

UNIVERSIDAD DE TARAPACÁ  
FACULTAD DE INGENIERÍA  
INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA  
ARICA – CHILE



**Sistema de gestión y seguimientos de eventos recreativos del Departamento  
de Deporte y Recreación de la Municipalidad de Arica**

**Equipo de desarrollo: Daniel Alday  
Diego Honores**

**Empresa: Municipalidad de Arica  
Curso: Proyecto IV  
Profesor: Diego Aracena Pizarro**

**05 de Noviembre del 2025  
Arica, Chile**



# Índice

<b>I. Introducción</b>	<b>3</b>
<b>II. Definición del problema</b>	<b>4</b>
Problema	4
Solución	4
<b>III. Objetivos</b>	<b>5</b>
Objetivo General	5
Objetivos Específicos	5
<b>IV. Requerimientos del sistema</b>	<b>6</b>
Prioridad de los requerimientos	6
Requerimientos funcionales	7
Requerimientos no funcionales	8
<b>V. Metodología de trabajo</b>	<b>9</b>
Scrum	9
Mock ups	9
<b>VI. Modelo de contexto</b>	<b>12</b>
<b>VII. Casos de uso</b>	<b>14</b>
Diagrama de casos de uso	14
Descripción de casos de uso	14
<b>VIII. Sistema</b>	<b>1</b>
Diagrama de clases	1
<b>IX. Modelo de procesos de negocio</b>	<b>1</b>
Sistema En General	1
Aregar, Modificar o Eliminar Eventos	1
Inscripción De Eventos	1
Asistencia	1
<b>X. Herramientas y tecnologías</b>	<b>1</b>
<b>XI. Repositorio</b>	<b>1</b>
<b>XII. Alcance</b>	<b>1</b>
Dentro del alcance	1
Fuera del alcance	1
<b>XIII. Diagramas interacción</b>	<b>1</b>
<b>XIV. Modelado de datos y sus herramientas</b>	<b>1</b>
<b>XV. Conclusión</b>	<b>1</b>



## I. Introducción

El Departamento de Deportes y Recreación dependiente de la Ilustre Municipalidad de Arica, es una entidad clave para la promoción del deporte y la actividad física en la comunidad local. Su misión principal es diseñar, implementar y gestionar programas y proyectos que fomenten la participación ciudadana en actividades deportivas y recreativas. El departamento opera en diversas áreas para cumplir sus objetivos, como la creación de eventos deportivos, gestión de recintos, programas comunitarios y fomento del deporte de alto rendimiento.

Para facilitar mejor la organización y distribución de eventos o programas, se pide la digitalización de los Recintos Deportivos. Este proyecto, tiene como objetivo crear un sistema de gestión y organización de eventos deportivos, facilitando el proceso.



## II. Definición del problema

### Problema

El proceso actual de gestión de eventos del Departamento de Deportes y Recreación presenta deficiencias que afectan tanto la eficiencia operativa como la experiencia de los usuarios. Las solicitudes de actividades se realizan de forma manual y sin un sistema de control centralizado, lo que provoca demoras significativas, falta de trazabilidad y dificultad para monitorear el avance de cada actividad.

Asimismo, el departamento carece de herramientas que permitan registrar la asistencia, identificar participantes y evaluar de manera sistemática el desarrollo de los eventos. Esta carencia limita la capacidad de análisis y la toma de decisiones informadas.

A ello se suma la ausencia de un mecanismo de notificación inmediata, que impide comunicar con rapidez cambios en fechas, horarios u otras características, generando desinformación, confusión y una disminución en la participación.

Estas limitaciones evidencian la necesidad de una solución tecnológica integral que optimice la gestión de eventos, automatice los procesos críticos y mejore la comunicación con los usuarios.

### Solución

Se propone el diseño e implementación de un sistema que gestione la información de los recintos y eventos, una aplicación móvil para que los usuarios de este sistema puedan ver los eventos y registrar los eventos que desea ir.

Esta plataforma permitirá:

- Gestión centralizada de eventos y recintos: creación, modificación, cancelación y monitoreo en tiempo real de actividades y espacios.
- Automatización del registro y control de asistencia, permitiendo al personal encargado verificar la participación de manera ágil y generar estadísticas de uso.
- Acceso móvil e interfaz intuitiva, garantizando facilidad de uso y disponibilidad de información actualizada desde cualquier dispositivo.



## III. Objetivos

### Objetivo General

Desarrollar un sistema de gestión y seguimiento de eventos recreativos para el Departamento de Deporte y Recreación de la Municipalidad de Arica, que permita digitalizar la administración de recintos y actividades, optimizar el proceso de inscripción y control de participantes, y mejorar la comunicación y análisis estadístico para la toma de decisiones.

### Objetivos Específicos

- Diseñar e implementar un módulo de gestión de eventos y recintos que permita al administrador crear, modificar o eliminar actividades y asignar encargados responsables.
- Desarrollar un sistema de inscripción en línea para los usuarios, accesible desde aplicación móvil, que simplifique la participación en talleres, canchas y programas recreativos.
- Incorporar funcionalidades para los encargados de eventos, permitiendo aprobar inscripciones, registrar asistencia y finalizar actividades de forma controlada.
- Implementar un módulo de reportes y estadísticas que entregue información sobre el uso de recintos, niveles de asistencia y participación histórica.



## IV. Requerimientos del sistema

Los requerimientos de un proyecto de software son los elementos fundamentales para el buen entendimiento entre los interesados del proyecto y el equipo encargado de desarrollar el proyecto. Una buena toma de requerimientos se traduce en una visión ordenada y estructurada de las funcionalidades que el sistema tiene que brindar.

### Prioridad de los requerimientos

Naturalmente en un proyecto el tiempo de desarrollo es limitado, por lo que es esencial generar un esquema de importancia para priorizar los requerimientos funcionales que más importantes sean.

Grado de prioridad	Descripción
Baja	Requerimiento con impacto mínimo en la cobertura de las funcionalidades. Puede implementarse al final del proyecto sin afectar de forma significativa el resultado global.
Medio	Requerimiento con impacto moderado en el funcionamiento. Es posible postergar para una entrega futura sin comprometer la operación principal del sistema.
Alto	Requerimiento que influye de manera importante en el desempeño del sistema, pero cuya ausencia no impide que las funciones esenciales sigan operando.
Esencial	Requerimiento crítico. Su ausencia imposibilita que el sistema cumpla el propósito para el cual fue diseñado.



## Requerimientos funcionales

ID	Requerimiento	Prioridad
RF00	El sistema deberá permitir al administrador gestionar (agregar, modificar o eliminar) un evento/actividad.	Esencial
RF01	El sistema deberá permitir al administrador gestionar (agregar, modificar o eliminar) un recinto donde se realizan actividades.	Esencial
RF02	El sistema deberá mostrar al administrador estadísticas respecto al uso de las canchas o recintos.	Alto
RF03	El sistema deberá permitir al administrador listar todos los eventos/actividades.	Alto
RF04	El sistema deberá permitir al administrador filtrar por alguna característica los eventos/actividades.	Medio
RF10	El sistema deberá permitir al usuario inscribirse a alguna actividad o evento disponible.	Esencial
RF11	El sistema deberá notificar a los participantes de un evento, si algún aspecto de este se modifica	Medio
RF12	El sistema deberá listar todos los eventos por orden de ocurrencia.	Bajo
RF13	El sistema deberá permitir al usuario eliminar su inscripción a alguna actividad o evento mientras esté disponible.	Bajo
RF14	El sistema deberá permitir al usuario ver listado a las actividades o talleres que ha asistido	Medio
RF15	El sistema deberá permitir al usuario crear un perfil con sus datos personales.	Alto
RF16	El sistema deberá permitir al usuario registrarse en el sistema.	Medio
RF20	El sistema deberá permitir al encargado del evento ver los eventos que va a realizar.	Esencial
RF21	El sistema deberá permitir al encargado ver los eventos que ha realizado.	Medio
RF22	El sistema deberá permitir al encargado marcar como presente a los asistentes del evento.	Esencial
RF23	El sistema deberá permitir al encargado registrarse en el sistema.	Esencial
RF24	El sistema deberá permitir al encargado terminar el evento cuando esto suceda.	Medio



## Requerimientos no funcionales

ID	Requerimiento
<b>RNF0</b>	El sistema debe soportar alto tráfico por peticiones para reservar o asistencia.
<b>RNF1</b>	El sistema de inscripción debe ser una aplicación móvil.
<b>RNF2</b>	El sistema debe entregar la última información que se haya modificado.
<b>RNF3</b>	El sistema debe ser fácil de entender y de fácil acceso.
<b>RNF4</b>	El sistema debe responder de manera fluida y correcta al realizar peticiones o enviar respuestas.
<b>RNF5</b>	El sistema debe notificar correctamente a todos los participantes de alguna modificación de un o varios eventos.
<b>RNF6</b>	El sistema debe ser capaz de desplegar el servidor en otros sistemas.



## V. Metodología de trabajo

### Scrum

Scrum es un proceso empírico y liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. Se basa en los principios de la Metodología Ágil (Agile).

Fomenta el trabajo colaborativo, la autoorganización del equipo y la entrega de productos en ciclos cortos e iterativos de tiempo fijo, conocidos como Sprints (generalmente de una a cuatro semanas).

El objetivo es que permita a los equipos responder de manera flexible e inmediata a los cambios y obtener retroalimentación constante del cliente, asegurando la mejora continua y la entrega de valor de forma incremental.

### Mock ups

#### Listar eventos



## Gestionar eventos

Crear evento

Titulo

Descripción

Recinto -- Seleccion recinto

Capacidad maxima 00

Subir foto

Guardar Salir

## Ver eventos

Eventos disponibles

Plan masiva de natacion

Piscina olímpica

15/09/20 - 13:00

Campeonato voleibol

Epicentro 1

14/09/20 - 13:00



## Inscribir eventos

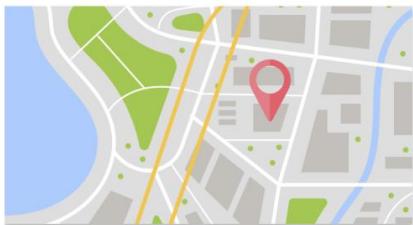


### Plan masiva de natacion



#### DESCRIPCION DEL EVENTO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce eu tempus metus. Vestibulum commodo sapien in nulla rutrum, quis cursus turpis pharetra. Nullam nulla diam, congue vestibulum ornare et, iaculis nec lorem. Phasellus placerat nisi commodo, fringilla arcu nec, molestie arcu. Donec vulputate justo quam, sed pharetra ex vestibulum id. Aenean et urna mi. Nam ullamcorper diam at tincidunt.

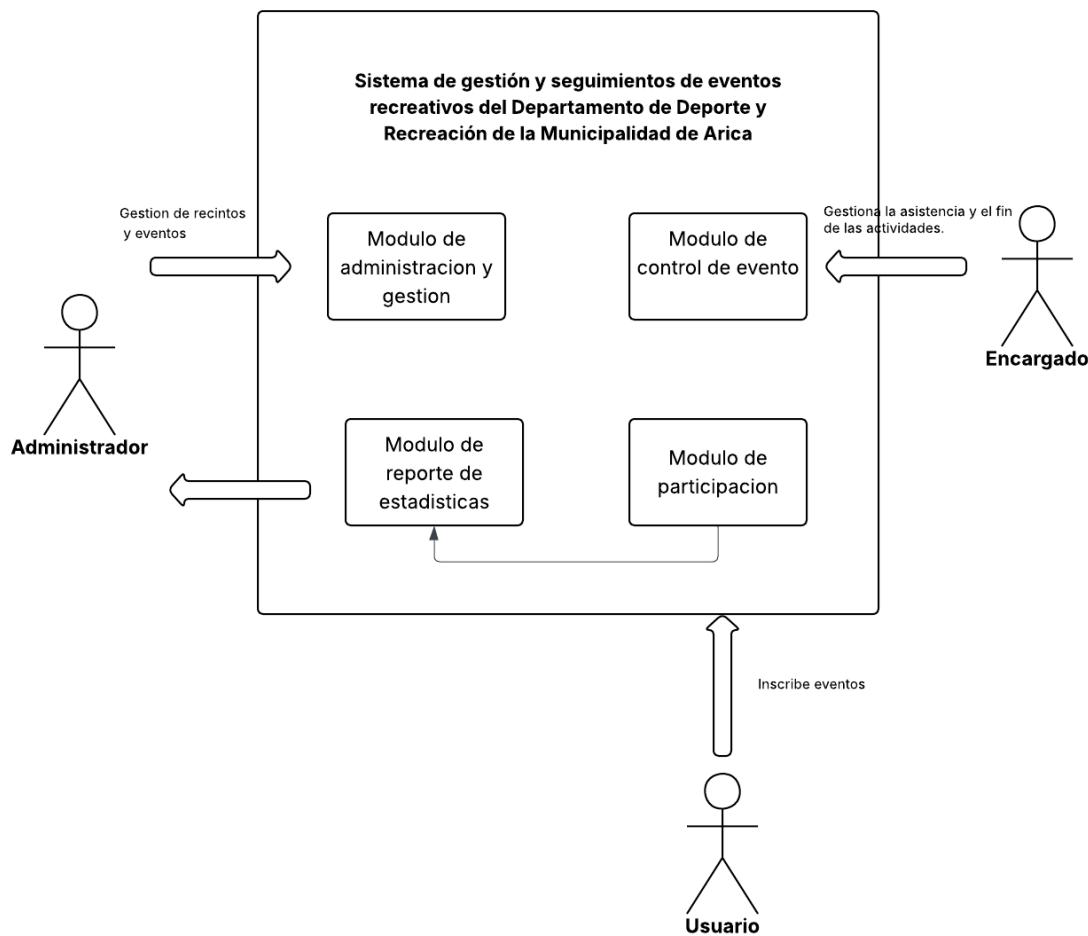


Encargado

**Inscribir**



## VI. Modelo de contexto



El sistema está conceptualmente dividido en cuatro módulos funcionales para gestionar las distintas responsabilidades del proyecto, de acuerdo con los objetivos específicos:

### A. Módulo de Administración y Gestión

Este subsistema constituye el funcionamiento esencial del sistema y está diseñado para interactuar principalmente con el Administrador del Departamento. Su función primordial es recibir la Gestión de Eventos y Recintos, que comprende los datos necesarios para crear, modificar o eliminar actividades y para digitalizar la administración de los recintos deportivos.

### B. Módulo de Interacción Móvil y Participación

Este módulo actúa como la interfaz directa para el Usuario/Participante, dado que el sistema de inscripción debe ser una aplicación móvil. El flujo de entrada más significativo es la Solicitud de Inscripción, la cual permite a los usuarios asegurar su participación en talleres y programas recreativos. Como flujo de salida, el módulo proporciona la Información de Eventos, incluyendo listados y las notificaciones por cambios en el evento.



### **C. Módulo de Control de Evento**

Este subsistema es la herramienta operativa para el Encargado del Evento, permitiéndole ejecutar las actividades de forma controlada. El flujo de entrada crítico que gestiona es el Registro de Asistencia, mediante el cual el Encargado marca la presencia de los asistentes y señala la finalización de las actividades.

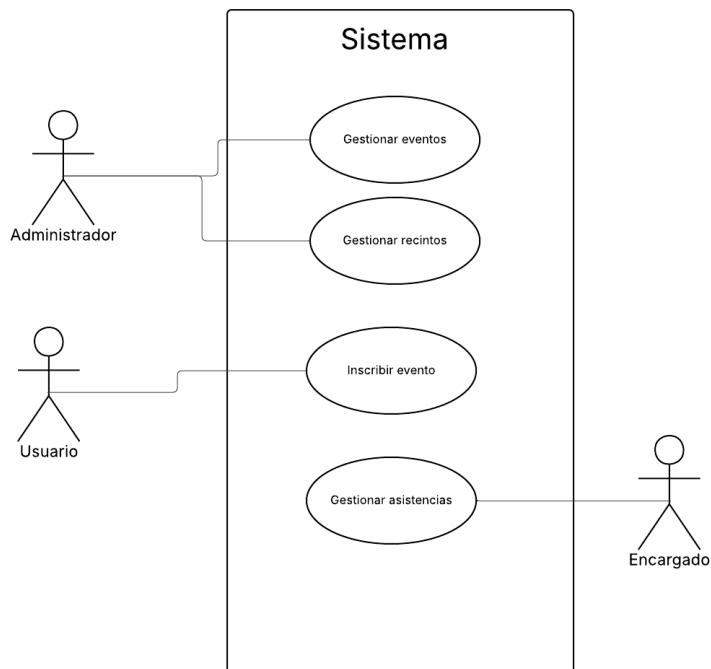
### **D. Módulo de Reportes y Estadísticas**

La función de este módulo es procesar la totalidad de la data histórica de inscripciones, asistencias y uso de recintos generada por los otros módulos. El flujo de salida principal son los Reportes y Estadísticas, los cuales son proporcionados al Administrador del Departamento para entregar información analítica que apoye la toma de decisiones.



## VII. Casos de uso

### Diagrama de casos de uso



### Descripción de casos de uso

#### Listar Eventos

<b>Casos de uso:</b> Listar eventos	
<b>Autor:</b> Daniel Alday	<b>Fecha:</b> 8/10/25
<b>Actor:</b> Administrador	
<b>Precondición:</b> El sistema debe tener eventos disponibles guardados.	
<b>Flujo Crear</b>	



<b>Administrador</b>	<b>Sistema</b>
<p>1. El administrador solicita listar los eventos disponibles.</p>	<p>2. El sistema muestra todos los eventos disponibles por orden de ocurrencia.</p> <p>Cada evento muestra:</p> <ul style="list-style-type: none"> <li>- Nombre de evento</li> <li>- Fecha/s y hora/ del/los evento/s</li> <li>- Foto del evento</li> </ul>
<b>Postcondición:</b> –NO APLICA–	

### Gestionar eventos

**Casos de uso:** Gestionar eventos

<b>Autor:</b> Daniel Alday	<b>Fecha:</b> 8/10/25
----------------------------	-----------------------

**Actor:** Administrador

**Precondición:** –NO APLICA–

#### Flujo Crear

<b>Administrador</b>	<b>Sistema</b>
<p>1. El administrador ingresa información del evento como:</p> <ul style="list-style-type: none"> <li>- Título de evento</li> <li>- Descripción de evento</li> <li>- Horario/s del evento/s</li> <li>- Lugar del evento</li> <li>- Encargado del evento</li> </ul>	<p>2. El sistema guarda la información del evento.</p> <p>3. El sistema notifica al administrador que el evento fue guardado.</p>

#### Flujo Editar



<p><b>Administrador</b></p> <p>2. Selecciona la información del evento que desea modificar.</p> <p>3. Ingresa la modificación al campo correspondiente.</p>	<p><b>Sistema</b></p> <p>1. <b>include “Listar eventos”</b></p> <p>4. El sistema notifica a los asistentes los cambios realizados.</p>
<b>Flujo Editar</b>	
<p><b>Administrador</b></p> <p>2. Selecciona el evento que desea eliminar.</p>	<p><b>Sistema</b></p> <p>1. <b>include “Listar eventos”</b></p> <p>3. El sistema notifica la eliminación del evento a los asistentes.</p>

### Gestionar recintos

**Casos de uso:** Gestionar recintos

<b>Autor:</b> Daniel Alday	<b>Fecha:</b> 8/10/25
----------------------------	-----------------------

**Actor:** Administrador

**Precondición:** **–NO APLICA–**

**Flujo Crear**



<p><b>Administrador</b></p> <p>1. El administrador ingresa información del recinto como:</p> <ul style="list-style-type: none"> <li>- Nombre del recinto</li> <li>- Descripción del recinto</li> <li>- Ubicación</li> <li>- Capacidad</li> <li>- Horario</li> </ul>	<p><b>Sistema</b></p> <p>2. El sistema guarda la información del recinto.</p> <p>3. El sistema notifica al administrador que el recinto fue guardado.</p>
<b>Flujo Editar</b>	
<p><b>Administrador</b></p> <p>2. Selecciona la información del recinto que desea modificar.</p> <p>3. Ingresa la modificación al campo correspondiente.</p>	<p><b>Sistema</b></p> <p>1. Lista todos los recintos, por nombre, descripción y capacidad.</p> <p>4. El sistema guarda los cambios.</p>
<b>Flujo Editar</b>	
<p><b>Administrador</b></p> <p>2. Selecciona el recinto que desea eliminar.</p>	<p><b>Sistema</b></p> <p>1. Lista todos los recintos, por nombre, descripción y capacidad.</p> <p>3. El sistema guarda elimina el recinto seleccionado.</p>

### Inscribir evento

**Casos de uso:** Inscribir evento

<b>Autor:</b> Daniel Alday	<b>Fecha:</b> 18/10/25
----------------------------	------------------------

**Actor:** Usuario



**Precondición: Deben haber eventos guardados en el sistema.**

**Flujo normal**

<b>Usuario</b>	<b>Sistema</b>
<p>2. El usuario elige el evento al que quiere asistir.</p>	<p>1. El sistema lista todos los eventos disponibles, por evento:</p> <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Descripción</li> <li>- Cupo</li> <li>- Fecha / hora</li> <li>- Ubicación.</li> </ul> <p>3. El sistema guarda la información de la inscripción.</p>

**Flujo sin cupo**

<b>Usuario</b>	<b>Sistema</b>
<p>2. El usuario elige el evento al que quiere asistir</p>	<p>1. El sistema lista todos los eventos disponibles, por evento:</p> <ul style="list-style-type: none"> <li>- Nombre</li> <li>- Descripción</li> <li>- Capacidad</li> <li>- Fecha / hora</li> <li>- Ubicación.</li> </ul> <p>4. El sistema notifica al usuario que no queda cupo para ese evento.</p>



## Marcar asistencia

**Casos de uso:** Inscribir evento

<b>Autor:</b> Daniel Alday	<b>Fecha:</b> 18/10/25
----------------------------	------------------------

<b>Actor:</b> Encargado
-------------------------

**Precondición:** El encargado debe tener eventos asignados.

### Flujo normal

#### Encargado

2. El encargado marca por cada usuario, si está presente o no.

#### Sistema

1. El sistema lista todos los participantes al evento, por cada participante:

- Nombre completo
- Correo
- Número telefónico
- Fecha de inscripción

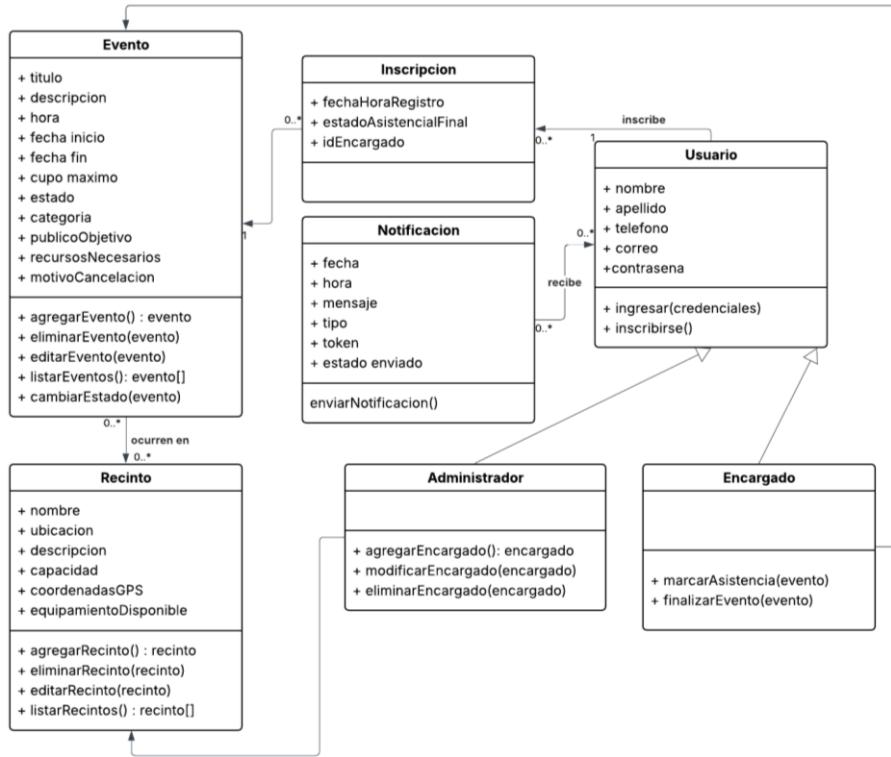
3. El sistema guarda la información de la inscripción.

**Postcondición:** La asistencia al evento queda guardada.



## VIII. Sistema

### Diagrama de clases



El diagrama se organiza en torno a la gestión de eventos y la participación de los usuarios, con una estructura de herencia para los roles y clases de asociación para el seguimiento.

#### 1. Entidades Fundamentales (Catálogo)

**Evento:** Clase central que almacena los detalles de la actividad (título, fechas, cupo máximo). Incluye atributos para la segmentación estadística (categoria, publicoObjetivo). Sus operaciones permiten la gestión completa (CRUD) por parte del Administrador.

**Recinto:** Digitaliza la información de las sedes físicas, incluyendo coordenadas GPS, capacidad, y ubicación.

#### 2. Gestión de Usuarios y Roles

**Usuario (Clase Base):** Contiene la información de registro básica (nombre, correo). Incluye la operación `inscribirse()` para iniciar la participación.

**Administrador:** Hereda de Usuario. Sus operaciones se centran en la gestión de personal (`agregarEncargado`, `eliminarEncargado`).

**Encargado:** Hereda de Usuario. Realiza las funciones operativas críticas, como `marcarAsistencia(evento)` y `finalizarEvento(evento)`.



### 3. Trazabilidad y Comunicación

**Inscripción:** Clase de asociación que vincula a un Usuario con un Evento. Contiene datos clave para las estadísticas como la fecha y hora de registro y el estado asistencial final ('Presente'/'Ausente').

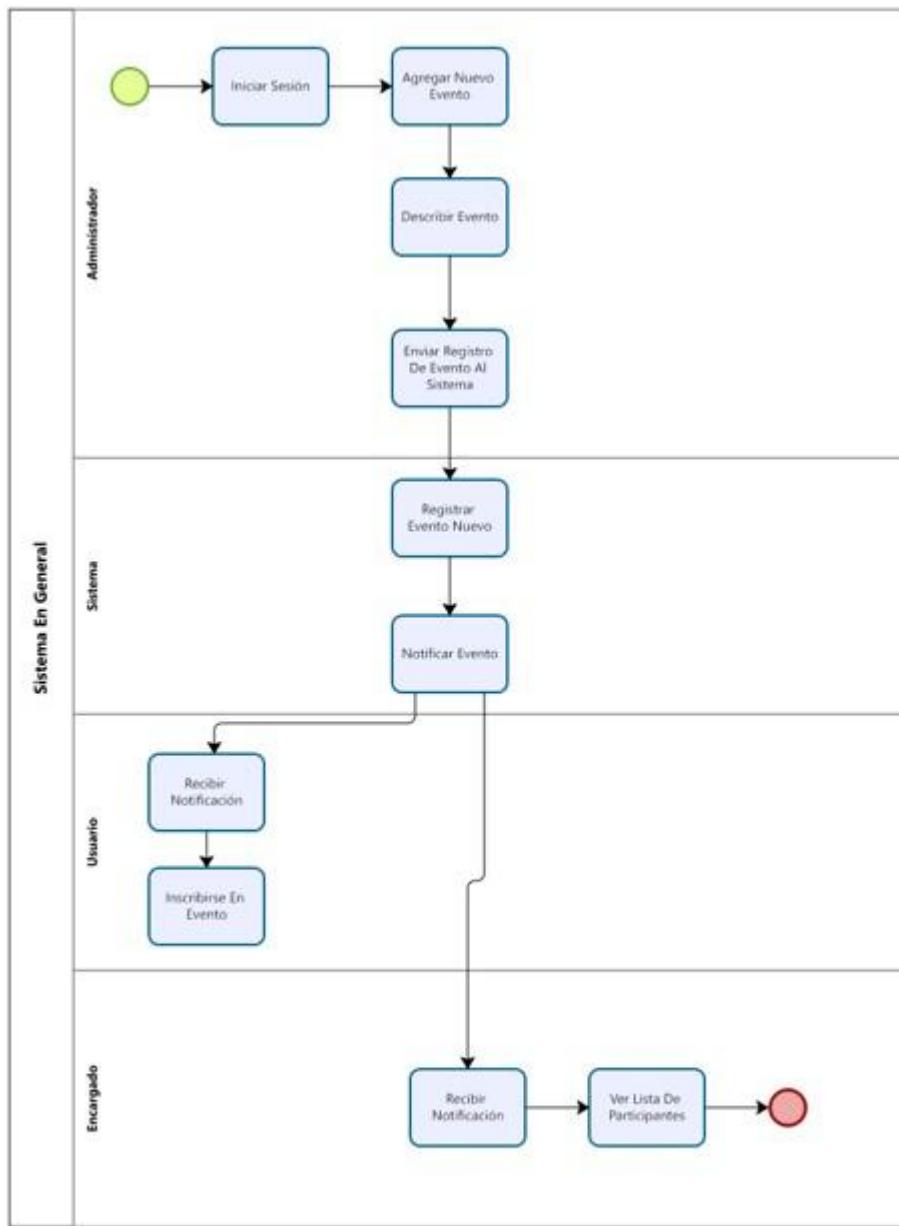
**Notificación:** Gestiona el sistema de alerta. Almacena el mensaje y el tipo, y se asocia al Usuario para cumplir con el requerimiento de comunicación inmediata.



## IX. Modelo de procesos de negocio

En las siguientes ilustraciones se presentará el modelo de proceso de negocio que encapsula los procesos más importantes para el desarrollo del sistema propuesto. Esta propuesta de BPM captura el proceso original de inscripción y gestión de recintos y le da una

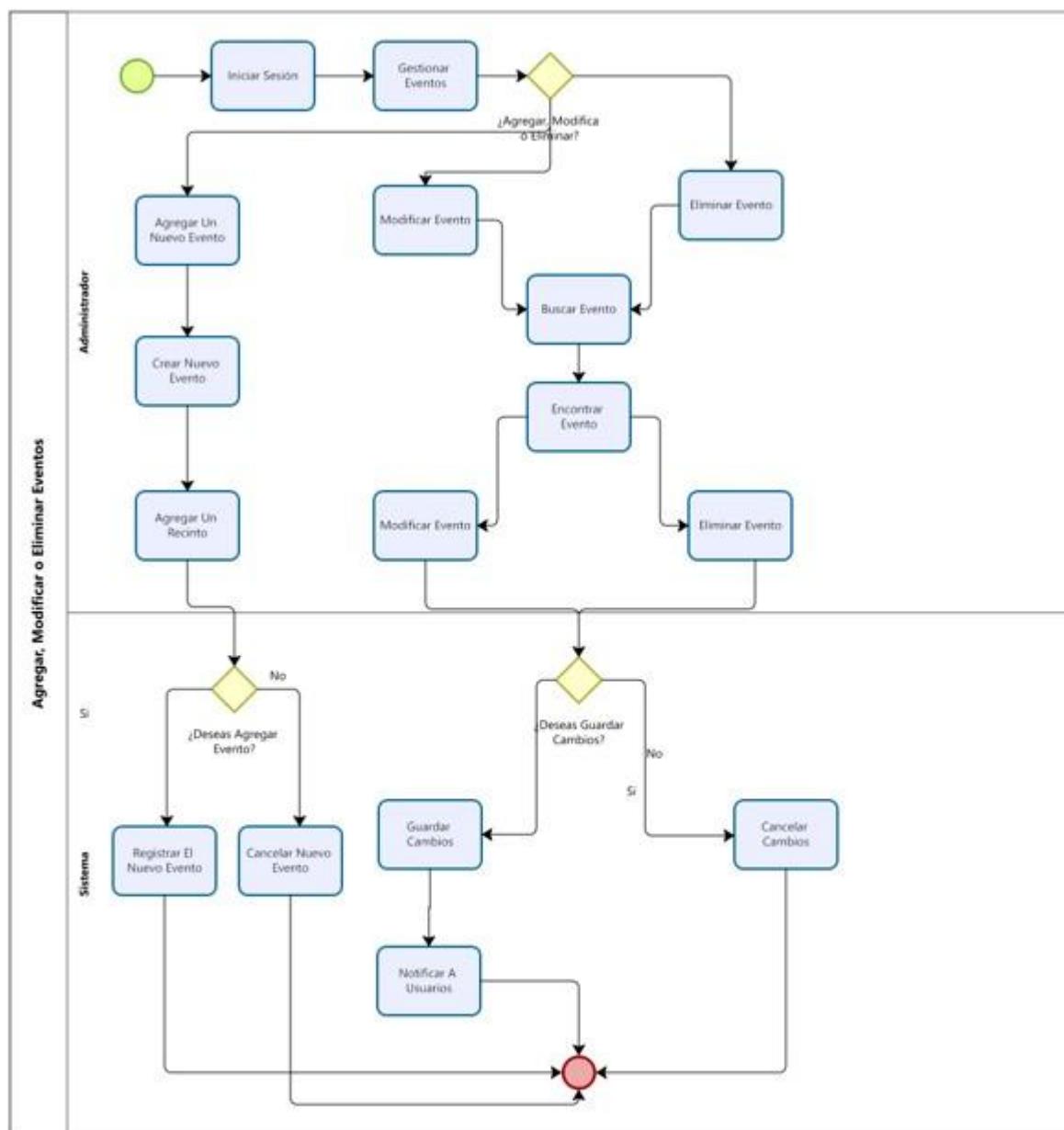
### Sistema En General



Powered by  
 Isagen Modeler

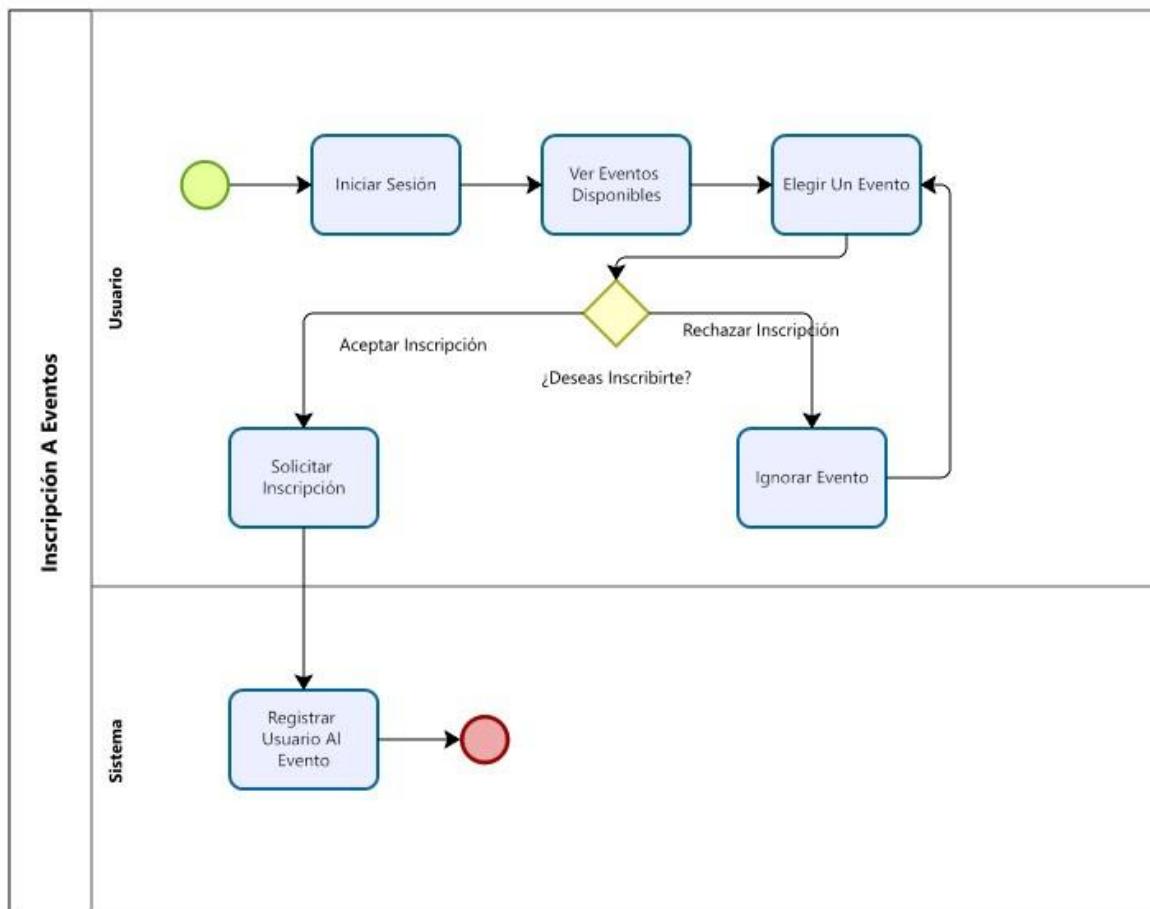


## Agregar, Modificar o Eliminar Eventos





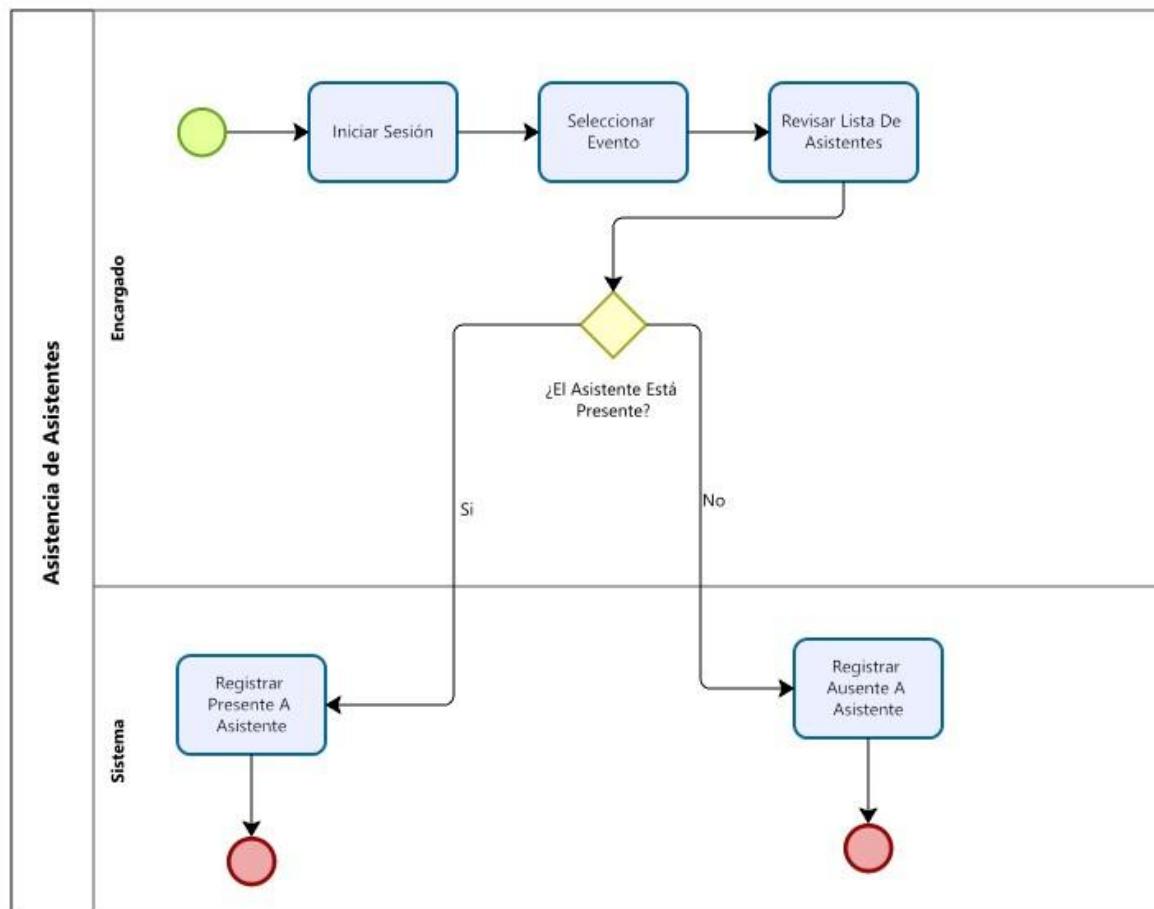
## Inscripción De Eventos



Powered by  
 **Modeler**



## Asistencia



Powered by  
bizagi  
Modeler



## X. Herramientas y tecnologías

Para poder implementar este proyecto se escogieron una serie de herramientas y tecnologías que ahora serán descritas y explicadas, su justificación así como también su alcance y experiencias previas con estas mismas.

Tecnología	<b>Angular (Frontend)</b>
Descripción	Este framework es una tecnología que permite implementar un frontend a través de un marco de trabajo robusto y basado en componentes para construir interfaces de usuario (UI) web modernas y de una sola página (SPA). Utiliza TypeScript, lo que añade tipado estático y mejora la mantenibilidad del código en proyectos grandes.
Justificación	Es la elección ideal para construir el panel de administración del sistema. Su arquitectura basada en componentes permite crear una interfaz compleja pero fácil de mantener, perfecta para que los administradores gestionen recintos, eventos, y encargados. El robusto sistema de formularios de Angular facilitará la captura de datos de reservas y la configuración de los recintos.

Tecnología	<b>Spring Boot (Backend)</b>
Descripción	Este framework es una tecnología que permite implementar un backend de forma rápida y eficiente. Es un marco de trabajo basado en Java que simplifica enormemente la creación de API RESTful robustas, seguras y de alto rendimiento.
Justificación	Esta parte del sistema es la que se encarga de implementar la reglas del negocio. Proporcionará la lógica de negocio y la API REST que consumirá tanto el panel de Angular como la app móvil. Es excelente para manejar reglas de negocio complejas.

Tecnología	<b>Kotlin Multiplatform (App móvil)</b>
Descripción	Este framework es una tecnología que permite implementar aplicaciones móviles nativas (iOS y Android) compartiendo código. No es un backend, sino un SDK que permite escribir la lógica de negocio (como conectarse a la API de Spring Boot, manejar los datos de una reserva, etc.) una sola vez en lenguaje Kotlin, y que esa lógica funcione tanto en Android como en iOS.
Justificación	Esta es la herramienta para los clientes (usuario que reserva). Les permitirá reservar eventos y ver horarios. Al usar Kotlin Multiplatform, ahorras tiempo de desarrollo, ya que solo programas la lógica de la app una vez. Esto asegura que la app de Android y la de iOS tengan exactamente las mismas reglas y funcionen igual, evitando inconsistencias.



Tecnología	<b>Docker (contenedores)</b>
Descripción	Una plataforma de virtualización a nivel de sistema operativo que empaqueta aplicaciones y todas sus dependencias (código, librerías, etc.) en unidades aisladas llamadas "contenedores".
Justificación	Se usará para "empaquetar" el backend de Spring Boot y probablemente la base de datos. Esto garantiza que el entorno de desarrollo sea idéntico al de producción, eliminando el clásico problema de "en mi máquina sí funciona". Facilita enormemente el despliegue, la escalabilidad y la gestión de los servicios en el servidor.

## XI. Repositorio

Con el objetivo de gestionar eficientemente el código fuente del proyecto, facilitar el trabajo colaborativo y mantener un control de versiones riguroso, todo el desarrollo del "Sistema de gestión y seguimientos de eventos recreativos" se aloja en un repositorio centralizado en la plataforma GitHub. Esta herramienta permitirá la trazabilidad de todos los cambios aplicados al *software* y servirá como respaldo principal del proyecto.



The screenshot shows a GitHub repository named "Proyectos4-Recintos" which is private. The repository has 1 branch and 0 tags. The commit history shows the following activity:

- DAHonor34 · Se Agrego Parte Del Fortnend · abf3a24 · last month · 3 Commits
- backend: creacion de carpetas del proyecto · last month
- frontend: Se Agrego Parte Del Fortnend · last month
- kotlin-app: creacion de carpetas del proyecto · last month
- LICENSE: Initial commit · last month
- README.md: Initial commit · last month

The repository description is as follows:

**Proyectos4-Recintos**

Este repositorio almacena infomacion del proyecto de gestion de recintos de la municipalidad, aca se almacena, informacion del proyecto y codigo del proyecto.



## XII. Alcance

El alcance de un proyecto principalmente define el marco respecto a las funcionalidades para desarrollar este mismo, es por eso que definir un buen alcance nos permite estar concentrados en las funcionalidades principales y crear un sistema que realmente resuelva problemas que el cliente tiene.

### Dentro del alcance

El proyecto incluye el diseño, desarrollo e implementación de las siguientes funcionalidades clave:

- **Gestión de Contenidos (Administrador):**
  - Permitir al administrador crear, modificar y eliminar eventos/actividades.
  - Permitir al administrador crear, modificar y eliminar recintos donde se realizan las actividades.
  - Permitir al administrador asignar encargados responsables a los eventos.
- **Gestión de Participación (Usuario/Participante):**
  - Desarrollar una aplicación móvil para el sistema de inscripción.
  - Permitir a los usuarios registrarse en el sistema y crear un perfil con sus datos personales.
  - Permitir a los usuarios inscribirse a las actividades o eventos disponibles.
  - Permitir a los usuarios eliminar su inscripción a una actividad.
  - Enviar notificaciones a los participantes si algún aspecto de un evento se modifica.
- **Control Operativo (Encargado):**
  - Permitir al encargado registrar la asistencia (marcar como presente) de los participantes del evento.
  - Permitir al encargado ver los eventos que tiene asignados y finalizar el evento cuando este suceda.
- **Ánalysis y Reportes (Administrador):**
  - Implementar un módulo que muestre reportes y estadísticas sobre el uso de recintos y los niveles de asistencia.



## Fuera del alcance

Para mantener el proyecto enfocado y cumplir con los plazos, las siguientes funcionalidades **NO** se incluirán en este sistema:

- Gestión de Pagos: El sistema no procesará pagos, cobros de entradas, ni se integrará con pasarelas de pago. La inscripción se asume gratuita.
- Gestión de Recursos Humanos: El sistema no gestionará contratos, horarios laborales, ni el pago de sueldos del personal o de los encargados. Solo gestiona su *asignación* a un evento.
- Gestión de Inventario Físico: El sistema gestiona los *recintos*, pero no el inventario de equipamiento deportivo (balones, redes, colchonetas, etc.) asociado a esos recintos.
- Integración Contable: El sistema no se integrará con el *software* financiero o contable de la Municipalidad.
- Marketing y Redes Sociales: El sistema notificará a los usuarios inscritos, pero no incluirá módulos para la promoción activa de eventos en redes sociales externas.



## XIII. Diagramas interacción

Diagrama de secuencia “Listar Eventos”

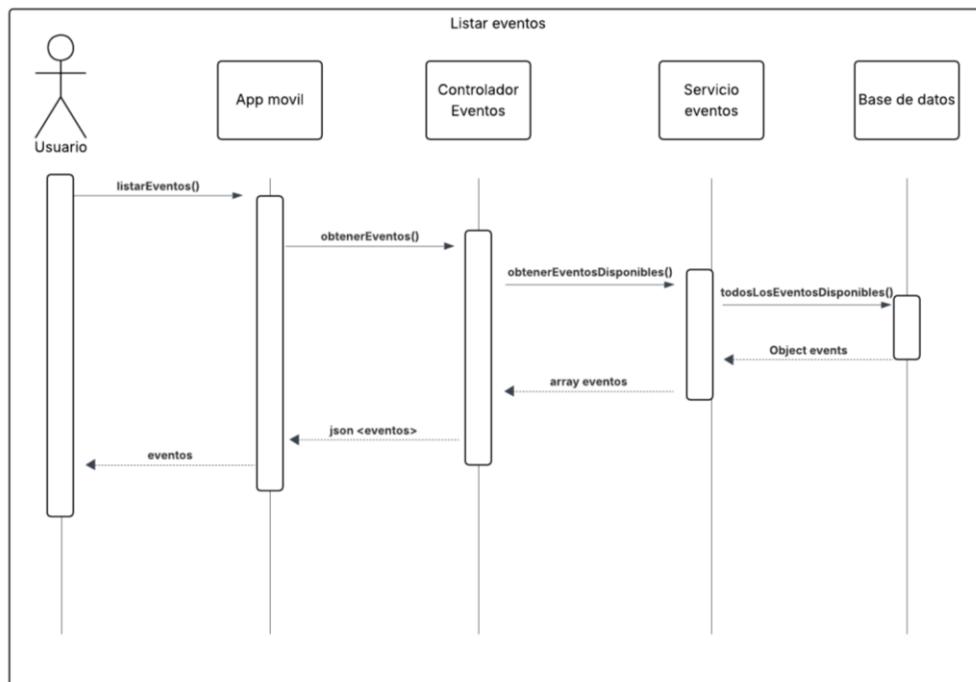
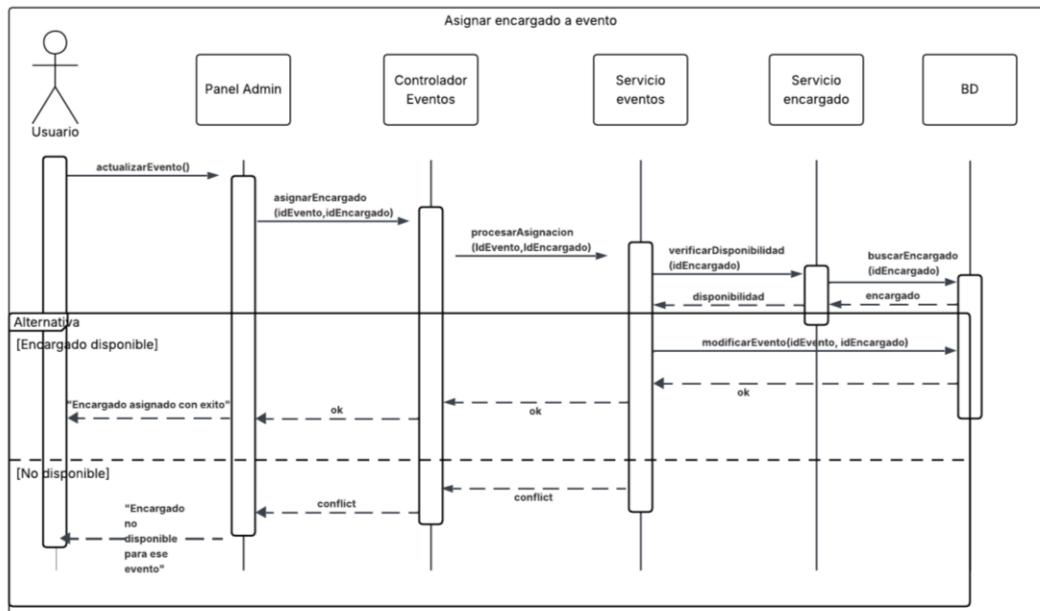
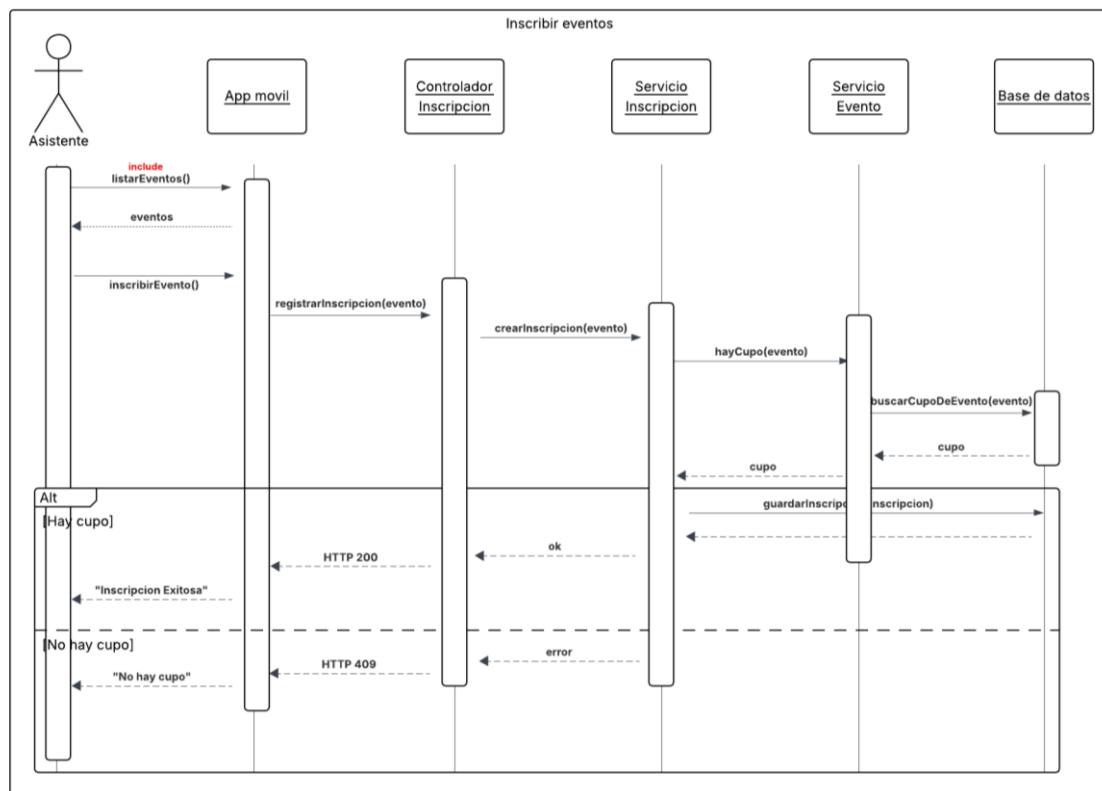


Diagrama de secuencia “Asignar encargado a evento”



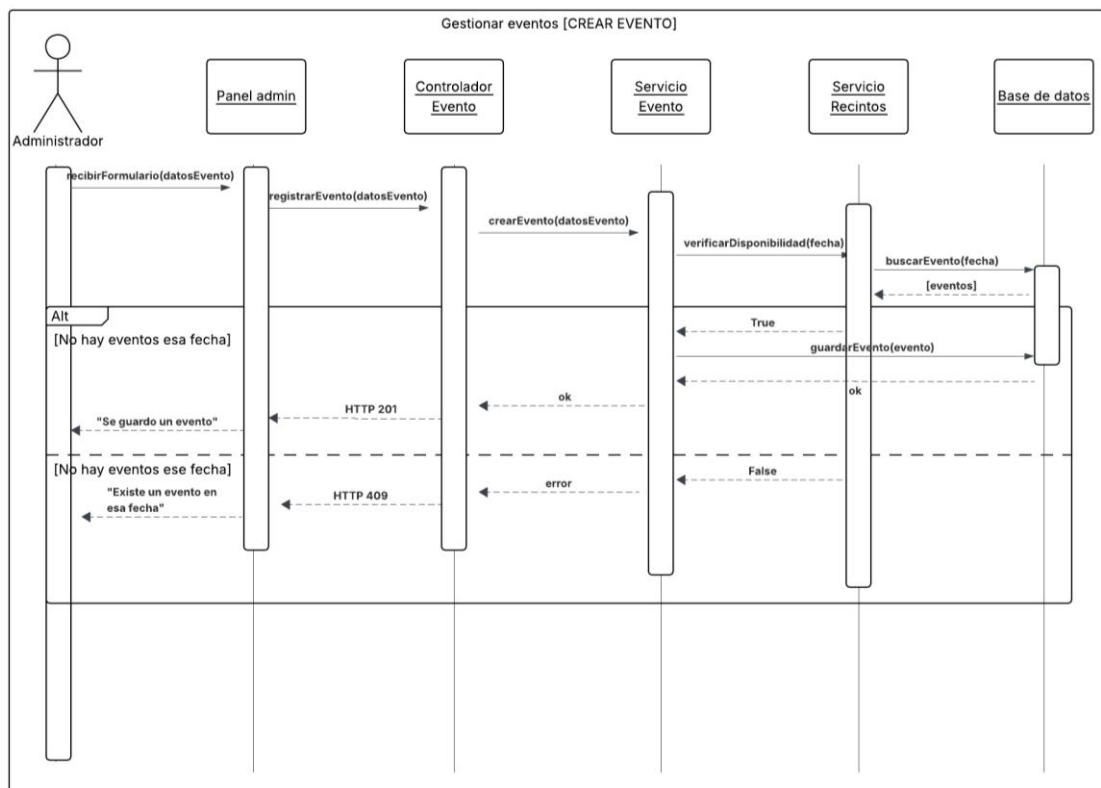


### Diagrama de secuencia “Inscribir evento”



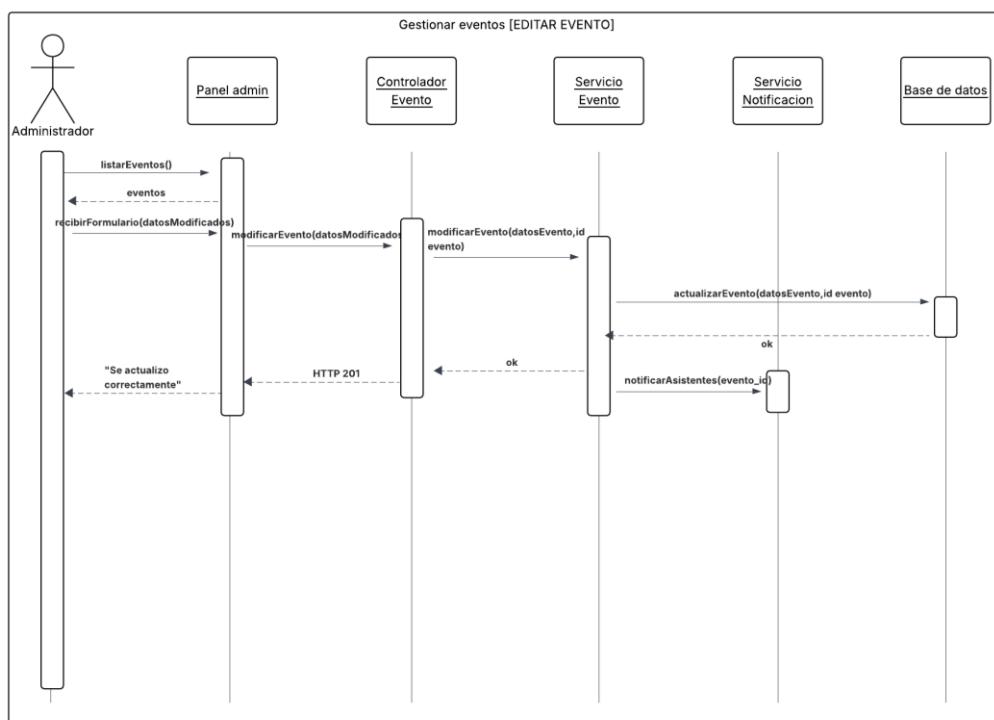
#

### Diagrama de secuencia “Crear Eventos”

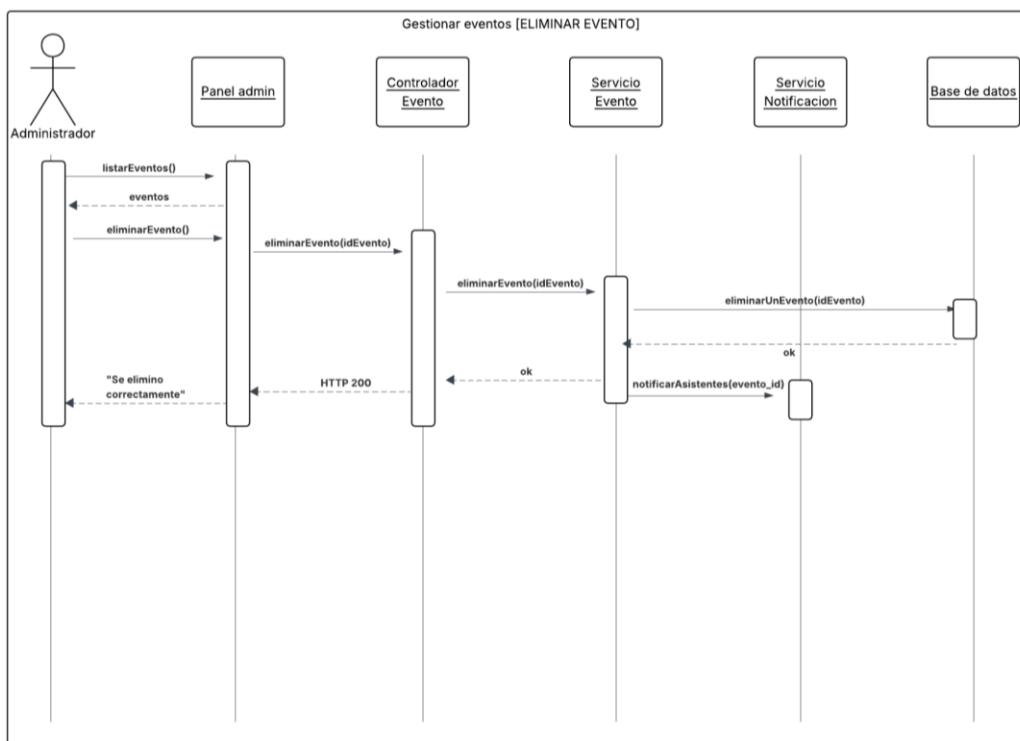




### Diagrama de secuencia “Editar Eventos”

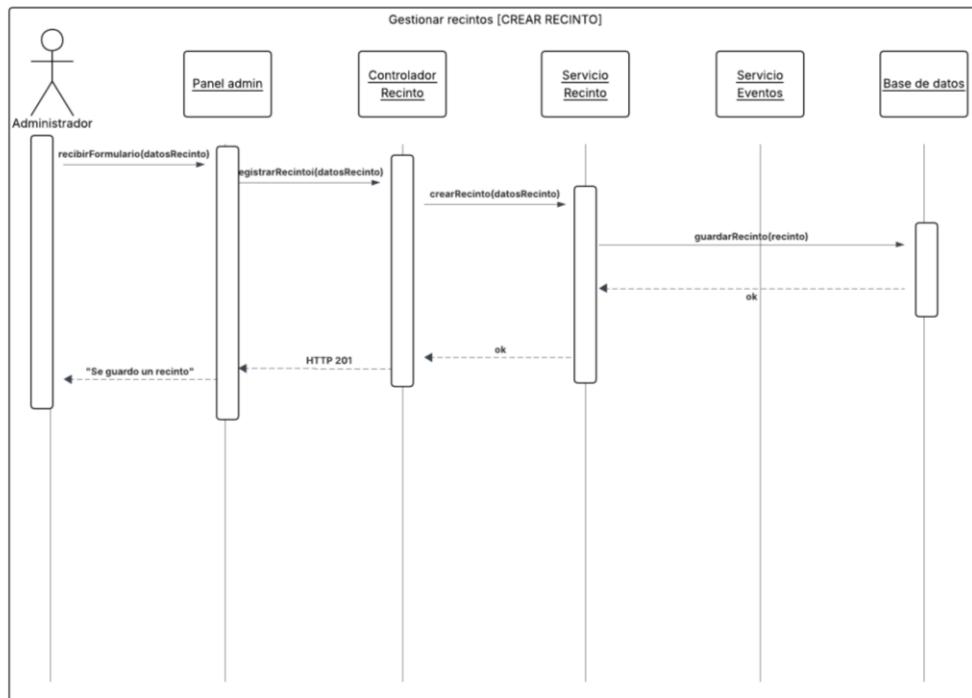


### Diagrama de secuencia “Eliminar Eventos”

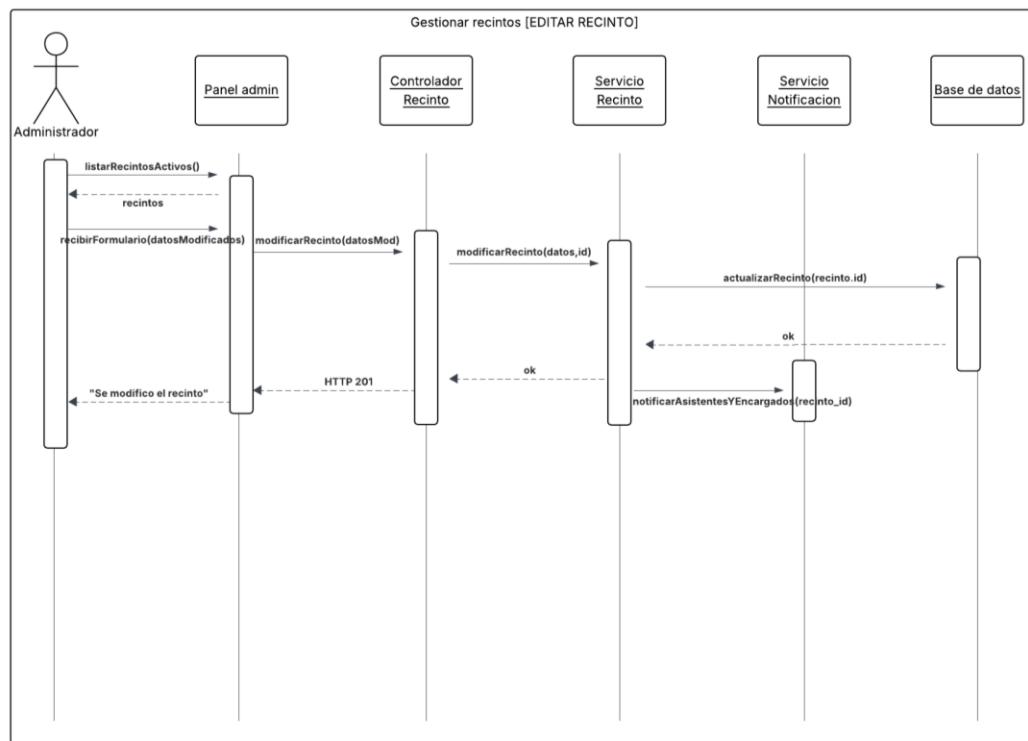




### Diagrama de secuencia “Crear recinto”



### Diagrama de secuencia “Editar recinto”



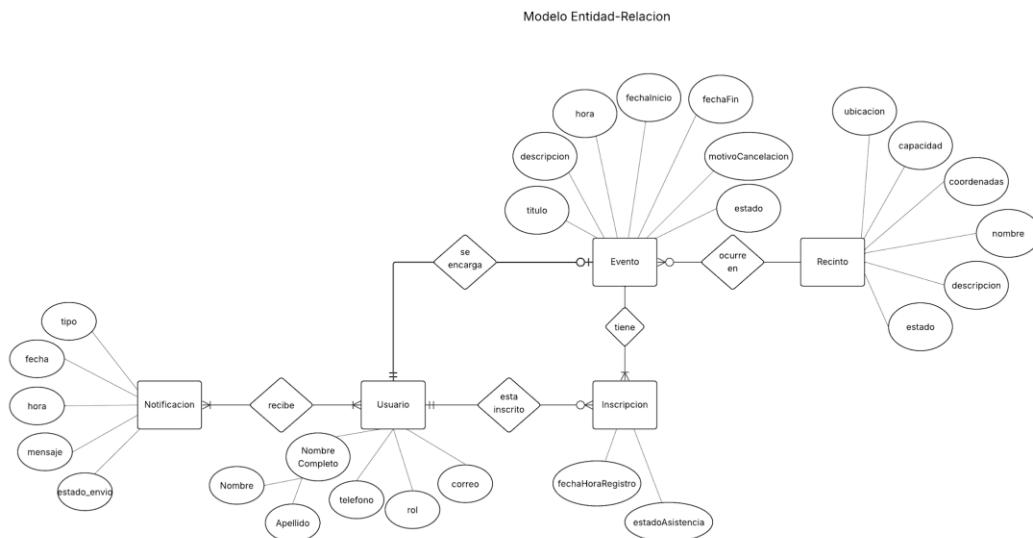


## XIV. Modelado de datos y sus herramientas

Para la capa de persistencia de datos del proyecto, se ha seleccionado como sistema gestor de base de datos **PostgreSQL**, y como framework de mapeo Objeto-Relacional (ORM), la **Java Persistence API (JPA)**, comúnmente implementada a través de Spring Data JPA. La elección de PostgreSQL se justifica por su reputación como una base de datos relacional de código abierto robusta, confiable y con un potente motor para ejecutar consultas complejas; esta capacidad es un requisito indispensable para el **módulo de reportes y estadísticas** del sistema.

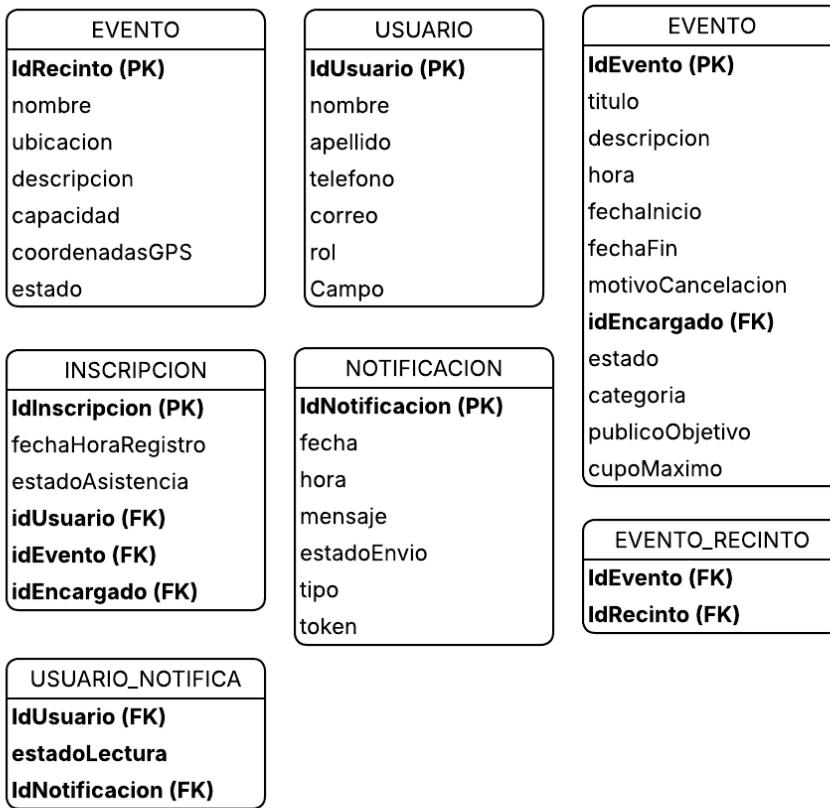
Por su parte, JPA se utilizará para abstraer la lógica de la base de datos, permitiendo al equipo de desarrollo interactuar con las tablas de PostgreSQL como si fueran objetos Java (basados en el diagrama de clases como Evento e Inscripción). Esta abstracción acelera el desarrollo, reduce significativamente el código SQL repetitivo (boilerplate) y mejora la mantenibilidad del sistema, permitiendo que el enfoque principal sea la lógica de negocio.

### Modelo Entidad-Relación





## Modelo relacional





## XV. Conclusión

La fase de diseño del "**Sistema de gestión y seguimientos de eventos recreativos**" ha culminado con la creación de un conjunto completo de artefactos de software que definen una arquitectura robusta, escalable y alineada con los objetivos del Departamento de Deporte y Recreación.

El alcance del proyecto se centró en digitalizar los procesos manuales, optimizar la inscripción móvil y proveer herramientas para la toma de decisiones. Para cumplir con esto, se seleccionó una arquitectura tecnológica moderna: PostgreSQL fue elegido por su capacidad para manejar las consultas complejas requeridas por el módulo de estadísticas, mientras que JPA (implementado vía Spring) se seleccionó para abstraer la lógica de la base de datos, acelerando el desarrollo y manteniendo la coherencia con el modelo de clases.

Para modelar la lógica de negocio, se optó por Diagramas de Secuencia por encima de los Diagramas de Colaboración, dada su superior claridad para representar el flujo cronológico de las interacciones y el manejo de flujos alternativos.

Estos diagramas de interacción sirvieron como base para definir el Modelo Entidad-Relación (MER) y su correspondiente Modelo Relacional (MR), asegurando que el esquema de la base de datos refleje fielmente la lógica de negocio validada. Finalmente, la implementación del proyecto se gestionará en un repositorio GitHub para garantizar la integridad, el control de versiones y la colaboración del equipo.

En conjunto, este diseño proporciona un plan de implementación completo que satisface los requerimientos esenciales del sistema, desde la gestión de eventos hasta el registro de asistencia y la comunicación con el usuario.