



**UNIVERSIDAD AUTÓNOMA DE ZACATECAS**



**FRANCISCO GARCÍA SALINAS**

**UNIDAD ACADÉMICA DE INGENIERÍA ELÉCTRICA**

**INGENIERÍA DE SOFTWARE**

**MATERIA: ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS**

**PROFESOR: CRISTIAN EDUARDO BOYAIN Y GOYTIA LUNA**

**ALUMNOS:**

**JOSÉ DE JESÚS MARTÍNEZ CERVANTES**

**JOSÉ DE JESÚS RUÍZ PASTRANA**

**JESÚS DANIEL GALLEGOS FLORES**

**PROYECTO: SISTEMA DE GESTIÓN DE TORNEOS DEPORTIVOS**

**FECHA: JUNIO 2025**

# Sistema de Gestión de Torneos Deportivos

---

## Documentación completa del sistema

### 1. Introducción General del Proyecto

En el ámbito deportivo local, la gestión de torneos suele realizarse de forma manual, lo cual genera problemas como errores en la inscripción de equipos, falta de control en los calendarios, duplicidad de resultados y dificultades para generar estadísticas confiables. Este proyecto tiene como objetivo desarrollar un Sistema de Gestión de Torneos Deportivos que permita organizar, controlar y supervisar torneos de manera digital, eficiente y accesible para todos los involucrados. El sistema está diseñado para ser utilizado en contextos como escuelas, clubes deportivos, ligas municipales o competencias comunitarias, donde los recursos son limitados y la necesidad de automatización es alta. A través de una plataforma intuitiva y basada en roles, se permitirá a los usuarios realizar acciones como registrar jugadores y equipos, programar partidos, registrar resultados, y consultar estadísticas y tablas de posiciones actualizadas en tiempo real. El desarrollo del sistema seguirá una metodología iterativa basada en el Proceso Unificado Ágil (UP Ágil), que prioriza la entrega temprana de valor, la validación continua de requerimientos y la mejora incremental del producto. Se aplicarán prácticas de análisis y diseño orientado a objetos (ADOO) para estructurar el sistema de forma escalable, reutilizable y mantenible, empleando diagramas UML, identificación de casos de uso y modelado de clases y objetos. Este documento contiene todas las fases del desarrollo del sistema, comenzando con la fase de Inception, en la que se presenta esta introducción, visión y objetivos generales, hasta llegar a los diagramas, prototipos, pruebas y conclusiones finales.

### 2. Fase de Inicio (Inception)

La fase de inicio es fundamental dentro del proceso unificado, ya que permite establecer una visión inicial del sistema, sus objetivos generales, los actores

involucrados, los riesgos del negocio, las restricciones principales y una primera delimitación del alcance del proyecto.

## 2.1 Descripción General

Este sistema busca reemplazar los métodos manuales y dispersos actualmente usados por los organizadores (como hojas de Excel o papel) por una solución centralizada y accesible. Permitirá la administración de torneos, equipos, jugadores, resultados, calendarios y tablas de posiciones, todo dentro de una interfaz amigable y segura.

## 2.2 Visión del Sistema

El Sistema de Gestión de Torneos Deportivos está diseñado para facilitar la administración de torneos en diferentes niveles (escolar, comunitario, municipal, etc.). Su propósito es eliminar procesos manuales y centralizar la información relacionada con:

Registro de usuarios con distintos roles (jugadores, organizadores, árbitros, administradores).

Creación y programación de torneos y partidos.

Inscripción de equipos y jugadores.

Registro de resultados.

Generación automática de calendarios y tabla de posiciones.

Visualización de estadísticas finales.

El sistema estará disponible como una aplicación de escritorio local (o adaptable a web en iteraciones futuras), fácil de usar y con validación en cada módulo.

## 2.3 Actores Principales

Actor	Descripción
Administrador	Administra usuarios, permisos y parámetros generales del sistema.
Organizador	Crea torneos, registra equipos y realiza calendarios.
Arbitro	Registra los resultados de los partidos.
Jugador	Consulta sus estadísticas y calendario.
Representante de equipo	Inscribe a su equipo y gestiona a sus jugadores asociados en algún torneo.

### 3. Metodología de Desarrollo: Unified Process (UP) Ágil

El desarrollo del Sistema de Gestión de Torneos Deportivos se realizará aplicando el enfoque del Proceso Unificado (Unified Process - UP) adaptado a un modelo ágil e iterativo, lo cual permite entregar valor de forma progresiva, aceptar el cambio como parte del proceso, y mejorar continuamente la calidad del sistema.

#### 3.1 ¿Qué es el Proceso Unificado?

El UP es una metodología de desarrollo de software centrada en el análisis y diseño orientado a objetos (ADOO), que divide el ciclo de vida del proyecto en fases bien definidas. Cada fase se compone de una o más iteraciones que entregan una parte funcional del sistema.

Este proceso ayuda a identificar y reducir riesgos tempranos, definir una arquitectura sólida desde el inicio, y construir el sistema paso a paso, priorizando los requisitos más críticos o de mayor valor.

#### 3.2 Fases del UP aplicadas al proyecto

Fase	Descripción
Inicio (Inception)	Se definió la visión general del sistema, actores y objetivos iniciales.
Elaboración (Elaboration)	Se modeló el dominio del problema, se identificaron los casos de uso más importantes y se definió la arquitectura base.
Construcción (Construction)	Se desarrollaron e integraron los módulos funcionales mediante iteraciones.
	Transición (Transition) Se realizaron pruebas finales, ajustes

	correctivos y validación por parte del usuario.
--	---

### 3.3 Características del enfoque ágil aplicado

Este proyecto no sigue un UP tradicional y rígido, sino un UP Ágil, el cual conserva la estructura, pero incorpora valores y principios ágiles. Algunas características clave del enfoque utilizado son:

Iteraciones cortas y funcionales: Cada entrega parcial contiene funcionalidades completas, probadas y utilizables.

Modelado ligero pero eficaz: Se utilizan los modelos UML necesarios para entender y comunicar el diseño, evitando exceso de documentación.

Participación continua del usuario: Las decisiones se validan continuamente con el usuario o evaluador.

Retroalimentación constante: Cada entrega genera observaciones que se incorporan en las siguientes iteraciones.

Priorización de requisitos: Se desarrollan primero las funcionalidades de mayor valor o riesgo.

Validación temprana: Se construye un MVP (Producto Mínimo Viable) desde las primeras iteraciones.

### 3.4 Justificación de la metodología

El UP Ágil es ideal para este proyecto por las siguientes razones:

Permite controlar el alcance y adaptar el sistema a cambios sin rehacer todo desde cero.

Favorece un desarrollo incremental y validado, perfecto para proyectos estudiantiles con tiempos limitados.

Ayuda a estructurar el trabajo sin imponer procesos excesivos ni generar documentación innecesaria.

Se adapta a equipos pequeños o desarrolladores individuales (como es el caso).

## 4. Requerimientos del Sistema

Los requerimientos del sistema se clasifican en dos grandes grupos:

Requerimientos funcionales (RF): describen lo que el sistema debe hacer.

Requerimientos no funcionales (RNF): especifican cómo debe comportarse el sistema en términos de calidad, rendimiento, seguridad, etc., siguiendo el modelo FURPS+.

### 4.1 Requerimientos Funcionales (RF)

ID	Descripción
RF01	El sistema debe permitir el registro de usuarios con distintos roles (jugador, organizador, árbitro, administrador).
RF02	El sistema debe permitir el inicio de sesión con autenticación.
RF03	El organizador debe poder crear torneos especificando nombre, tipo, fechas y número de equipos.
RF04	El representante de equipo debe poder inscribir su equipo a un torneo disponible.
RF05	El sistema debe generar automáticamente el calendario de partidos.
RF06	El árbitro debe registrar los resultados de cada partido.
RF07	El sistema debe calcular y mostrar la tabla de posiciones actualizada.
RF08	Los usuarios deben poder consultar estadísticas del torneo.
RF09	El administrador debe poder gestionar usuarios (activar, desactivar, editar roles).

## **4.2 Requerimientos No Funcionales (RNF) – Modelo FURPS+**

### **F – Funcionalidad:**

RNF01: El sistema debe validar todos los datos obligatorios antes de guardar información.

RNF02: Debe garantizar integridad en la tabla de posiciones y estadísticas.

### **U – Usabilidad:**

RNF03: La interfaz debe ser clara, simple y comprensible para usuarios sin conocimientos técnicos.

RNF04: La navegación debe seguir una estructura consistente en todas las pantallas.

### **R – Fiabilidad:**

RNF05: El sistema debe detectar y manejar errores durante el ingreso de datos.

RNF06: En caso de cierre inesperado, no debe perder información ya guardada.

### **P – Rendimiento:**

RNF07: Las operaciones críticas (guardar resultados, generar calendarios) deben ejecutarse en menos de 2 segundos.

RNF08: El sistema debe poder operar correctamente con al menos 100 equipos inscritos en un torneo.

### **S – Soportabilidad:**

RNF09: El sistema debe ser compatible con Windows 10 o superior.

RNF10: La base de datos debe permitir respaldos y restauración manual.

### **+ – Otros:**

RNF11: Se debe poder exportar la tabla de posiciones y estadísticas a formato CSV o PDF.

RNF12: El sistema debe cumplir con la normativa de protección de datos aplicable al contexto educativo y comunitario.

## **5. Casos de Estudio y Modelo de Negocio**

### **5.1 Caso de Estudio: Gestión de Torneos Deportivos**

En diversas instituciones educativas, clubes y ligas deportivas comunitarias, la gestión de torneos suele llevarse a cabo con herramientas manuales o poco especializadas, como hojas de cálculo, listas en papel y calendarios impresos. Esta forma de trabajo implica varios desafíos:

Errores en la inscripción de equipos.

Confusiones en las fechas y horarios de partidos.

Falta de control sobre el registro de resultados.

Ausencia de estadísticas automáticas y actualizadas.

Dificultad para generar reportes y visibilidad del torneo.

El sistema propuesto busca automatizar completamente este proceso, reduciendo errores y mejorando la eficiencia administrativa y deportiva.

### **5.2 Solución Propuesta**

El Sistema de Gestión de Torneos Deportivos digitaliza todo el ciclo de vida de un torneo:

Registro de usuarios con distintos roles.

Creación y configuración de torneos por parte del organizador.

Inscripción de equipos y jugadores.

Generación automática de calendarios de partidos.

Registro de resultados por el árbitro.

Actualización automática de estadísticas y posiciones.

Todo esto a través de una interfaz sencilla, validaciones automáticas y control de acceso por roles.



## Ejemplo de implementación del modelo del sistema

### 5.3 Modelo de Negocio Académico

Ámbito	Justificación
Escuelas y preparatorias	Fomentar la organización de torneos internos y de convivencia.
Ligas comunitarias	Agilizar la gestión sin depender de sistemas comerciales costosos.
Universidades	Integrar con áreas deportivas y brindar experiencia práctica a estudiantes.

### 5.4 Objetivo del Caso de Estudio

El caso de estudio busca demostrar cómo un sistema desarrollado con enfoque ADOO y ciclo de vida iterativo puede:

Resolver problemas reales detectados en la gestión de torneos.

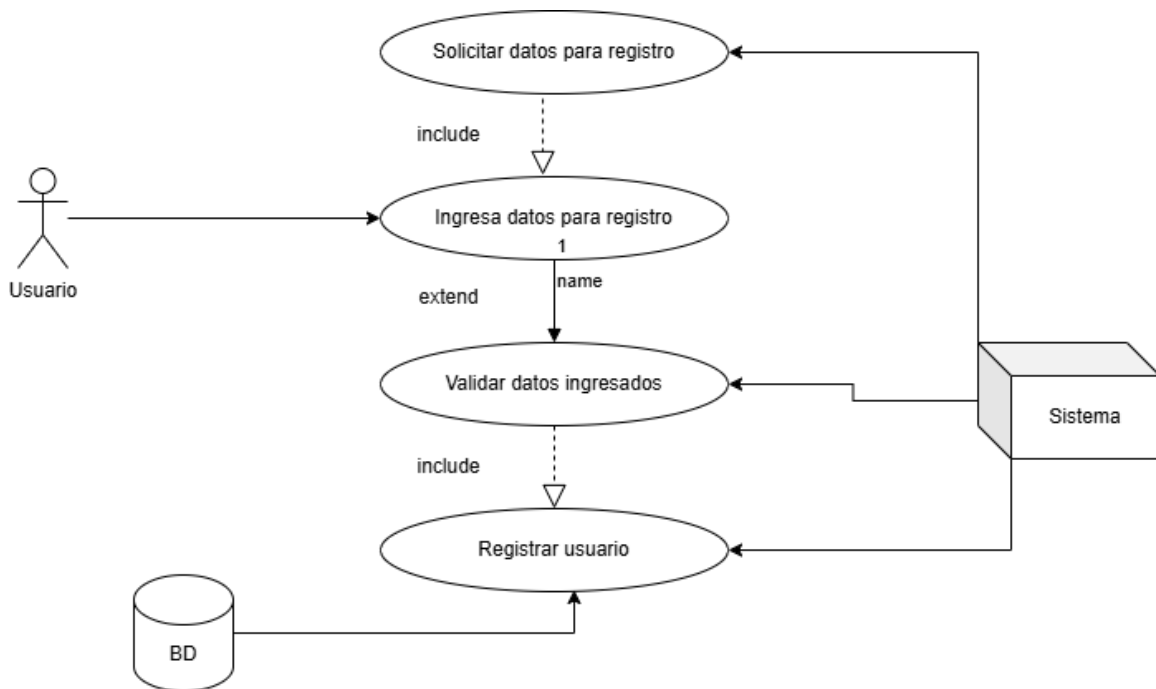
Adaptarse a distintos niveles de complejidad (desde torneos pequeños hasta eventos grandes).

Evolucionar con base en retroalimentación constante de los usuarios.

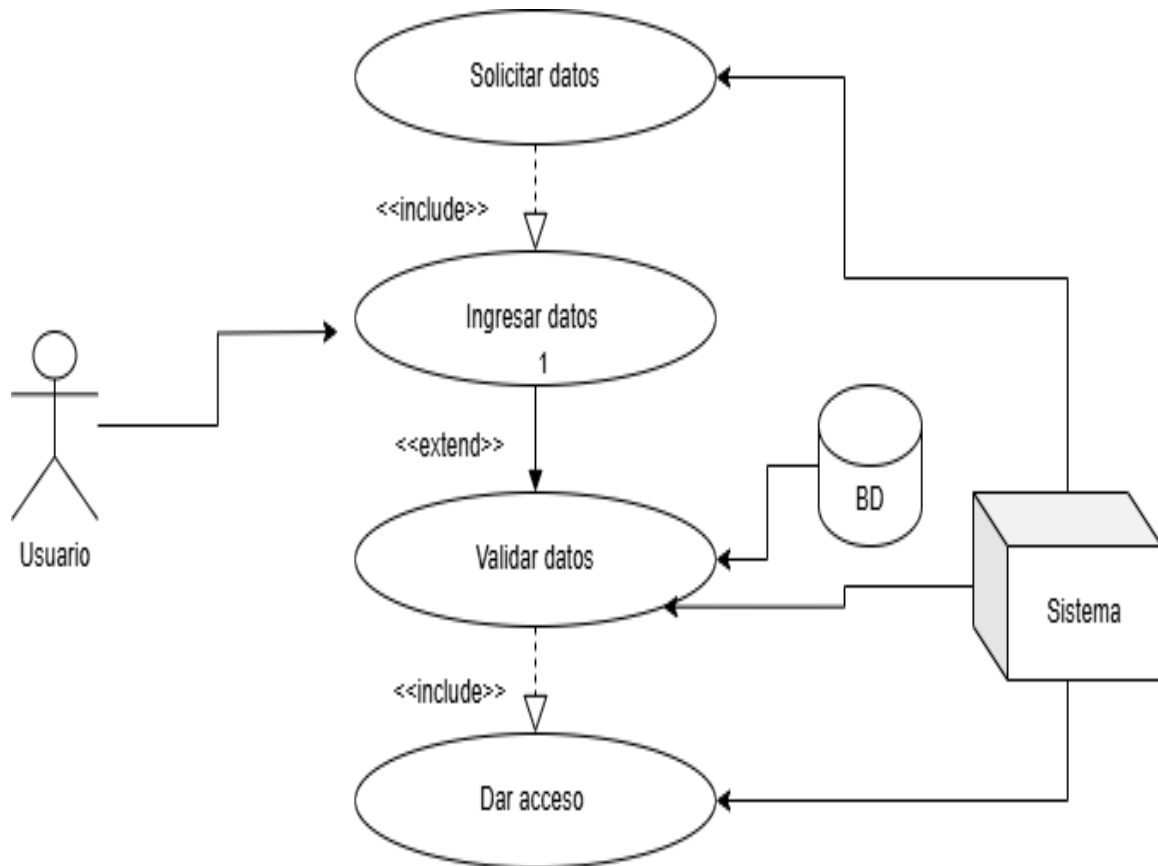
## 6. Casos de Uso Detallados

Esta sección detalla los principales casos de uso que describen cómo los usuarios interactúan con el sistema. Cada caso de uso define una funcionalidad específica, asociada a uno o más actores, y detalla el comportamiento esperado del sistema.

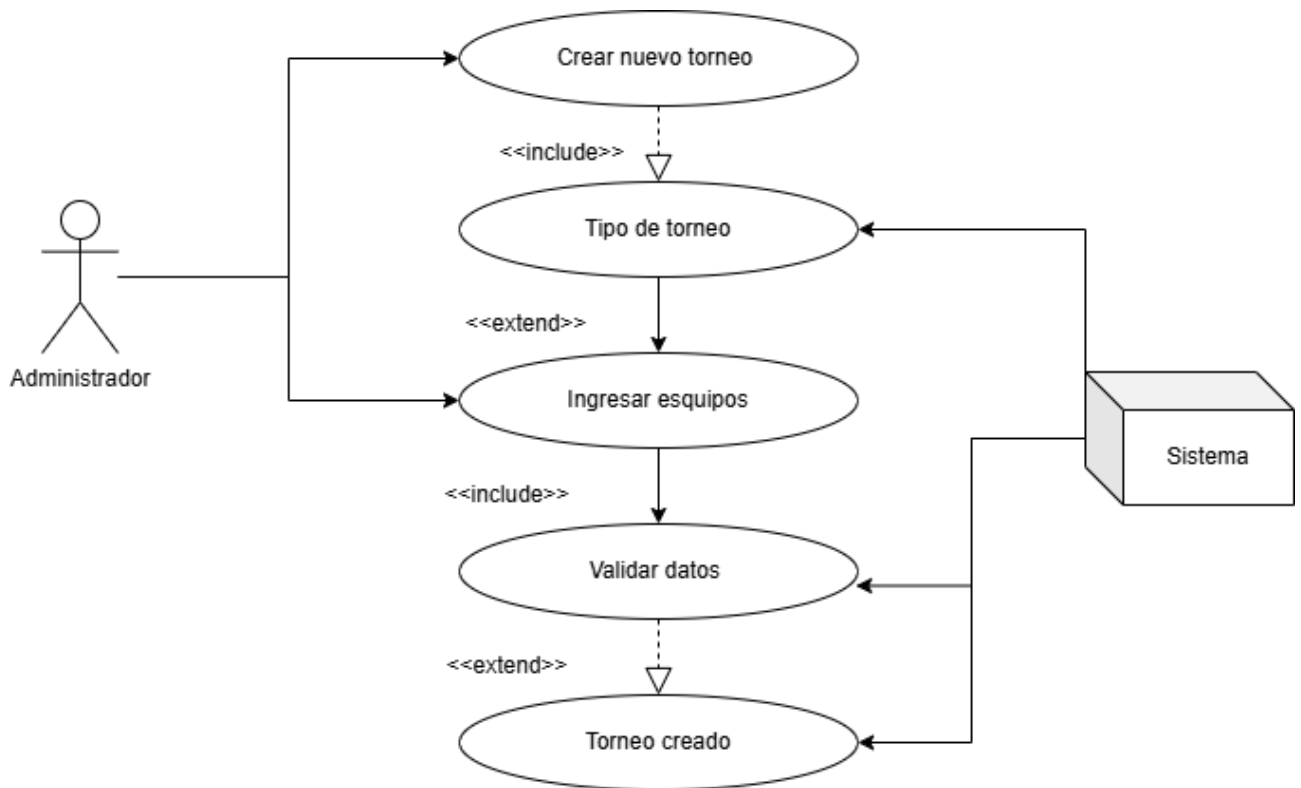
Incluye CU01: Registro, CU02: Crear torneo, CU03: Inscribir equipo, CU04: Registrar resultados, cada uno con su flujo de eventos y actores.



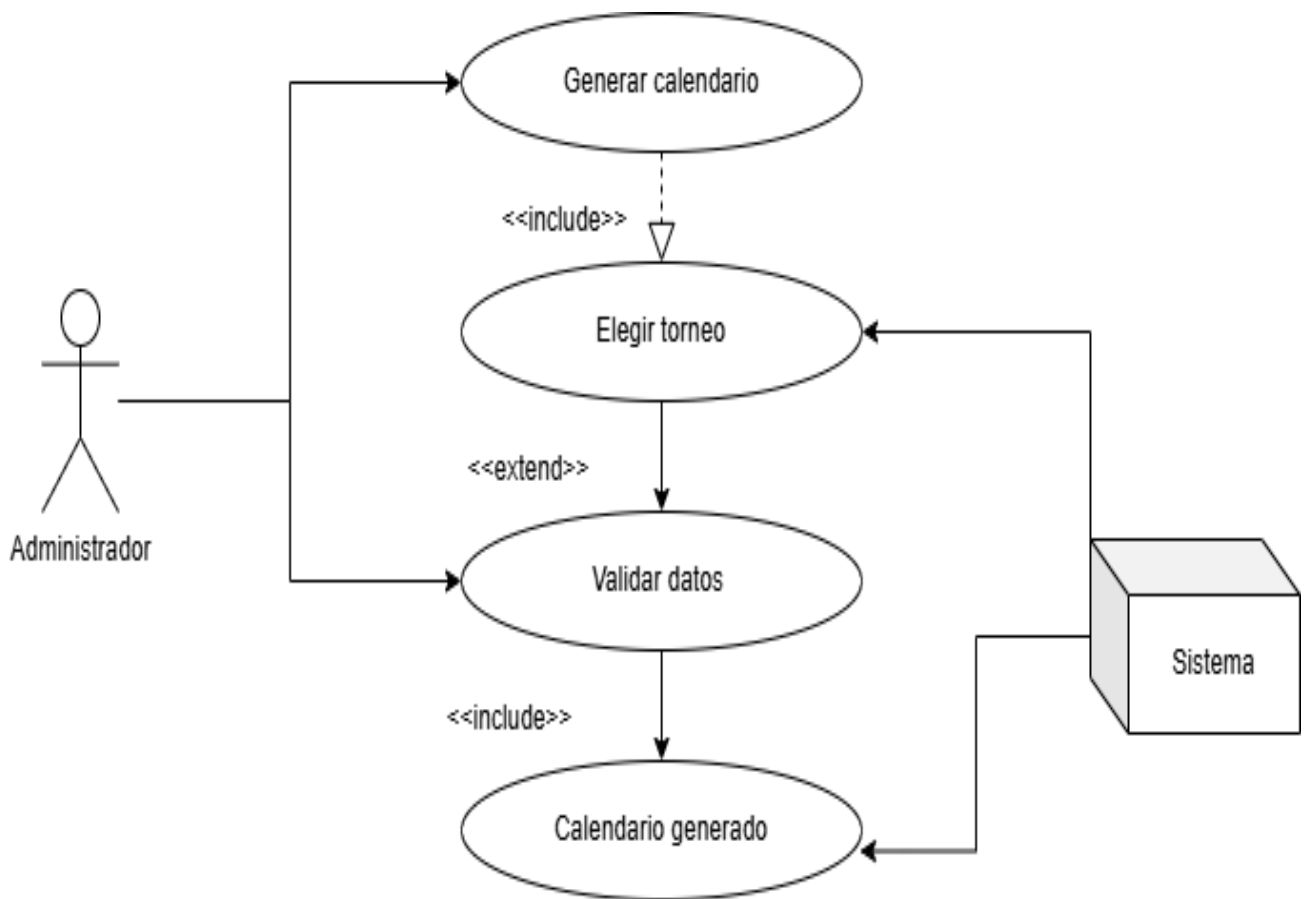
<b>Caso de uso ID:</b> Cu-01	<b>Registrar usuario</b>	
<b>Descripción breve</b>	Un usuario ingresa sus datos al sistema para por crear una cuenta.	
<b>Actores primarios</b>	Jugador, Administrador	
<b>Precondiciones</b>	No debe existir otra cuenta con las mismas credenciales a utilizar	
<b>Postcondiciones</b>	El usuario ingresa con su cuenta al sistema	
<b>Principal escenario de éxito</b>	El usuario tiene éxito al crear su cuenta	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. El usuario ingresas los datos solicitados por el sistema		1. El sistema solicita al usuario ingresar sus credenciales 2. El sistema comprueba que las credenciales sean válidas y no estén repetidas 3. El sistema genera un acceso para el usuario
<b>Flujos alternativos</b>		
<b>2a</b>	el sistema detecta que las credenciales han sido usadas anteriormente	
<b>2b</b>	El sistema detecta campos vacíos en las entradas del usuario	



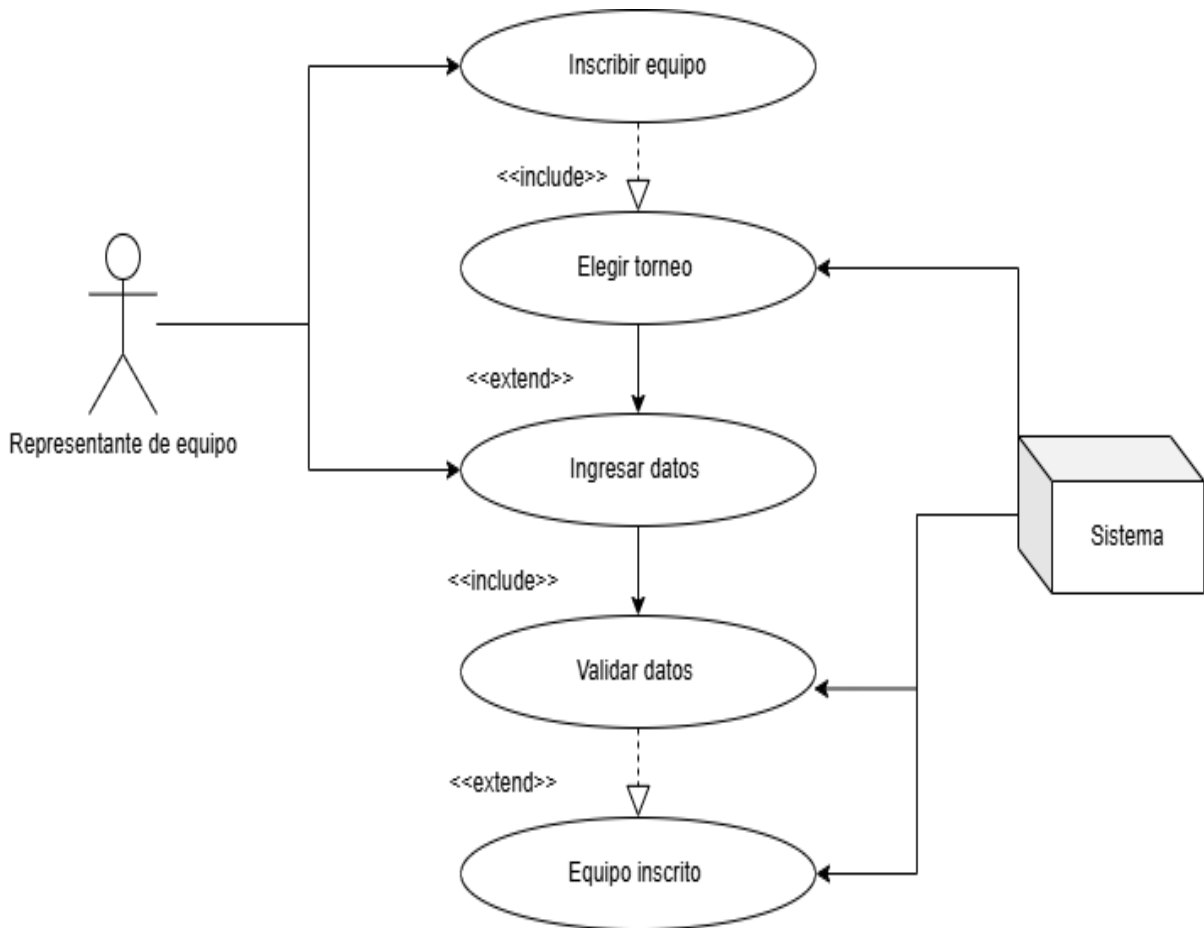
<b>Caso de uso ID:</b> Cu-02	<b>Iniciar sesión</b>	
<b>Descripción breve</b>	El usuario ingresa con sus credenciales en la plataforma.	
<b>Actores primarios</b>	Usuario	
<b>Precondiciones</b>	Tener una cuenta registrada.	
<b>Postcondiciones</b>	Usuario autenticado y con acceso a funcionalidades.	
<b>Principal escenario de éxito</b>	Usuario ingresa credenciales correctas y accede a su perfil.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Introduce correo y contraseña y hace clic en “Iniciar sesión”.		1. El sistema valida credenciales y redirige a la página principal.
<b>Flujos alternativos</b>		
<b>1a</b>	Credenciales incorrectas → Se muestra mensaje de error y se solicita reintento.	



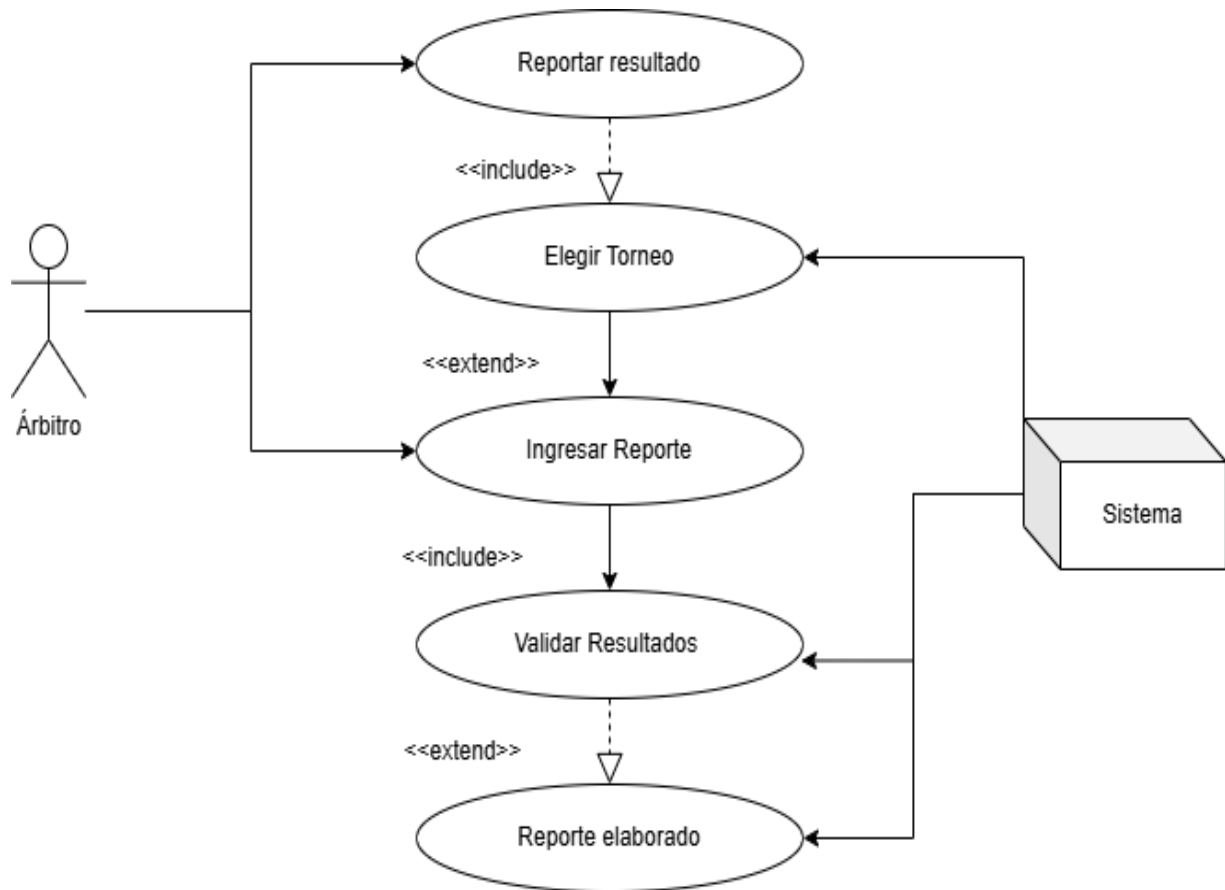
<b>Caso de uso ID:</b> Cu-03	<b>Crear torneo</b>	
<b>Descripción breve</b>	Organizador genera un torneo en la plataforma con sus respectivas configuraciones.	
<b>Actores primarios</b>	Organizador	
<b>Precondiciones</b>	Tener cuenta y permisos de organizador	
<b>Postcondiciones</b>	Torneo creado y listo para inscripciones	
<b>Principal escenario de éxito</b>	Organizador completa los datos y genera el torneo.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Rellena el formulario de creación del torneo y lo confirma.		1. Torneo creado y visible en el sistema.
<b>Flujos alternativos</b>		
<b>1a</b>	Datos incompletos → Se solicita completar información antes de proceder.	



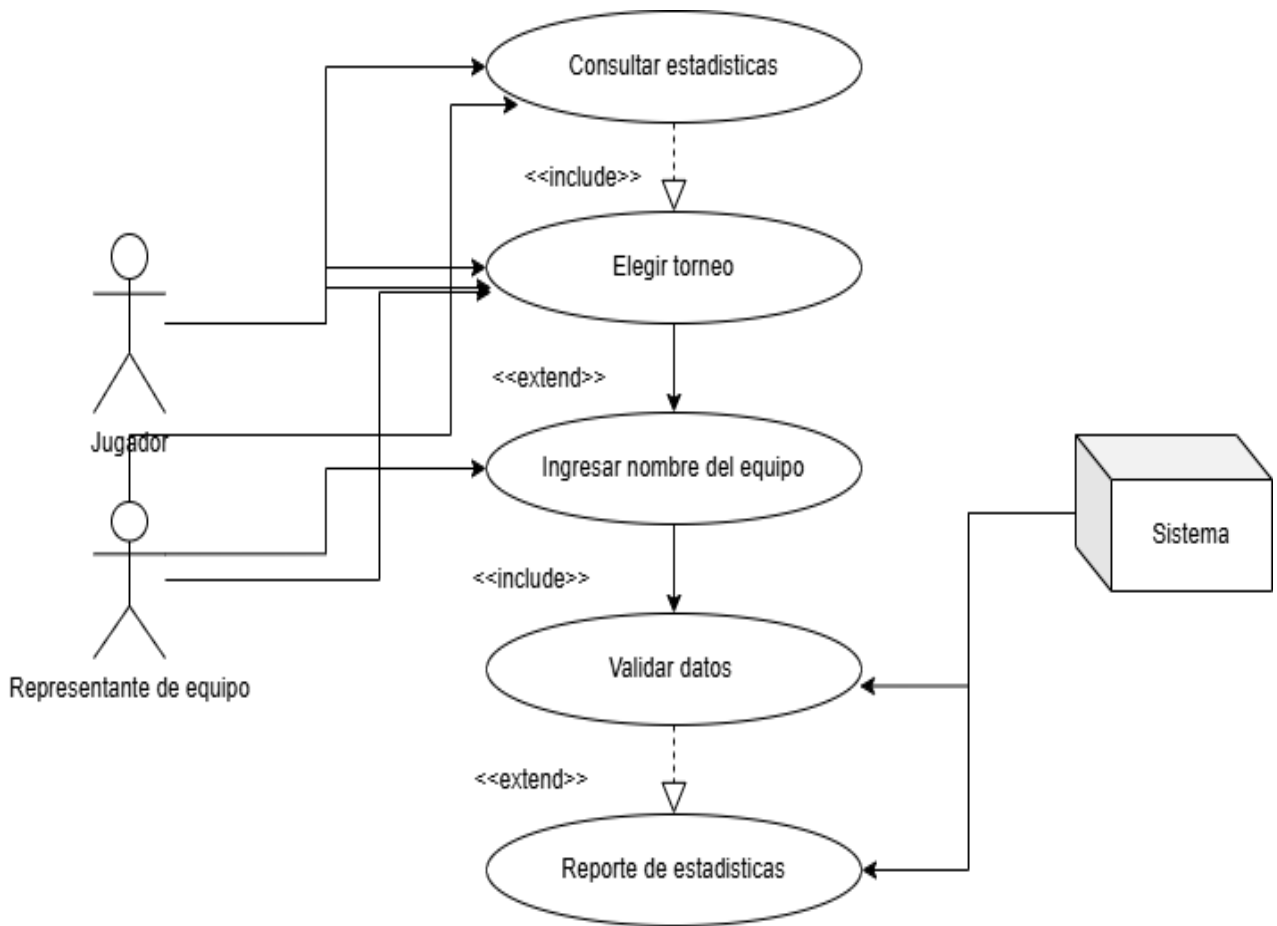
<b>Caso de uso ID:</b> Cu-04	<b>Generar calendario</b>	
<b>Descripción breve</b>	Se genera el calendario de partidos con base en los equipos inscritos.	
<b>Actores primarios</b>	Organizador	
<b>Precondiciones</b>	Tener un torneo creado con equipos inscritos.	
<b>Postcondiciones</b>	Calendario disponible para consulta.	
<b>Principal escenario de éxito</b>	Organizador confirma la generación y se genera el calendario correctamente.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1. Hace clic en “Generar calendario” en el torneo creado.	1. Sistema calcula fechas y publica el calendario.	
<b>Flujos alternativos</b>		
<b>1a</b>	No hay suficientes equipos → Se muestra mensaje indicando que faltan inscripciones.	



<b>Caso de uso ID:</b> Cu-05	<b>Inscribir equipo</b>	
<b>Descripción breve</b>	Representante registra su equipo en un torneo.	
<b>Actores primarios</b>	Representante de equipo	
<b>Precondiciones</b>	Tener cuenta y torneo disponible.	
<b>Postcondiciones</b>	Equipo registrado en el torneo.	
<b>Principal escenario de éxito</b>	Representante registra su equipo con datos correctos y se confirma la inscripción.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Completo formulario de inscripción y lo envía.		1. El sistema registra el equipo en el torneo y muestra confirmación.
<b>Flujos alternativos</b>		
<b>1a</b>	Datos inválidos → Se muestra mensaje de error y se requiere corrección.	

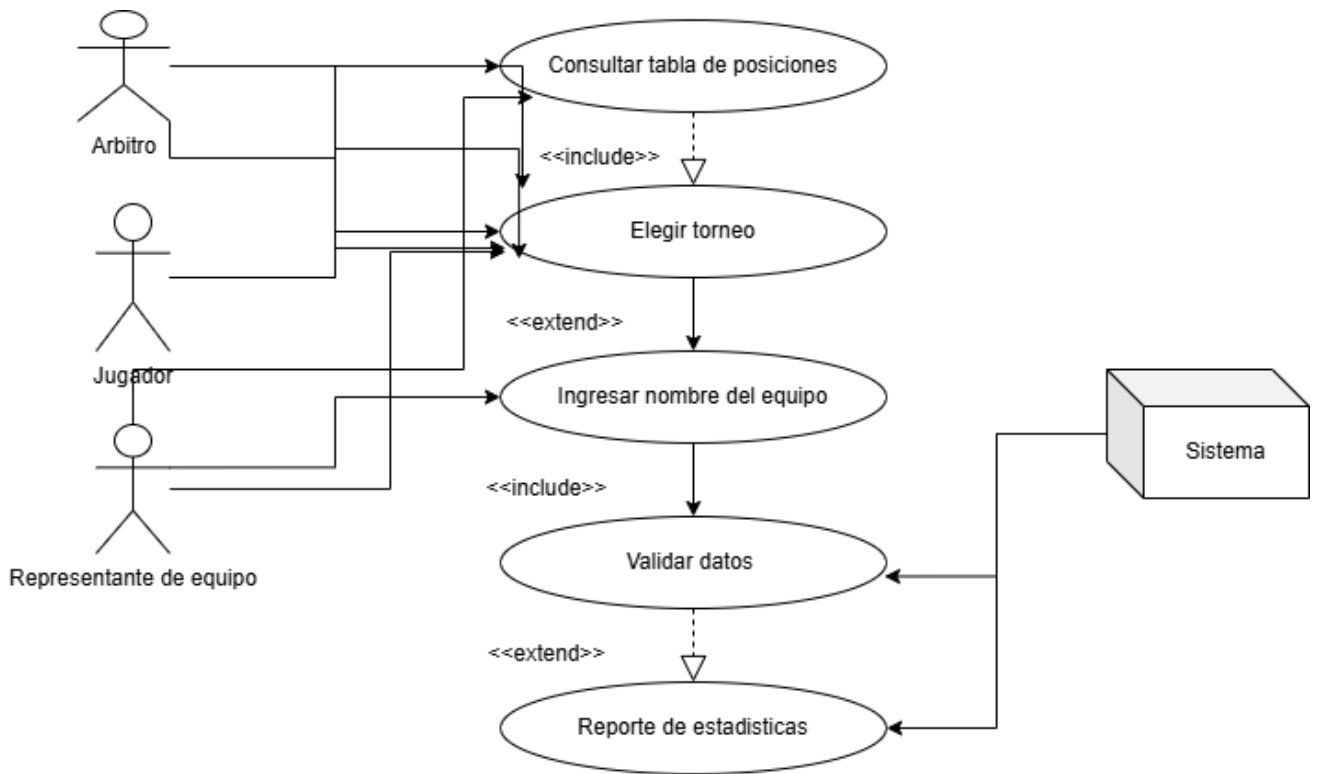


<b>Caso de uso ID:</b> Cu-06	<b>Reportar resultado</b>	
<b>Descripción breve</b>	Árbitro sube el resultado de un partido a la plataforma.	
<b>Actores primarios</b>	Árbitro	
<b>Precondiciones</b>	Partido jugado y registrado en el sistema.	
<b>Postcondiciones</b>	Resultado reflejado en estadísticas y tabla de posiciones.	
<b>Principal escenario de éxito</b>	Jugador accede a su perfil y revisa su desempeño en el torneo.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Hace clic en “Ver estadísticas” desde su cuenta.		1. Se muestran estadísticas detalladas del jugador y el equipo.
<b>Flujos alternativos</b>		
<b>1a</b>	Torneo no tiene resultados aún → Se muestra mensaje indicando falta de datos.	

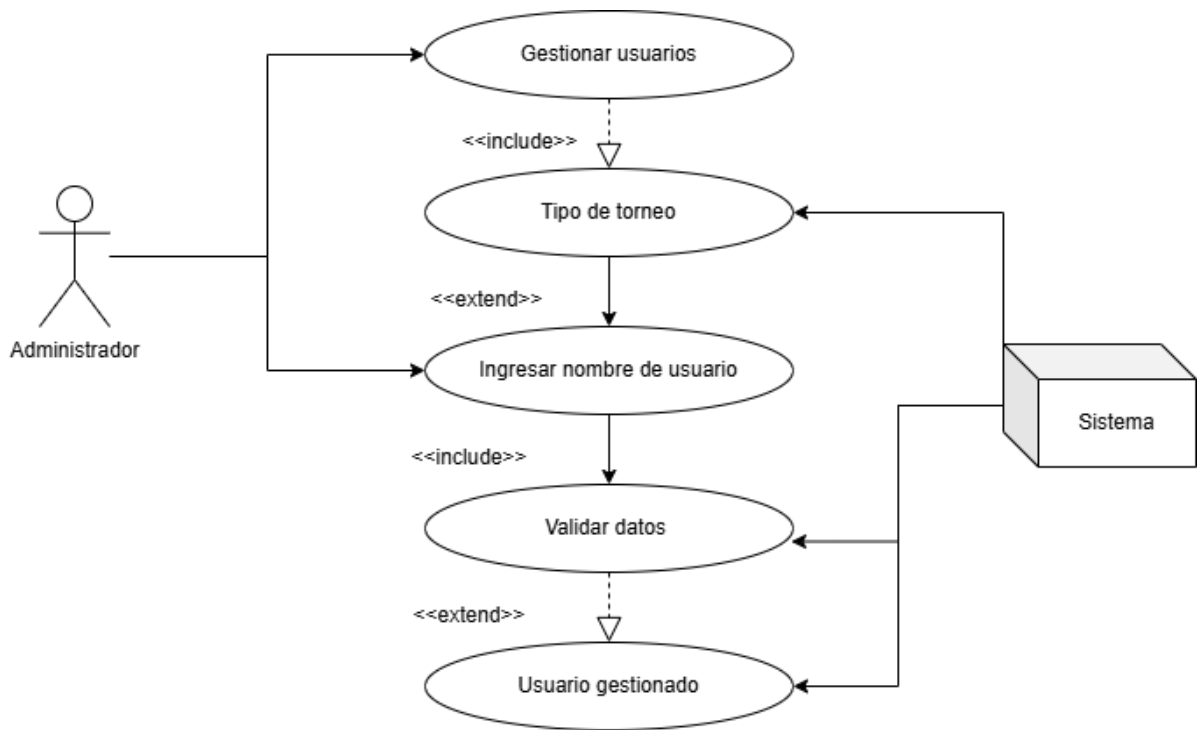


<b>Caso de uso ID:</b> Cu-07	<b>Consultar estadísticas</b>	
<b>Descripción breve</b>	Jugadores pueden revisar sus estadísticas personales y del torneo.	
<b>Actores primarios</b>	Jugador	
<b>Precondiciones</b>	Torneo activo con partidos jugados.	
<b>Postcondiciones</b>	Información accesible de desempeño	
<b>Principal escenario de éxito</b>	Jugador accede a su perfil y revisa su desempeño en el torneo.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Hace clic en “Ver estadísticas” desde su cuenta.		1. Se muestran estadísticas detalladas del jugador y el equipo.
<b>Flujos alternativos</b>		
<b>1a</b>	Torneo no tiene resultados aún → Se muestra mensaje indicando falta de datos.	





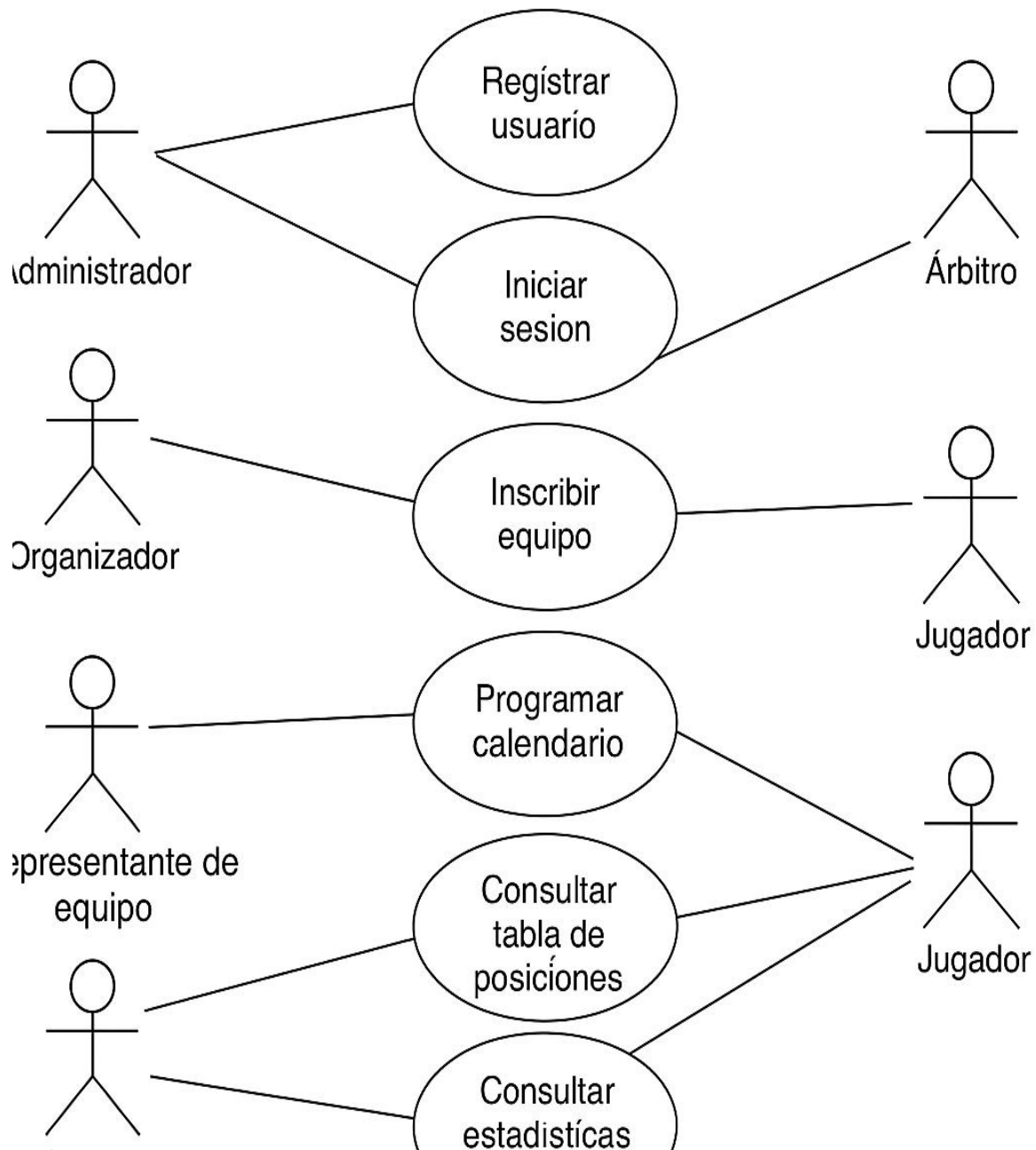
<b>Caso de uso ID:</b> Cu-08	<b>Consultar tabla de posiciones</b>	
<b>Descripción breve</b>	Jugadores y representantes consultan la clasificación de los equipos en el torneo.	
<b>Actores primarios</b>	Jugador/Representante	
<b>Precondiciones</b>	Resultados reportados de los partidos.	
<b>Postcondiciones</b>	Tabla actualizada con posiciones de equipos.	
<b>Principal escenario de éxito</b>	Usuario revisa la tabla actualizada del torneo.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Accede a la sección de clasificación y selecciona el torneo.		1. Sistema muestra tabla con posiciones y puntos acumulados.
<b>Flujos alternativos</b>		
<b>1a</b>	No hay partidos jugados → Se muestra mensaje indicando que no hay datos disponibles	



<b>Caso de uso ID:</b> Cu-09	<b>Gestionar usuarios</b>	
<b>Descripción breve</b>	Administrador maneja cuentas de usuarios y permisos.	
<b>Actores primarios</b>	Administrador	
<b>Precondiciones</b>	Ser administrador del sistema.	
<b>Postcondiciones</b>	Control de accesos y gestión de cuentas.	
<b>Principal escenario de éxito</b>	Administrador edita permisos y gestiona cuentas.	
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Accede al panel de administración y realiza cambios.		1. Sistema actualiza configuraciones y confirma cambios.
<b>Flujos alternativos</b>		
<b>1a</b>	Permisos insuficientes → Se muestra mensaje de restricción y se deniega la acción.	

## 7. Diagrama de Casos de Uso (Descripción Visual)

### Sistema de Gestión de Torneos Deportivos



### 7.1 Actores del Sistema

**Administrador:** Gestiona usuarios y configura parámetros globales del sistema.

**Organizador:** Crea torneos, define reglas, gestiona participantes.

**Representante de equipo:** Inscribe equipos y consulta resultados.

**Árbitro:** Registra los resultados de los partidos.

**Jugador:** Consulta estadísticas y desempeño.

**Usuario (General):** Accede a funciones básicas como iniciar sesión y registrarse.

### 7.2 Funciones del Sistema

Caso de uso	Actor principal
Registrar usuario	Usuario
Iniciar sesión	Usuario
Crear torneo	Organizador
Generar Calendario	Organizador
Inscribir equipo	Representante de equipo
Reportar resultado	Arbitro
Consultar estadísticas	Jugador
Consultar tabla de posiciones	Jugador/Representante
Gestionar usuarios	Administrador

### 7.3 Especificaciones de Casos de Uso

ID-CU-01	Registrar Usuario
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	El caso de uso "Registrar Usuario" permite a los usuarios crear una cuenta en el sistema, especificando datos como nombre, correo electrónico, contraseña y rol.
Actor(es) primarios(s):	Usuario (jugador, organizador, árbitro, representante)
Actor(es) secundarios(s):	Administrador (solo si es aprobado)
Precondiciones(es):	El usuario no debe estar registrado previamente.
Happy Path:	

	1. El usuario accede al formulario de registro. 2. Ingresa nombre, correo, contraseña y rol. 3. El sistema valida los datos. 4. Se guarda el usuario. 5. Se muestra mensaje de éxito.
Post-condicion(es):	El usuario queda registrado y puede iniciar sesión.
Alternativos:	FA-01: Datos incompletos. FA-02: Correo ya registrado. Se muestra mensaje de error.

ID-CU-02	Iniciar Sesión
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	Permite a los usuarios autenticarse en el sistema mediante correo y contraseña.
Actor(es) primarios(s):	Usuario (jugador, organizador, árbitro, representante, administrador)
Actor(es) secundarios(s):	Ninguno
Precondiciones(es):	El usuario debe estar previamente registrado.
Happy Path:	1. El usuario accede al formulario. 2. Ingresa sus credenciales. 3. El sistema valida los datos. 4. Se permite el acceso.
Post-condicion(es):	El usuario accede al sistema.
Alternativos:	FA-01: Credenciales incorrectas. FA-02: Campos vacíos. Se muestra mensaje de error.

ID-CU-03	Crear Torneo
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo

Descripción:	Permite al organizador crear un nuevo torneo con datos como nombre, tipo y fechas.
Actor(es) primarios(s):	Organizador
Actor(es) secundarios(s):	Ninguno
Precondiciones(es):	El organizador debe estar autenticado.
Happy Path:	1. Accede a la opción "Crear torneo". 2. Llena los campos requeridos. 3. El sistema guarda la información.
Post-condicion(es):	Torneo creado exitosamente.
Alternativos:	FA-01: Campos incompletos. FA-02: Fechas inválidas. Se muestra error.

ID-CU-04	Inscribir Equipo
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	Permite a un representante de equipo inscribir su equipo en un torneo disponible.
Actor(es) primarios(s):	Representante de equipo
Actor(es) secundarios(s):	Ninguno
Precondiciones(es):	El torneo debe estar disponible y el equipo debe estar registrado.
Happy Path:	1. Accede a la lista de torneos. 2. Selecciona uno y su equipo. 3. Confirma la inscripción.
Post-condicion(es):	El equipo queda inscrito.
Alternativos:	FA-01: Equipo ya inscrito. FA-02: Torneo no disponible.

ID-CU-05	Generar Calendario
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo

Descripción:	Genera automáticamente el calendario de partidos del torneo con base en los equipos inscritos.
Actor(es) primarios(s):	Organizador
Actor(es) secundarios(s):	Sistema
Precondiciones(es):	El torneo debe tener mínimo 4 equipos inscritos.
Happy Path:	1. Accede al torneo. 2. Elige "Generar calendario". 3. El sistema crea los enfrentamientos.
Post-condicion(es):	Calendario disponible.
Alternativos:	FA-01: Menos de 4 equipos. FA-02: Error en la generación. Se muestra mensaje.

ID-CU-06	Reportar Resultado
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	Permite al árbitro asignado registrar el marcador final de un partido.
Actor(es) primarios(s):	Árbitro
Actor(es) secundarios(s):	Sistema
Precondiciones(es):	El árbitro debe estar autenticado y el partido debe estar programado.
Happy Path:	1. Accede al listado de partidos. 2. Selecciona partido. 3. Ingresa resultado. 4. Se guarda y actualiza tabla.
Post-condicion(es):	Resultado guardado y clasificación actualizada.
Alternativos:	FA-01: Campos vacíos o negativos. Se muestra mensaje de error.

ID-CU-07	Consultar Estadísticas
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	El jugador puede consultar su rendimiento y estadísticas de su equipo.
Actor(es) primarios(s):	Jugador

Actor(es) secundarios(s):	Sistema
Precondiciones(es):	Deben existir partidos jugados con resultados.
Happy Path:	1. Accede a "Mis estadísticas". 2. Selecciona torneo. 3. El sistema muestra los datos.
Post-condicion(es):	Estadísticas consultadas correctamente.
Alternativos:	FA-01: No hay datos registrados.

ID-CU-08	Consultar Tabla de Posiciones
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	Permite visualizar la clasificación de equipos en un torneo.
Actor(es) primarios(s):	Jugador / Representante de equipo
Actor(es) secundarios(s):	Sistema
Precondiciones(es):	Debe haber al menos un resultado registrado.
Happy Path:	1. Accede a la tabla. 2. Selecciona torneo. 3. Visualiza posiciones.
Post-condicion(es):	Clasificación mostrada correctamente.
Alternativos:	FA-01: No hay resultados registrados.

ID-CU-09	Gestionar Usuarios
Fecha:	2025
Autor:	Estudiante de ingeniería / equipo de desarrollo
Descripción:	El administrador del sistema puede editar roles, activar o desactivar cuentas de usuarios registrados.
Actor(es) primarios(s):	Administrador
Actor(es) secundarios(s):	Sistema
Precondiciones(es):	El administrador debe estar autenticado.
Happy Path:	1. Accede al panel de gestión de usuarios. 2. Selecciona usuario. 3. Modifica rol o estado. 4. Confirma cambios.



Post-condicion(es):	Cambios aplicados correctamente en la base de datos.
Alternativos:	FA-01: Usuario no encontrado. FA-02: Error de permisos. El sistema muestra mensajes de error adecuados.

## 8. Modelo del Dominio (Descripción Conceptual)

### 8.1 Modelo del Dominio

El modelo del dominio representa las entidades principales del sistema, sus atributos y la forma en que se relacionan entre sí. Está centrado en el conocimiento del negocio (torneos deportivos) y no en los detalles técnicos.

#### Las principales entidades (clases) del sistema son:

Usuario (atributos: ID, nombre, contraseña, rol)

Torneo (ID, nombre, tipo, fechaInicio, fechaFin)

Equipo (ID, nombre, representantes, jugadores)

Jugador (ID, nombre, número de camiseta, posición)

Partido (ID, equipoLocal, equipoVisitante, fecha, resultado)

Resultado (golesEquipoLocal, golesEquipoVisitante)

Estadística (partidosJugados, puntos, golesAFavor, golesEnContra)

Rol (Administrador, Organizador, Árbitro, Representante, Jugador)

### 8.2 Relaciones

1.-Un Usuario tiene un Rol (herencia o composición).

2.-Un Torneo tiene muchos Equipos.

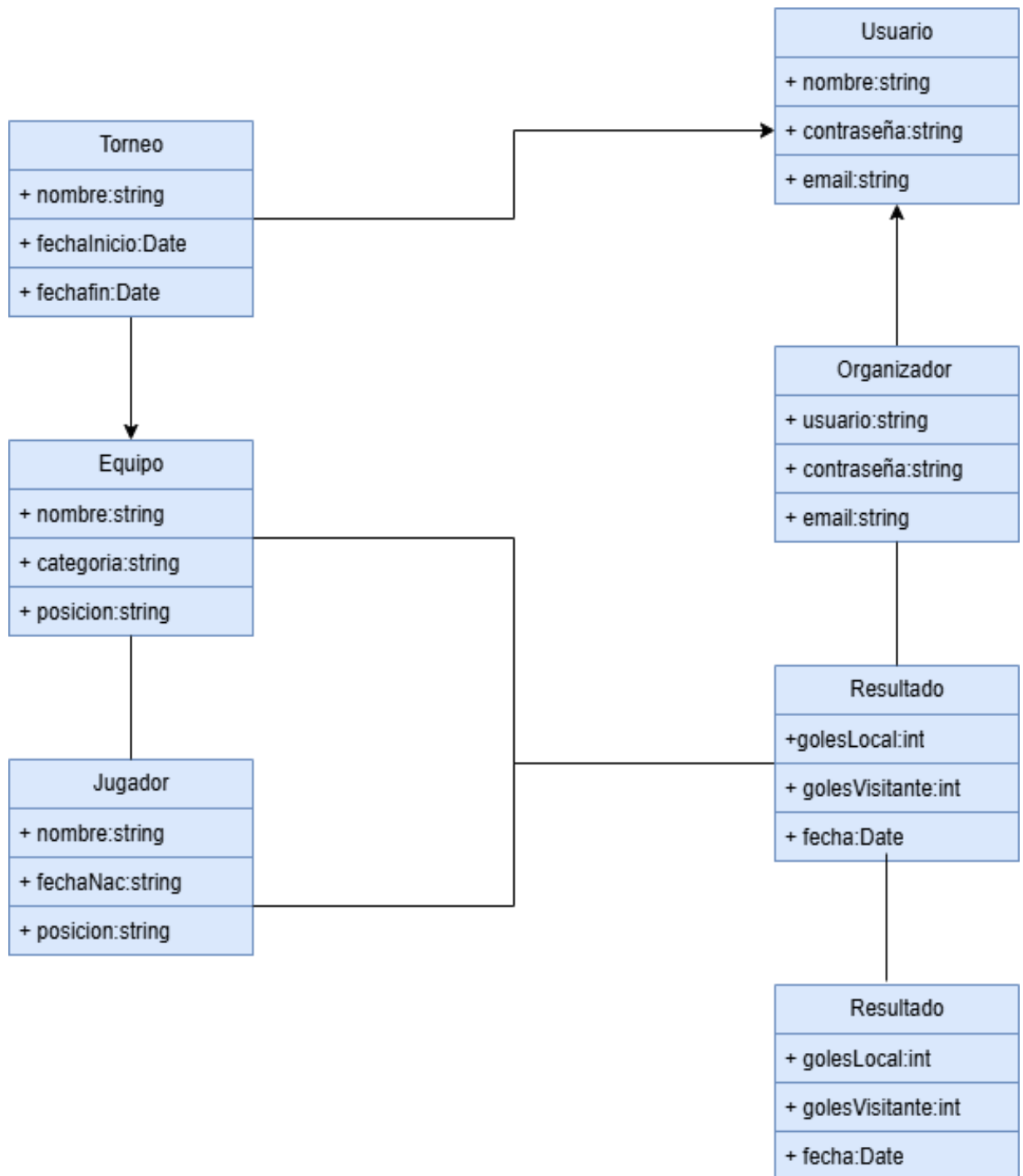
3.-Un Equipo tiene muchos Jugadores.

4.-Un Torneo tiene muchos Partidos.

5.-Cada Partido está vinculado a dos Equipos y tiene un Resultado.

6.-Las Estadísticas se generan para cada Equipo y Jugador en función de los partidos jugados.

### **8.3 Diagrama de Clases UML**



## 9. Diagrama de Clases

El diagrama de clases representa la estructura estática del sistema y muestra cómo se organizan sus componentes en términos de clases, atributos, métodos y relaciones entre ellas. Este diagrama es parte esencial del análisis y diseño orientado a objetos (ADOO) y permite definir la base lógica sobre la cual se construirá el sistema.

### 9.1 Clases Principales

Las principales clases identificadas en el sistema son:

**Usuario:** Clase base con atributos comunes como nombre de usuario, contraseña y correo electrónico. Dependiendo del rol, puede especializarse como Organizador, Árbitro, etc.

**Organizador:** Hereda de Usuario. Tiene la capacidad de crear, editar y eliminar torneos.

**Torneo:** Contiene la información general del evento deportivo, incluyendo fechas, tipo y equipos participantes.

**Equipo:** Representa a los equipos inscritos. Tiene atributos como nombre, categoría y posición.

**Jugador:** Perteneciente a un equipo. Contiene información como nombre, posición y número de camiseta.

**Partido:** Relaciona dos equipos con fecha y resultado.

**Resultado:** Guarda el marcador del partido y la fecha en que se jugó.

### 9.2 Relaciones entre clases

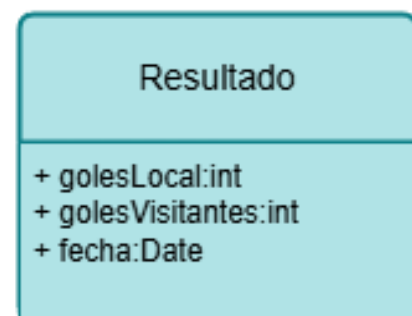
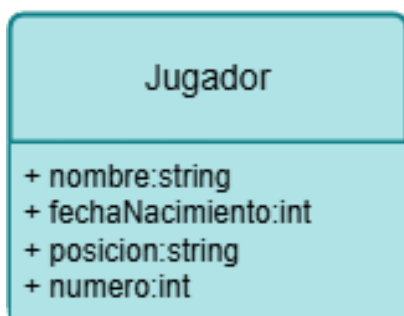
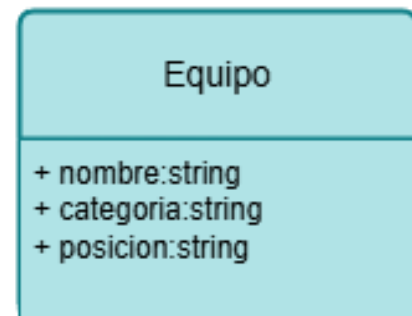
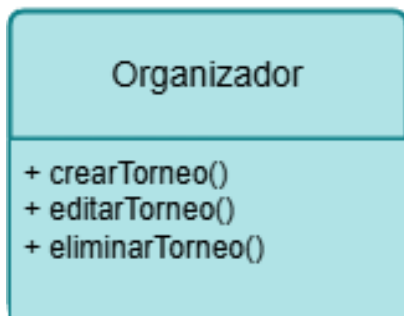
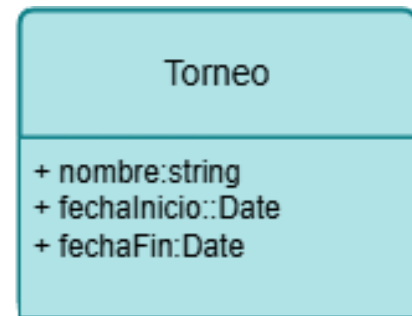
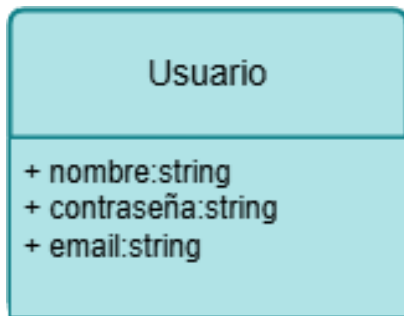
Un Usuario puede ser un Organizador, quien administra Torneos.

Un Torneo tiene múltiples Equipos.

Un Equipo contiene varios Jugadores.

Un Torneo contiene múltiples Partidos, cada uno con su Resultado.

## Diagrama UML



## 10.Iteraciones del Proyecto

El desarrollo del sistema de gestión de torneos deportivos se organizó en iteraciones. Cada iteración representa un ciclo de desarrollo completo, incluyendo análisis, diseño, implementación y pruebas. Este enfoque permite entregar valor incremental al cliente, facilitando la adaptación a cambios y la mejora continua.

### 10.1 Planeación de Iteraciones

Se propusieron las siguientes iteraciones para el proyecto:

Iteración	Objetivo Principal	Duración	Entregables Principales
1	Análisis inicial e identificación de requisitos	1 semana	Documento de requerimientos, actores y CU
2	Diseño de arquitectura y modelos	1 semana	Diagramas UML de clases y casos de uso
3	Implementación del registro e inicio de sesión	2 semanas	Módulo funcional con pruebas unitarias
4	Gestión de torneos y equipos	2 semanas	CRUD para torneos, equipos y jugadores
5	Registro de resultados y clasificación	2 semanas	Registro de partidos y actualización de tabla
6	Validaciones, mejoras y pruebas finales	1 semana	Testing de integración y validaciones finales

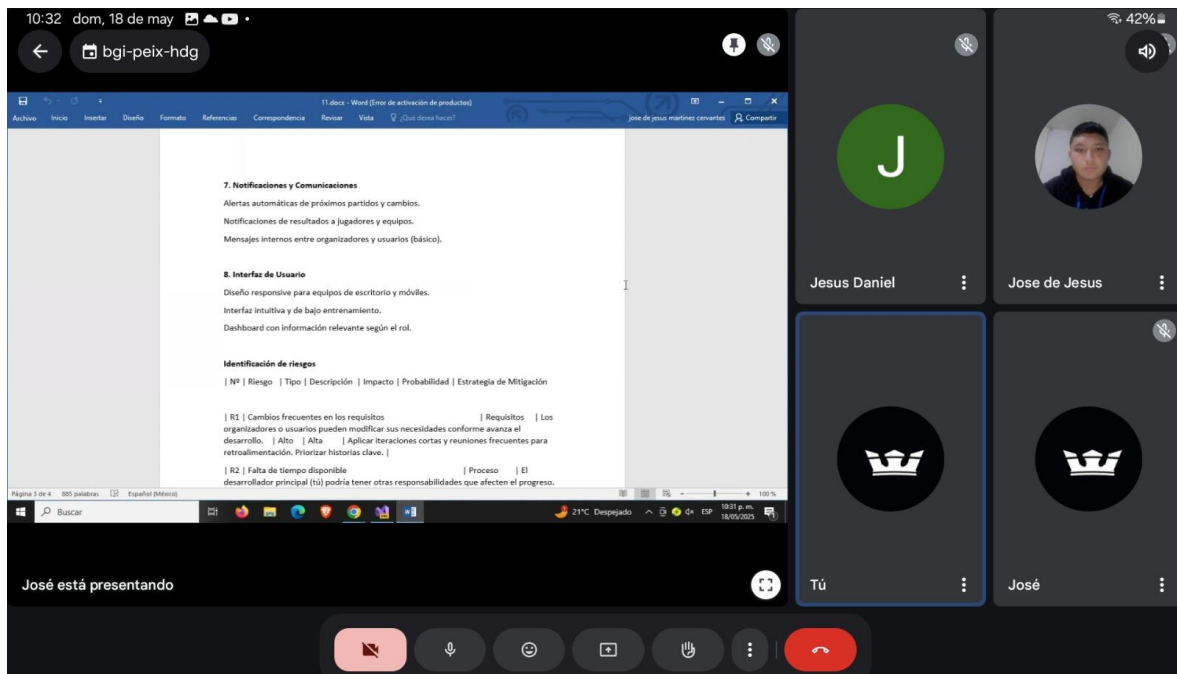
### 10.2 Beneficios de la Iteración

Mejora continua y revisión frecuente del avance

Reducción del riesgo en cada entrega parcial

Mayor capacidad de adaptación a los cambios

Feedback temprano de los usuarios finales



## 11. Diagrama de secuencias

El diagrama de secuencia es un tipo de diagrama de interacción que muestra cómo los objetos interactúan en un escenario particular de un caso de uso, ordenados cronológicamente.

Este tipo de diagrama es útil para entender el flujo de mensajes entre componentes del sistema y clarifica la lógica de negocio implementada en cada operación.

### 11.1 Caso de Uso: CU04 - Registrar Resultado

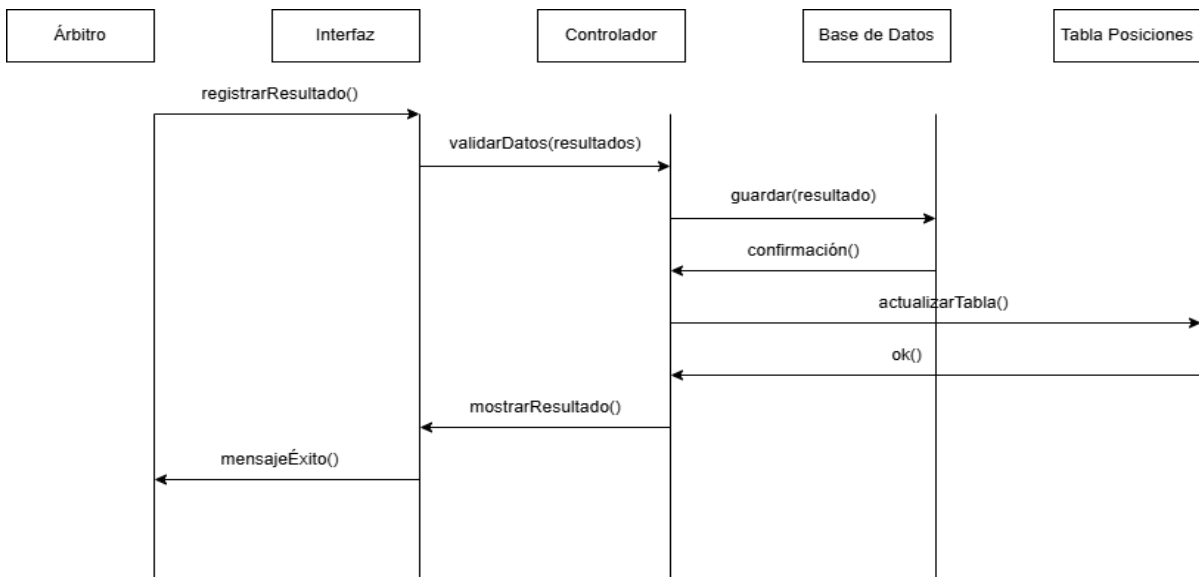
Este diagrama ilustra cómo el actor Árbitro registra el resultado de un partido, desencadenando una serie de acciones internas en el sistema.

## 11.2 Participantes en el Diagrama

Participante	Rol
Árbitro	Usuario que registra el resultado
Interfaz	Pantalla donde se ingresan los datos
Controlador	Lógica de negocio que valida y coordina la acción
BaseDatos	Almacena la información del resultado
TablaPosiciones	Calcula y actualiza las posiciones de los equipos

## 11.3 Flujo de Mensajes

- 1.El Árbitro inicia el proceso desde la Interfaz.
- 2.La interfaz llama al Controlador para validar los datos del resultado.
- 3.Si son válidos, el controlador solicita a la Base de Datos que guarde el resultado.
- 4.La base de datos responde con una confirmación.
- 5.El controlador solicita actualizar la Tabla de Posiciones.
- 6.La tabla responde con éxito.
- 7.El controlador confirma a la interfaz.
- 8.La interfaz muestra un mensaje de éxito al Árbitro.





## 12. Validación y Pruebas

La validación y pruebas son fundamentales para asegurar que el sistema de gestión de torneos deportivos cumple con los requisitos definidos y se comporta de forma confiable ante distintos escenarios.

### 12.1 Estrategia de Pruebas

Se aplicaron pruebas en cada iteración para asegurar la calidad del desarrollo incremental. Las pruebas se dividieron en las siguientes categorías:

Tipo de Prueba	Objetivo
Pruebas Unitarias	Validar el correcto funcionamiento de clases y métodos individuales.
Pruebas de Integración	Asegurar que los módulos interactúan correctamente entre sí.
Pruebas Funcionales	Verificar que cada caso de uso se ejecuta como lo requiere el usuario.
Pruebas de Interfaz	Evaluar que la experiencia visual y de uso sea intuitiva y sin errores.
Pruebas de Rendimiento	Verificar que el sistema responde adecuadamente bajo carga controlada.

### 12.2 Escenarios de Prueba

Caso de Uso	Escenario Validado	Resultado Esperado
CU01 - Iniciar sesión	Usuario con credenciales válidas	Accede al sistema correctamente
CU02 - Registrar torneo	Completa formulario con datos válidos	Torneo almacenado en la base de datos
CU03 - Registrar equipo	Equipo sin nombre	Muestra mensaje de validación
CU04 - Registrar resultado	Árbitro registra partido entre dos equipos	Resultado almacenado y tabla actualizada
CU06 - Consultar tabla	Usuario consulta clasificación actualizada	Se muestran los datos correctamente

### 12.3 Resultados Obtenidos

Se identificaron errores menores de validación en formularios que fueron corregidos.

Todos los casos de uso principales se ejecutaron satisfactoriamente.

El sistema mostró una interfaz estable y sin fallos críticos.

Se documentaron los bugs corregidos en cada iteración para control de calidad.

Caso de Uso	Escenario Validado	Resultado Esperado
CU01 - Iniciar sesión	Usuario con credenciales válidas	Accede al sistema correctamente
CU02 - Registrar torneo	Completa formulario con datos válidos	Torneo almacenado en la base de datos
CU03 - Registrar equipo	Equipo sin nombre	Muestra mensaje de validación
CU04 - Registrar resultado	Árbitro registra partido entre dos equipos	Resultado almacenado y tabla actualizada
CU06 - Consultar tabla	Usuario consulta clasificación actualizada	Se muestran los datos correctamente

## 13. Glosario de Términos y Reglas del Negocio

### 13.1 Glosario de Términos

<b>Término</b>	<b>Definición</b>
Torneo	Conjunto de partidos entre equipos registrados, con un formato definido.
Equipo	Agrupación de jugadores que participan en un torneo.
Jugador	Persona integrante de un equipo.
Árbitro	Usuario autorizado para registrar resultados de los partidos.
Resultado	Información registrada sobre el marcador final de un partido.
Tabla de Posiciones	Clasificación automática basada en resultados de partidos.
Organizador	Usuario responsable de crear y gestionar torneos y equipos.
CRUD	Operaciones básicas de Crear, Leer, Actualizar y Eliminar datos.
Caso de Uso (CU)	Escenario funcional que representa una acción del usuario dentro del sistema.
Validación	Proceso de verificación de entrada correcta de datos.

### **13.2 Reglas del Negocio**

<b>ID</b>	<b>Regla</b>
RN01	Un equipo no puede participar en más de un torneo activo a la vez.
RN02	Un torneo no puede iniciar sin al menos 4 equipos registrados.
RN03	Solo el árbitro asignado puede registrar el resultado de un partido.
RN04	No se puede registrar un partido si alguno de los equipos no está registrado en el torneo.
RN05	La tabla de posiciones debe actualizarse automáticamente tras cada resultado registrado.
RN06	Un jugador no puede estar en más de un equipo dentro del mismo torneo.
RN07	Solo el organizador puede crear, editar o eliminar torneos y equipos.
RN08	Las credenciales de acceso deben ser validadas para iniciar sesión.
RN09	Todo partido debe tener una fecha y hora definida para ser válido.

## **13. Conclusiones del proyecto**

El desarrollo del Sistema de Gestión de Torneos Deportivos permitió aplicar de manera práctica los principios del Análisis y Diseño Orientado a Objetos (ADOO), utilizando la metodología Unified Process Ágil, lo cual favoreció un enfoque iterativo, incremental y adaptable a los cambios.

### **14.1 Logros Alcanzados**

- ✓ Se identificaron claramente los requerimientos funcionales y no funcionales del sistema.
- ✓ Se elaboraron los diagramas UML necesarios: casos de uso, clases, secuencia.
- ✓ Se aplicó correctamente el modelo de negocio, permitiendo la digitalización del proceso tradicional.
- ✓ Se implementaron reglas de negocio coherentes que aseguran la integridad del sistema.
- ✓ El sistema contempla roles diferenciados (organizador, árbitro, jugador) y sus respectivas funciones.

### **14.2 Ventajas del Proceso Ágil**

Permite entregas tempranas y funcionales.

Facilita la retroalimentación constante del usuario.

Mejora la calidad del producto final gracias a la validación continua.

Promueve el trabajo colaborativo entre desarrolladores, analistas y usuarios finales.

### **14.3 Retos Enfrentados**

Ajuste de requerimientos según nuevas necesidades del usuario.

Garantizar la integridad y consistencia de datos en múltiples entidades.

Diseño de una interfaz intuitiva que pueda ser utilizada por personas sin experiencia técnica.

Reunión de equipo debido a los horarios con los que cuenta cada integrante.

### **14.4 Conclusión Final**

Este proyecto permitió consolidar conocimientos clave de ADOO en un entorno realista y estructurado. El sistema propuesto representa una herramienta efectiva y escalable para la gestión digital de torneos deportivos locales, promoviendo la transparencia, organización y eficiencia en cada fase del proceso competitivo.

### Bitácora de Defectos Corregidos

ID	Fecha	Módulo Afectado	Descripción del Defecto	Tipo de Defecto	Solución Aplicada	Estado
D-01	17/05/2025	Inicio de Sesión	El sistema permitía campos vacíos al intentar iniciar sesión.	Validación	Se implementó verificación para evitar campos nulos o vacíos.	Corregido
D-02	18/05/2025	Registro de Equipos	Permitía registrar equipos con el mismo nombre duplicado.	Lógica de negocio	Se añadió validación única por nombre en la base de datos.	Corregido
D-03	19/05/2025	Registrar Resultado	Permitía ingresar goles negativos en partidos.	Validación	Se agregó restricción de valores $\geq 0$ .	Corregido
D-04	21/05/2025	Tabla de Posiciones	No actualizaba los puntos tras registrar resultado.	Lógica de cálculo	Se ajustó función que calcula puntos automáticamente.	Corregido
D-05	23/05/2025	Interfaz de Torneos	El botón "Crear torneo" no respondía tras llenar el formulario.	UI/UX	Se solucionó bug en el evento del botón con control de estado.	Corregido
D-06	24/05/2025	Consultar Estadísticas	Estadísticas mostraban datos incompletos o sin orden.	Presentación	Se ordenaron datos por partidos jugados y se corrigió formato.	Corregido
D-07	24/05/2025	Creación caso de uso	Fue elaborado de manera errónea ya que no se enlazo correctamente el flujo.	Colocación	Se relacionó de manera correcta dándole el flujo que era el indicado	Corregido