísticas.
- Manager puede crear uno o varios Torneos, y gestionar los Equipos que los componen.
- Equipo está compuesto por varios Jugadores, y participa en Partidos.
- Partido vincula a dos o más Equipos, y puede formar parte de un Torneo.
- Sistema puede enviar Notificaciones a cualquier tipo de Usuario.

Cobertura del sistema

Este modelo cubre la lógica principal de la plataforma *FixtureAR*, incluyendo los distintos perfiles de usuario, la creación y gestión de torneos, la participación en partidos, y el registro de estadísticas deportivas. Las relaciones entre las clases permiten una representación coherente de los flujos funcionales y establecen una base sólida para el desarrollo del sistema.

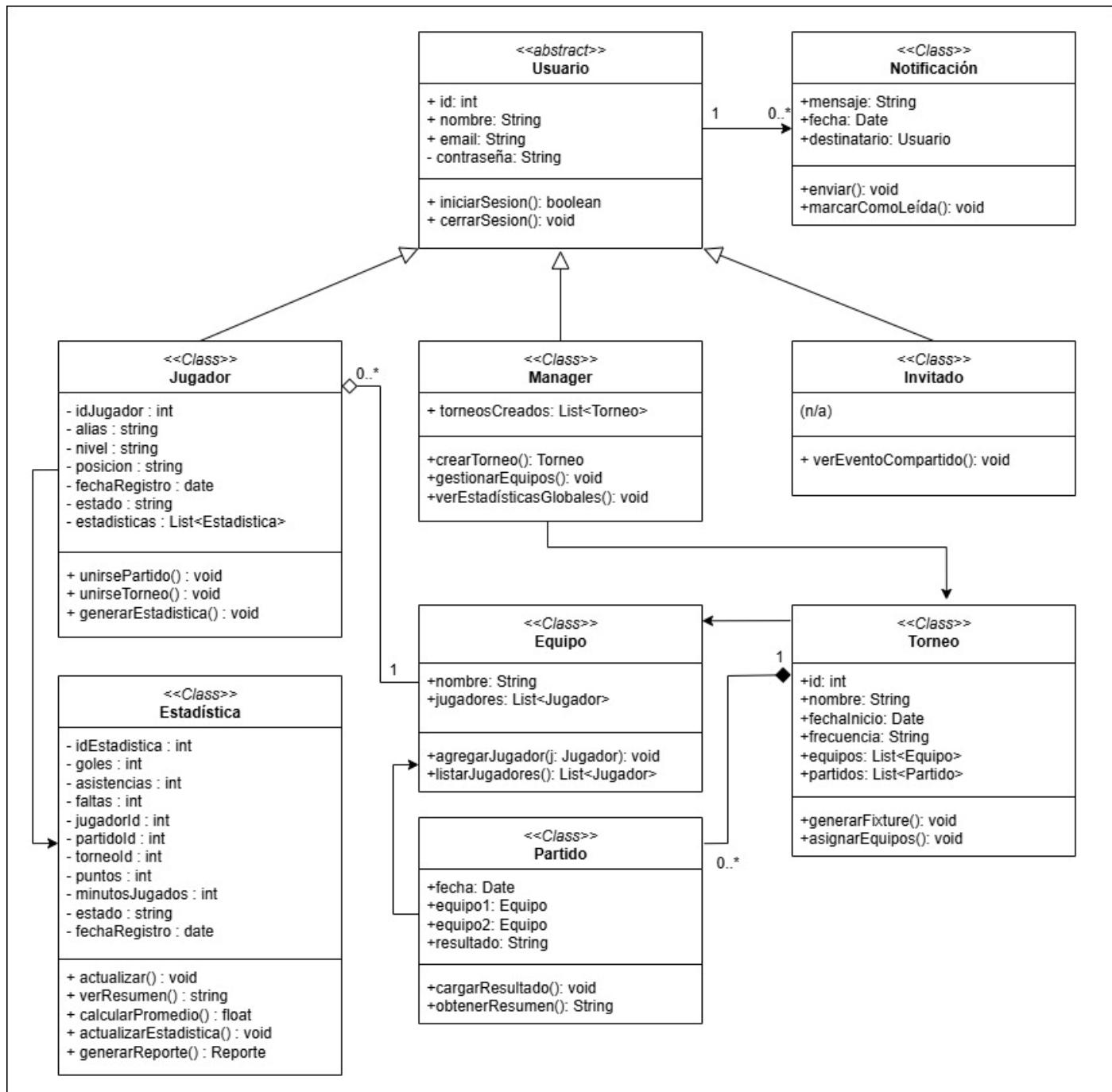


Figura 10.2 – Diagrama de clases principal del sistema *FixtureAR*

B) Diagramas de Secuencia

Para complementar la vista estática del sistema que ofrece el diagrama de clases, elaboramos tres **diagramas de secuencia** asociados a casos de uso clave: creación de torneo, registro de jugador y visualización de torneo público. Estos diagramas muestran cómo se comunican entre sí los distintos objetos durante la ejecución de cada escenario, y permiten verificar que el comportamiento dinámico del sistema sea coherente con el modelo funcional definido en los casos de uso.

En todos los casos se distinguen claramente el **actor** (Manager, Jugador o Invitado), la **interfaz de usuario** correspondiente, el **controlador** que concentra la lógica de aplicación y los objetos o componentes que intervienen en la operación (entidades de dominio y bases de datos específicas). De esta manera, los diagramas nos ayudaron a revisar el rol de cada elemento y a asegurar que no haya responsabilidades duplicadas o pasos innecesarios en los flujos principales.

Creación de torneo – Manager

En el primer diagrama se modela el caso de uso “**Creación de Torneo**” desde la perspectiva del Manager. El actor inicia el proceso desde la interfaz **UI_CrearTorneo**, que recopila los datos del formulario y los envía al **ControladorTorneo**. Este controlador crea una instancia de **Torneo**, que a su vez solicita al objeto **Fixture** que genere la lista de partidos en función de los parámetros ingresados. Una vez generado el fixture, el Torneo devuelve la información al controlador, que se encarga de guardar los datos en la **BaseDTorneo**. Finalmente, la base confirma el guardado y el controlador notifica a la interfaz que el torneo fue creado con éxito, permitiendo mostrar el fixture al Manager.

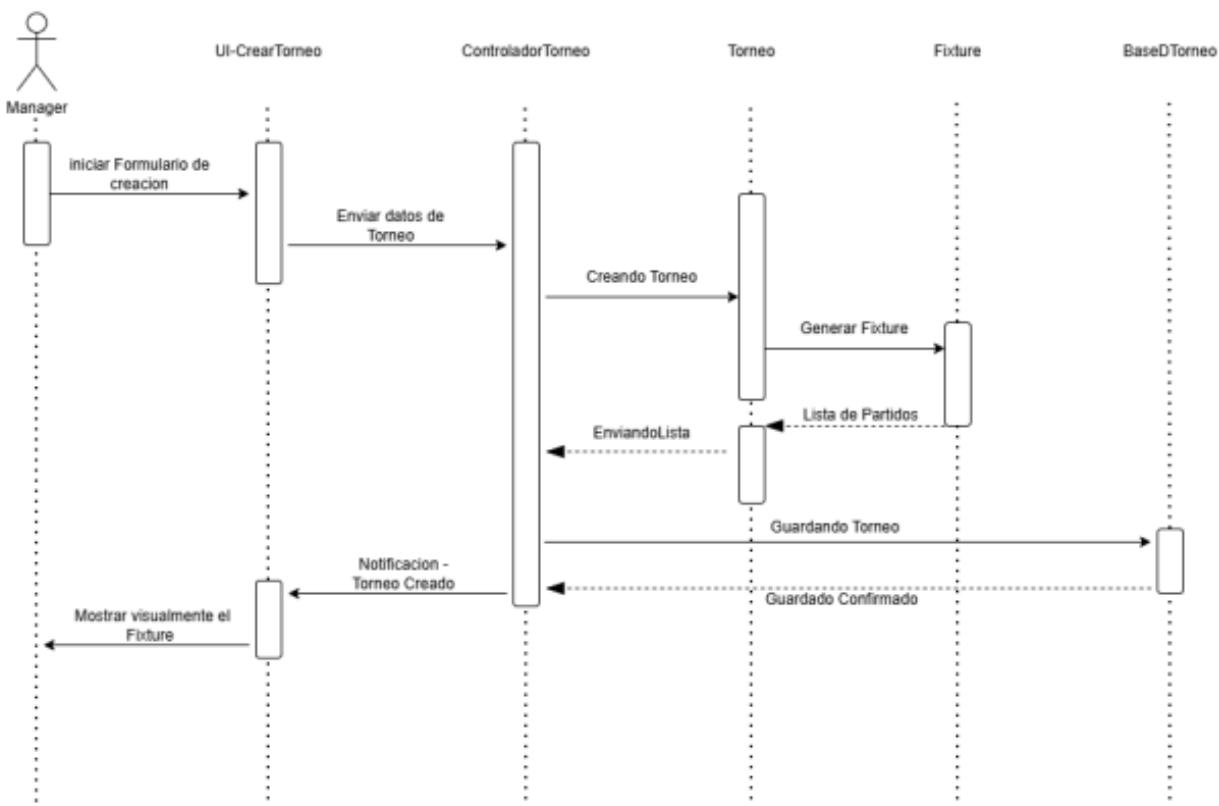


Figura 10.3 – Diagrama de secuencia para el caso de uso “Creación de Torneo” (Manager).

Registro de jugador

El segundo diagrama representa el caso de uso “**Registrarse**”, asociado al actor **Jugador**. El flujo se inicia cuando el Jugador abre la pantalla de registro en **UI_Registrar** e ingresa sus datos. La interfaz envía esa información al **Controlador_Usuario**, que se encarga de validar los datos recibidos (por ejemplo, formato del correo o contraseñas). Una vez validados, el controlador procede a guardar el nuevo usuario en la **BD_Usuario**. La base confirma que el registro se almacenó correctamente y el controlador envía una respuesta a la interfaz, que muestra al jugador un mensaje de registro exitoso. Este diagrama permitió revisar que todo el proceso de alta de usuario quede encapsulado en el controlador y en la base correspondiente.

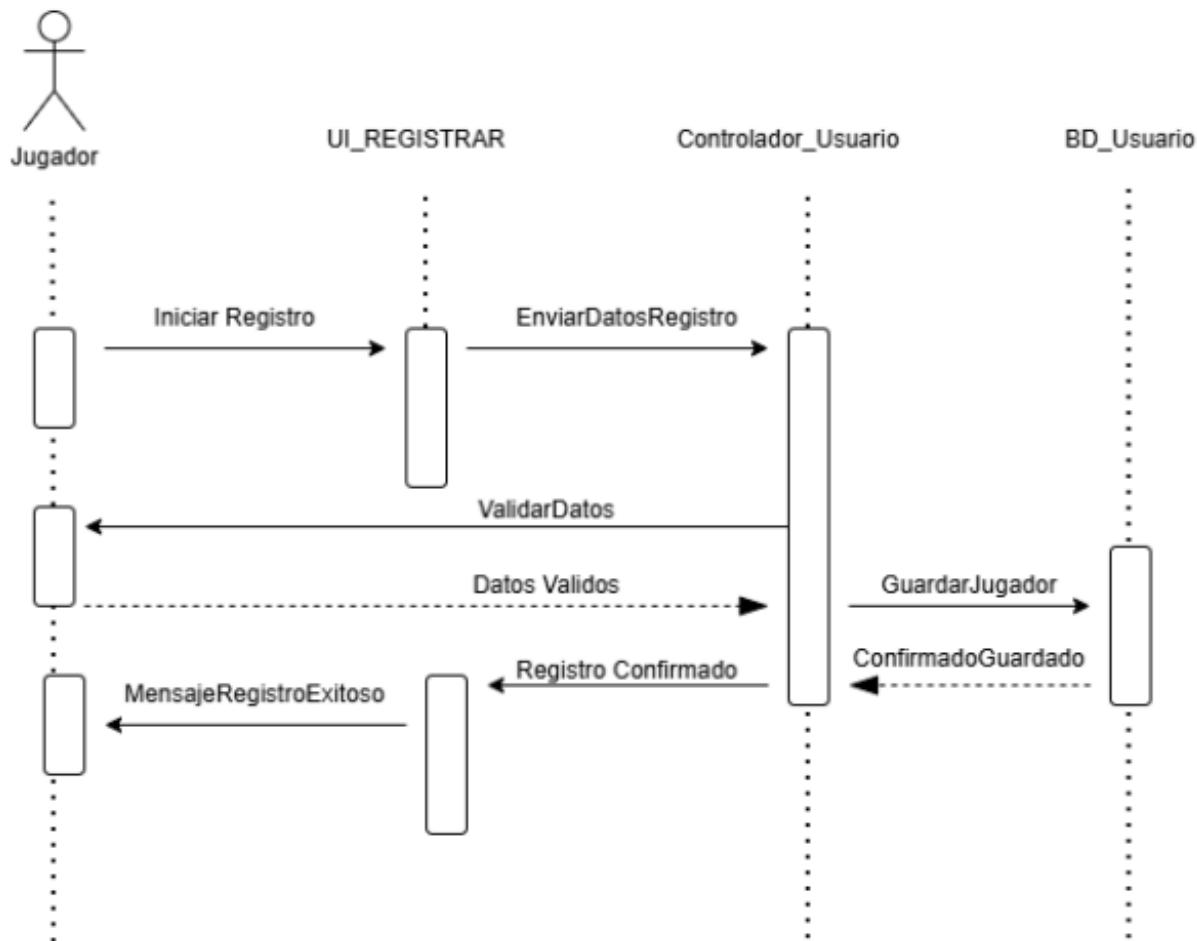


Figura 10.4 – Diagrama de secuencia para el caso de uso “Registrarse” (Jugador).

Visualización de torneo público

El tercer diagrama aborda el caso de uso “**Visualizar Torneo Público**”, desde el punto de vista del actor **Invitado**. En este escenario, el Invitado accede a la pantalla **UI_PublicoTorneo** y selecciona el torneo que quiere consultar. La interfaz solicita los datos al **ControladorTorneo**, que a su vez envía una consulta a la **BD_Torneos** para obtener la información necesaria (equipos, partidos, horarios, etc.). La base devuelve los datos al controlador y este los reenvía a la interfaz, que finalmente muestra el contenido al Invitado. Este flujo refuerza la idea de que incluso para usuarios no autenticados la información siempre se obtiene de la misma fuente central, manteniendo la coherencia con la arquitectura propuesta.

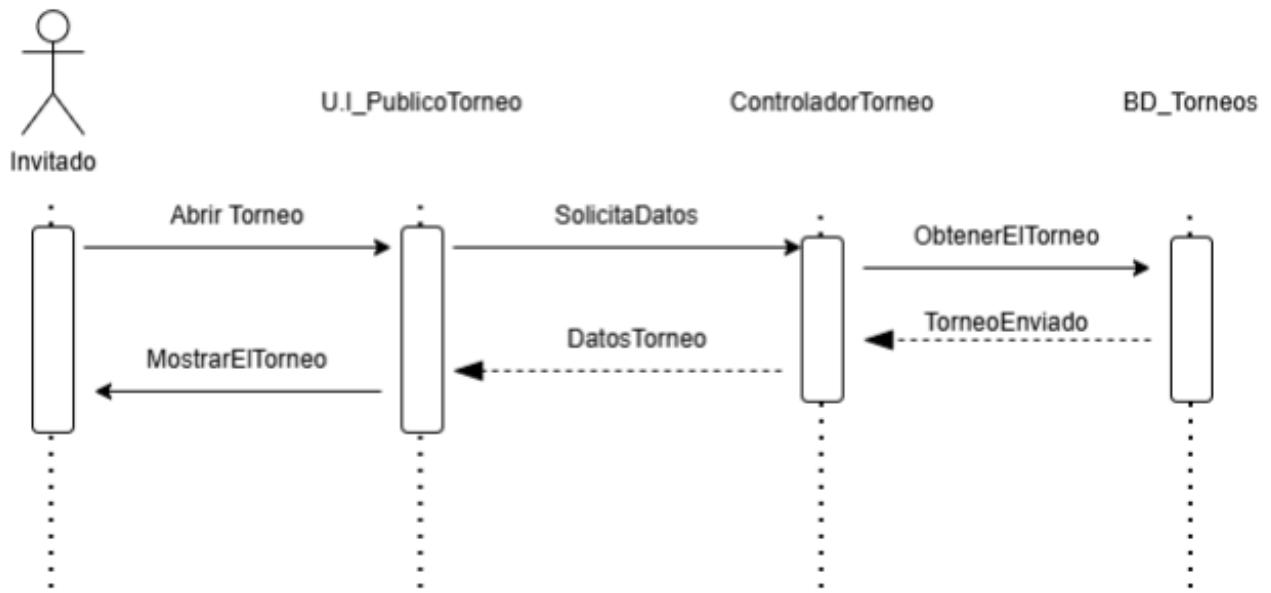


Figura 10.5 – Diagrama de secuencia para el caso de uso “Visualizar Torneo Público” (Invitado).

C) Diagramas de Actividad

Los siguientes diagramas de actividades representan el flujo principal de acciones que cada tipo de usuario puede realizar dentro de la aplicación **FixtureAR**. Cada rol accede a funcionalidades específicas según sus permisos y objetivos dentro del sistema.

Rol: Manager (Organizador)

El manager es responsable de crear y gestionar torneos. Su flujo de actividad incluye el ingreso al sistema, la carga de datos del torneo, la generación automática del fixture y la posterior publicación mediante un enlace. También puede crear partidos dentro del torneo o visualizar los ya generados.

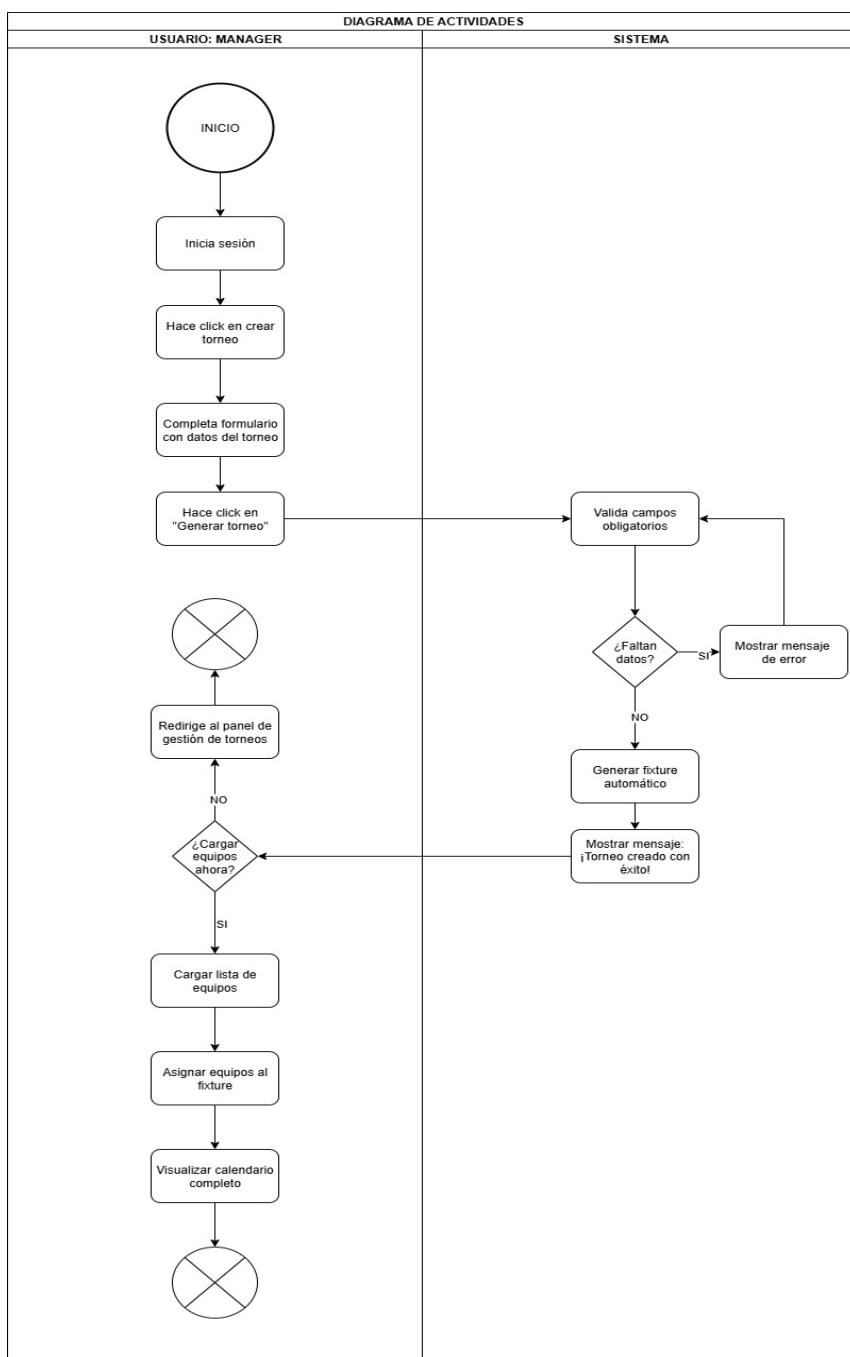


Figura 10.6 – Diagrama de actividad del rol Manager: creación y gestión de torneos

Rol: Jugador

El jugador accede a la aplicación para crear partidos informales, ver próximos encuentros o consultar información relevante. Su flujo se enfoca en la creación rápida de un partido, la selección de datos básicos como fecha, hora, sede y jugadores convocados, y la obtención de un link para compartir con sus compañeros.

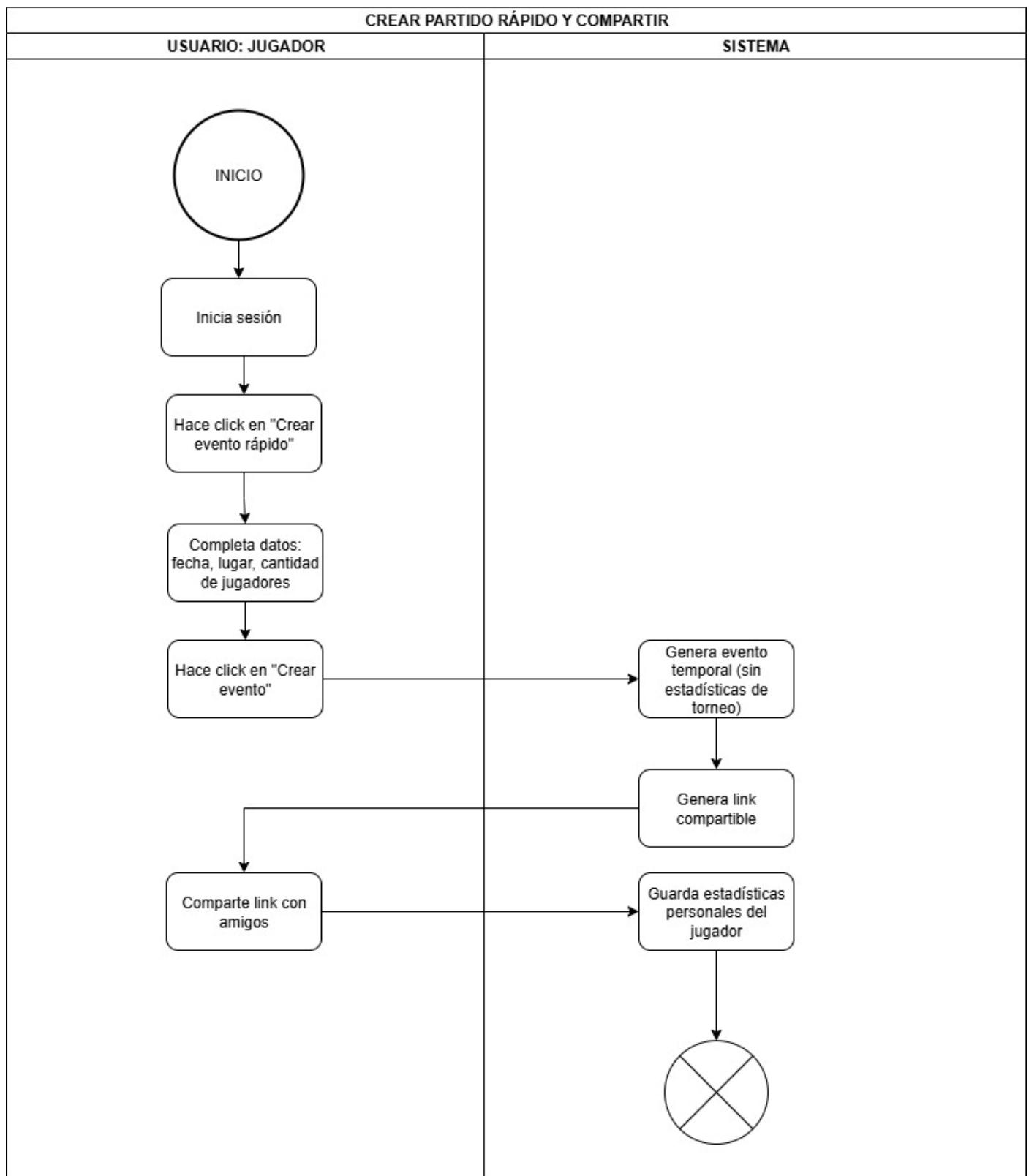


Figura 10.7 – Diagrama de actividad del rol Jugador

Rol: Invitado

El invitado no necesita iniciar sesión. Accede directamente desde un enlace recibido para consultar información puntual sobre un partido o torneo. Su flujo se limita a la visualización: puede ver los detalles del encuentro, la ubicación y la fecha, pero no interactuar con los datos.

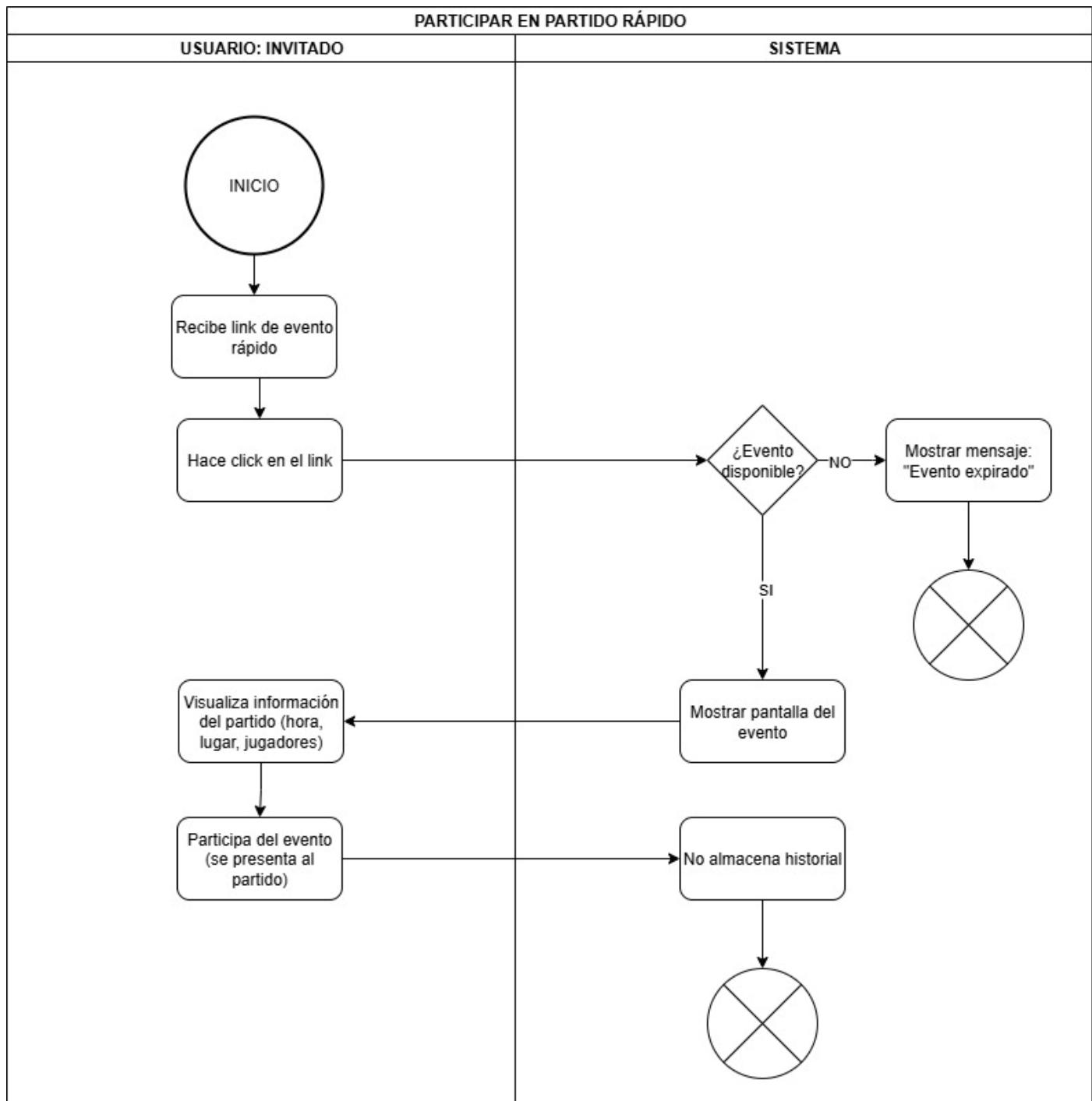


Figura 10.8 – Diagrama de actividad del rol Invitado

10.3 Interfaces de usuario (UI)

El diseño de las interfaces se desarrolló siguiendo criterios de **usabilidad, consistencia visual y jerarquía clara**, priorizando que cada usuario pueda identificar rápidamente las acciones principales dentro de la plataforma. Se empleó una estructura fija con **menú lateral persistente** y un **panel central de contenido**, lo que facilita la navegación y reduce la carga cognitiva. Las pantallas mantienen una disposición limpia, con elementos espaciados y tipografías legibles que permiten interpretar fácilmente el estado de cada torneo, partido o flujo de interacción.

La **paleta de colores** se compone de tonos verdes y azules suaves, asociados con claridad, confianza y deporte. Estos colores se utilizan tanto para resaltar acciones principales (como botones o estados activos) como para diferenciar secciones del sistema sin generar saturación visual. Los rectángulos de datos y las tarjetas mantienen un fondo claro para reforzar la lectura y la coherencia entre pantallas.



El **logotipo** de FixtureAR utiliza una síntesis visual simple y directa para representar la esencia del sistema. La figura combina nodos conectados por líneas rectas, evocando la estructura de un **fixture o cuadro de torneo**, donde cada punto simboliza un equipo y cada conexión representa un cruce o instancia de competencia. Su diseño minimalista permite una rápida identificación y mantiene coherencia con la estética general de la plataforma, priorizando claridad y facilidad de reconocimiento en distintos tamaños y soportes.



A continuación, se incluye un enlace al prototipo en **Figma**, donde se pueden visualizar más detalles sobre el **flujo de navegación, transiciones, efectos interactivos y comportamiento de los componentes** dentro del sistema:

<https://www.figma.com/proto/B2KwxYmG5OpZAoomVFhJyr/FixtureAR?node-id=0-1&t=NGEP3y1lyL7Leepw-1>

Prototipo



Figura 10.9 – Pantalla inicial de FixtureAR (Home)

Muestra las acciones principales disponibles para los usuarios: crear un torneo, iniciar sesión, registrarse o acceder a la demo. Esta pantalla es el punto de entrada al sistema y establece la estructura visual general del prototipo.

Se priorizó una jerarquía visual clara, con acciones primarias destacadas y un diseño limpio que permite identificar con rapidez cómo comenzar, especialmente útil para usuarios con baja alfabetización digital.



Figura 10.10 – Pantalla inicial de FixtureAR (Home)

El scroll sostiene la misma estructura visual y mantiene la coherencia en spacing, colores y tipografías, reforzando la continuidad del flujo según principios de Gestalt.



Figura 10.11 – Pantalla de login del usuario

Corresponde al acceso para Jugadores o Managers que deban autenticarse. Es el punto de ingreso para las funcionalidades personalizadas según cada perfil.

El diseño busca minimizar la fricción: pocos campos, textos claros y patrones reconocibles que facilitan la autenticación sin distracciones.



Figura 10.12 – Pantalla de registro del jugador

Formulario donde el usuario crea una cuenta ingresando sus datos básicos. Desde aquí obtiene acceso a las funcionalidades asociadas a su perfil.

Se aplicó el principio de ‘reconocer antes que recordar’: cada campo está etiquetado de forma explícita y el flujo guía al usuario paso a paso, reduciendo errores por carga incompleta.

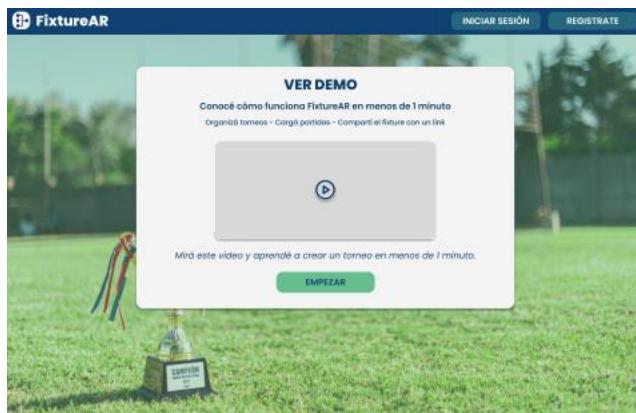


Figura 10.13 – Pantalla una vez que se accede a la sección “Ver demo”

Esta vista funciona para observar el funcionamiento general sin requerir interacción compleja, facilitando la comprensión del sistema a primera vista.

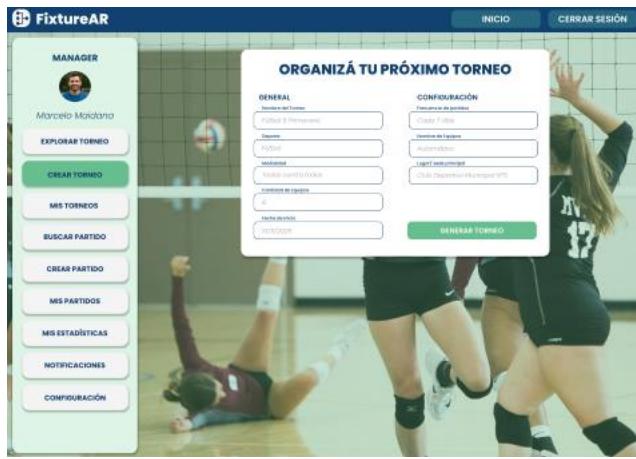


Figura 10.14 – Pantalla de creación de torneos

Los campos están agrupados según relevancia para facilitar la lectura. La distribución en bloques ayuda a reducir la carga cognitiva y permite completar el formulario de manera progresiva.

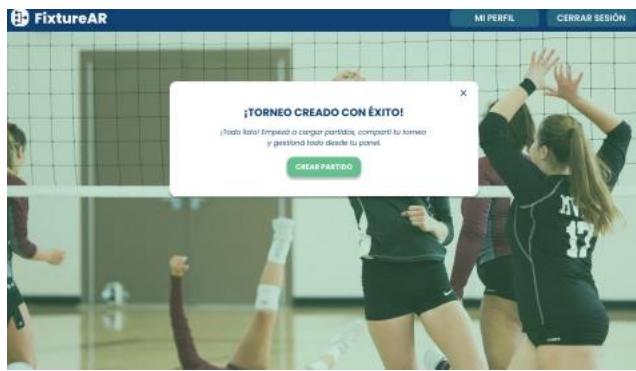


Figura 10.15 – Pantalla de confirmación de torneo creado exitosamente

El mensaje de feedback inmediato cumple con la heurística de visibilidad del estado del sistema, indicando claramente que la acción se completó y ofreciendo el enlace como siguiente paso natural.



Figura 10.16 – Pantalla de creación de partida rápida

El formulario está simplificado para favorecer acción rápida. Se priorizó el mínimo de campos necesarios y un diseño directo para que la creación del partido sea ágil.



Figura 10.17 – Confirmación de partida rápida creada exitosamente

Al igual que en la creación de torneo, se ofrece feedback inmediato y un enlace destacado. Esto permite cerrar el flujo de manera intuitiva y facilita compartir la información sin pasos extra.



Figura 10.18 – Pantalla de datos de partida para invitados

Se diseñó con un enfoque de lectura rápida: datos esenciales visibles, sin opciones secundarias. Pensado para usuarios esporádicos que solo necesitan consultar y no interactuar.

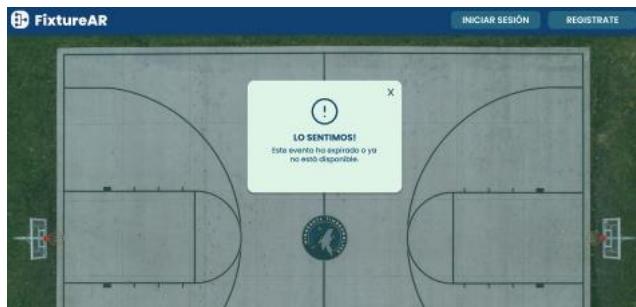


Figura 10.19 – Pantalla de error

Vista que se muestra cuando el usuario accede a un partido cuyo enlace ha expirado o ya no está disponible.

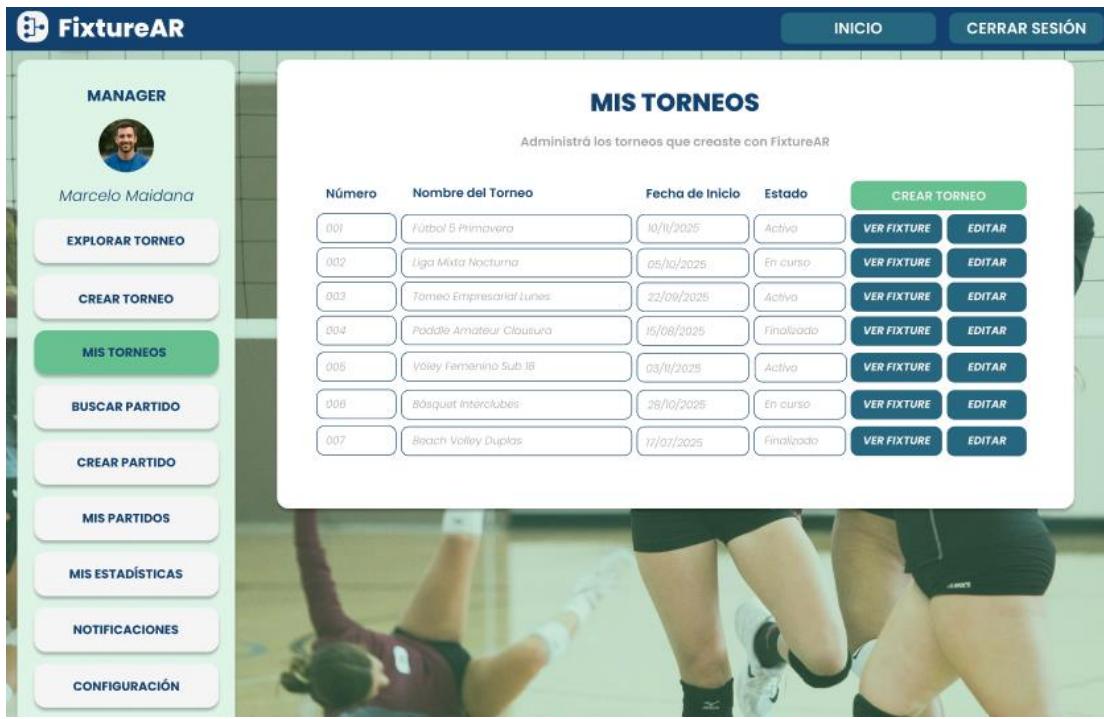


Figura 10.20 – Pantalla “Mis Torneos”

Vista del panel del Manager donde se listan todos los torneos creados por el usuario. La pantalla presenta una tabla con número de torneo, nombre, fecha de inicio, estado y acciones disponibles, permitiendo acceder rápidamente al fixture o editar la configuración de cada torneo.



Figura 10.21 – Pantalla de fixture

Pantalla de detalle del torneo seleccionado, donde se visualiza la estructura del fixture, permitiendo al usuario consultar la organización completa del torneo y navegar entre sus distintas instancias.

10.4 Interfaces con otros sistemas

En la versión actual del proyecto no se implementan integraciones concretas con sistemas externos (APIs o servicios web). Sin embargo, la arquitectura definida contempla la posibilidad de incorporar, en etapas futuras, servicios de terceros como:

- envío de correos o notificaciones,
- autenticación externa,
- o pasarelas de pago para inscripciones.

Estas integraciones se ubicarían como servicios conectados al backend mediante APIs, sin modificar la lógica principal del sistema.

11. Conclusiones

A lo largo de este trabajo fuimos pasando por distintas etapas: entender el problema real que queríamos abordar, traducirlo en requerimientos, pensar a los usuarios, diseñar la experiencia, modelar el sistema y, finalmente, ordenar todo en un documento único. En ese recorrido no solo definimos cómo podría funcionar FixtureAR, sino que también tuvimos que organizarnos como equipo, repartir tareas y aprender a combinar miradas distintas sobre un mismo producto.

Como grupo sentimos que el principal aprendizaje estuvo en **conectar la teoría con algo concreto**. Conceptos como casos de uso, diagramas UML, MVP o gestión de riesgos dejaron de ser únicamente contenidos de la materia para convertirse en herramientas que usamos efectivamente para tomar decisiones: qué incluir en la primera versión, qué dejar para futuras iteraciones y cómo justificar cada elección frente a la cátedra. También fue importante aprender a documentar todo ese proceso para que cualquiera pueda entender qué propone el sistema y cómo está pensado.

En cuanto a **fortalezas** del sistema, creemos que FixtureAR aporta claridad y organización en un contexto donde hoy predominan las planillas sueltas y los mensajes dispersos. La idea de un fixture centralizado, accesible desde un enlace público y pensado para distintos perfiles (organizador, jugador, invitado) responde directamente al problema inicial. Además, el diseño de la interfaz y el enfoque en la simplicidad ayudan a que el sistema sea viable en ámbitos educativos donde el tiempo y los recursos técnicos son limitados.

Al mismo tiempo, identificamos **debilidades y desafíos pendientes**. La solución actual se centra en los flujos básicos de creación y consulta de torneos, dejando para el futuro funcionalidades más avanzadas como estadísticas detalladas, administración de comunidades o notificaciones automáticas. También reconocemos que la implementación real requeriría un trabajo más profundo en temas de seguridad, rendimiento y escalabilidad, así como la participación de perfiles técnicos adicionales (desarrolladores, QA, infraestructura).

Mirando hacia adelante, vemos varias **líneas de mejora** posibles. Una de ellas es avanzar hacia una implementación funcional del sistema, priorizando primero el backend de generación de fixture y las vistas públicas. Otra línea interesante es explorar el uso de **inteligencia artificial** para tareas como la generación de fixtures optimizados según restricciones de horarios o sedes, o la detección de conflictos en la programación de partidos. Estas ideas quedaron registradas como posibles evoluciones, conscientes de que exceden el alcance de esta primera entrega, pero abren un camino claro para seguir iterando sobre FixtureAR.

Posibles incorporaciones futuras: análisis preliminar sobre el uso de IA

Durante el desarrollo del proyecto surgió la posibilidad de evaluar el uso de técnicas de inteligencia artificial como una mejora futura del sistema. Si bien esta funcionalidad no forma parte del MVP ni de los requerimientos aprobados, consideramos relevante dejar documentadas algunas líneas de exploración que podrían aportar valor en versiones posteriores.

En particular, la IA podría utilizarse para:

- **Optimizar la generación de fixtures** en torneos más complejos, evaluando múltiples combinaciones para minimizar cruces repetidos o desigualdades en los horarios.
- **Detectar inconsistencias o errores** en la carga de datos del organizador antes de generar el fixture, sugiriendo correcciones automáticas.
- **Proponer horarios alternativos** ante reprogramaciones, considerando disponibilidad y restricciones definidas por el usuario.
- **Analizar estadísticas de encuentros** (en versiones futuras que incluyan resultados) para generar resúmenes automáticos.

Estas propuestas representan líneas de evolución posibles, pero requieren un análisis técnico más profundo y herramientas adicionales que exceden el alcance de esta entrega. Sin embargo, dejamos asentado el interés en continuar evaluando su incorporación en una versión posterior del proyecto.