

Universidad Iberoamericana

Facultad de Ingeniería

Ingeniería de software.

**Proyecto de software**

Andres Felipe Pipicano Villa Código Banner 100149864

Unidad de Aprendizaje IIII

## Nota Importante

Buenas noches profesora, en este documento envió toda la documentación que se genero con el proyecto de software, ya que en las clases se menciono que si dejaba la nota para que supiera que de este documento dependo poder aprobar la materia ya que por circunstancias no pude entregar ninguna de las notas del corte numero 2 le solicito su colaboración que se mencionó en clase.

## Enlaces y accesos importantes del entregable

- OneDrive: Se encuentra la documentación correspondiente al entregable (Prototipo en PDF, Documentación del proyecto, Diagramas, Scripts SQL , Documentación de las pruebas unitarias de integración y pruebas automatizadas prototipo wirefre y demás documentación generada) [Click para abrir OneDrive](#)
- Link ejecutable de prototipo figma: [Click para abrir ejecutable prototipo figma](#)
- Enlace Repositorio GitHub: <https://github.com/AnDrEs-F15/Guarderia-Ibero>  
(También está la documentación del proyecto en la carpeta Documentación V2)  
El repositorio esta tanto el API en SpringBoot y el APP en React
- Video Mostrando prototipo [Ver Video](#)  
se hace referencia al entregable de análisis y diseño de software
- 
- 
- Enlace aplicación desplegada: <https://guarderia-ibero.onrender.com>
- Enlace Video mostrando aplicación: [Click ver video](#)

## Credenciales Para ingresar a la página administradora:

**User:** IberoUser

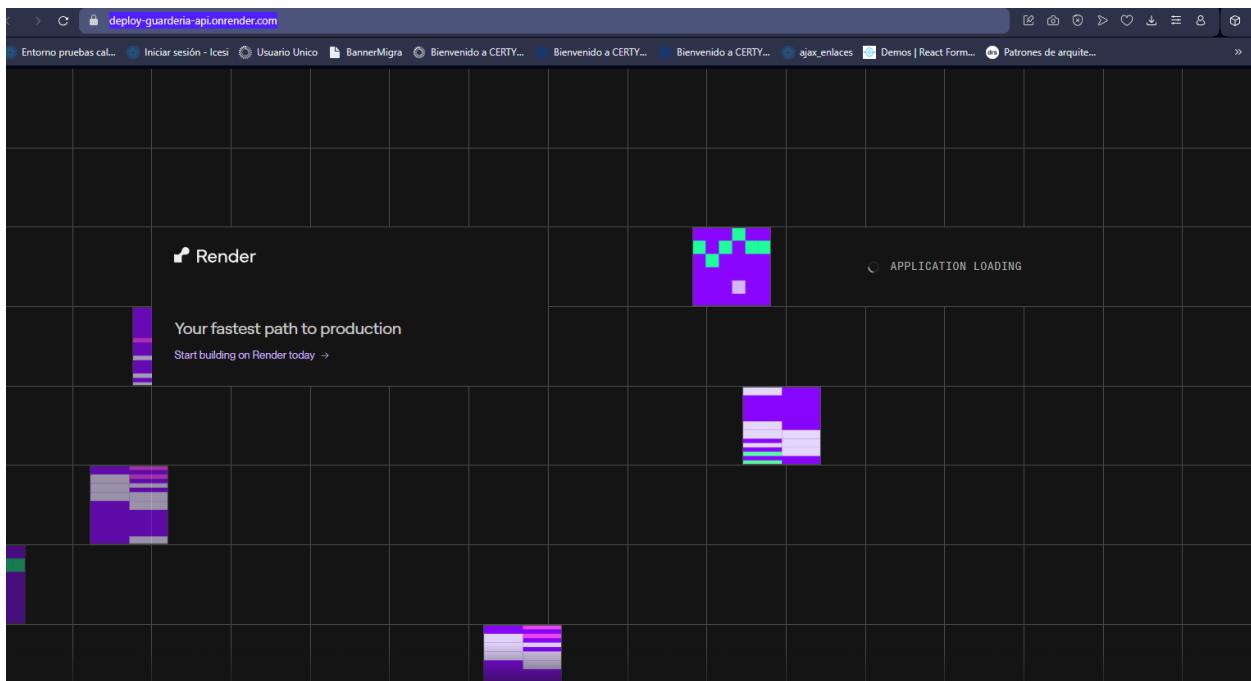
**Password:** ProySoftware123

**Nota** importante el servicio (Render) el cual use para desplegar la aplicación detecta si esta inactivo y se pausa es posible que cuando trate de ingresar el servicio este inactivo para esto por favor antes de realizar una petición al backend por favor ingrese a este URL cuando este nuevamente up le dará un mensaje de ok

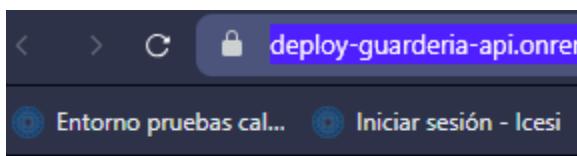
 Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

<https://deploy-guarderia-api.onrender.com>

Así se ve cuando está levantando el servicio



Cuando termine y pueda realizar peticiones al backend nuevamente se vería así



OK

## **Documento de formulación del proyecto.**

### **Introducción:**

A través de este proyecto, se propone el diseño e implementación de un sistema digital de inscripción que permita a los padres registrar a sus hijos en línea de manera rápida y segura. Utilizando tecnologías modernas como React para el frontend y Spring Boot para el backend, el sistema busca automatizar y optimizar el proceso, ofreciendo una solución práctica, accesible y confiable.

Nota (Este proyecto es la idea del curso análisis y diseño de software por lo tanto en la bibliografía hace referencia a dicho trabajo)

### **Preguntas formuladas:**

- ✓ ¿Qué dificultades enfrentan actualmente al gestionar las inscripciones de los niños de forma manual?
- ✓ ¿Cuánto tiempo promedio toma procesar una inscripción desde que el padre entrega el formulario hasta que se confirma el cupo?
- ✓ ¿Han ocurrido errores al transcribir los datos de los formularios físicos al sistema o registro interno? ¿Con qué frecuencia?
- ✓ ¿Cómo creen que un formulario digital impactaría su carga de trabajo y la eficiencia del proceso de inscripción?

### **Conclusión de las respuestas**

El proceso manual de inscripción genera demoras, errores en la transcripción de datos y sobrecarga de trabajo para el personal administrativo. El equipo reconoce que un formulario digital ayudaría a agilizar el proceso, reducir errores y organizar mejor la información. Por tanto, se confirma la necesidad de implementar una solución digital para mejorar la eficiencia y facilitar el trabajo diario.

### **Problemática:**

La guardería ubicada en Cal, Barrio Decepaz desarrolla el proceso de inscripción de manera manual, solicitando a los padres diligenciar formularios físicos para registrar a sus hijos. Esta práctica ha generado diversos inconvenientes, como pérdida de información, errores en la transcripción de datos y demoras en la confirmación de cupos, lo que afecta tanto a los padres como al personal administrativo. Ante esta situación, se consideran como posibles soluciones la implementación de un formulario digital y un sistema automatizado que facilite la gestión de inscripciones. Esto permitiría optimizar tiempos, reducir errores y mejorar la comunicación con las familias. Por tanto, el problema se formula de la siguiente manera: ¿Cómo puede la Guardería mejorar el proceso de inscripción para hacerlo más ágil, preciso y eficiente tanto para los padres como para el personal administrativo?

Alcance del proyecto:

### Funcionales

- Digitalizar los procesos manuales y con herramientas poco eficientes mediante la solución del formulario de inscripción de clases
- Implementación de un sistema web donde los padres puedan inscribirse a los cursos a sus hijos.
- Captura o recopilar datos esenciales del estudiante (nombre, ID, carrera, cursos deseados, etc.).
- Asignación automática de cupos basada en disponibilidad y requisitos del curso.
- Control de permisos según el rol (estudiantes, administrativos).
- Notificación por correo o en el sistema al completar la inscripción.

### Objetivo general

Desarrollar un sistema digital de inscripción para la guardería que permita automatizar el proceso de registro de niños, con el fin de mejorar la eficiencia administrativa, reducir errores y optimizar la gestión de datos.

### Objetivos específicos

- Recolectar información sobre el proceso actual de inscripción mediante encuestas y entrevistas al personal administrativo y a los padres de familia.
- Analizar los requerimientos funcionales y no funcionales del sistema a desarrollar, en base a la información recolectada.
- Diseñar la estructura del formulario digital y la base de datos necesaria para almacenar de forma segura la información de inscripción.

- Implementar el formulario digital y el sistema de gestión en una plataforma accesible desde el sitio web de la guardería.
- Evaluar el funcionamiento del sistema mediante pruebas con usuarios reales y ajustes según la retroalimentación recibida.

### Introducción idea principal

En la actualidad, la guardería enfrenta desafíos significativos en la gestión de inscripciones a cursos debido a su sistema manual basado en documentos físicos. Este método tradicional genera pérdida de información, retrasos en el procesamiento y errores en la asignación de cupos, lo que afecta la eficiencia administrativa y la experiencia estudiantil.

### Objetivo Principal

Desarrollar e implementar un sistema de inscripción digital que automatice el proceso de registro de estudiantes en cursos, optimizando la asignación de cupos, reduciendo la carga administrativa y asegurando la trazabilidad y seguridad de la información académica.

### Calidad de los Resultados Esperados

La implementación de este sistema garantizará un proceso de inscripción más eficiente y accesible para todos los actores involucrados. Se espera una reducción significativa en los tiempos de procesamiento, una mejora en la precisión de los registros y una mayor satisfacción tanto para los estudiantes como para el personal administrativo.

### Reflexión

La transformación digital en el ámbito académico no solo implica el uso de nuevas tecnologías, sino también un cambio en la forma en que se gestionan y optimizan los procesos. ¿Cómo puede la digitalización contribuir a mejorar otros aspectos de la administración educativa? ¿Qué otras áreas de la universidad podrían beneficiarse de esta evolución tecnológica? La implementación de este sistema representa un paso

hacia la modernización de la universidad, abriendo nuevas oportunidades para la mejora continua en la gestión académica.

## Soluciones al Problema

Para abordar las dificultades en la gestión de inscripciones en la Universidad XYD, se plantea una solución integral basada en la digitalización del proceso. El objetivo es reducir la carga administrativa, minimizar errores en la asignación de cupos y mejorar la experiencia de los estudiantes mediante la automatización de tareas clave.

- Desarrollo de un formulario en línea para que los estudiantes puedan inscribirse sin necesidad de documentos físicos.
- Validaciones en tiempo real para evitar errores en el ingreso de datos.
- Confirmación automática de inscripción a través de correo electrónico o notificaciones en el sistema.

Stackholders:

Estudiantes

- ✓ Proceso de inscripción rápido y sin trámites físicos.
- ✓ Validación inmediata de disponibilidad de cupos.

Personal Administrativo

- ✓ Reducción de la carga manual en el procesamiento de inscripciones.
- ✓ Mayor control sobre la inscripción y asignación de cupos.
- ✓ Eliminación de errores en la asignación manual de estudiantes.

Directivos de la Universidad

- ✓ Mejora en la eficiencia del proceso académico.
- ✓ Ahorro de costos operativos al eliminar procesos manuales y documentos físicos.

## Alcance del Proyecto

El proyecto busca diseñar e implementar un sistema digital de inscripciones para la guardería, que permita a los padres registrar a sus hijos de manera rápida, segura y desde cualquier lugar. El sistema incluirá un formulario en línea accesible desde el sitio web, almacenamiento automatizado en base de datos, confirmaciones automáticas por

correo y un panel administrativo para el personal. El objetivo es mejorar la eficiencia operativa, reducir errores y brindar una mejor experiencia a los usuarios.

## Módulos del Sistema

### Formulario de inscripción digital

- Permite a los padres registrar a sus hijos llenando un formulario en línea con información básica como nombres, edades, datos de contacto y observaciones médicas. Sistema de gestión de inscripciones

### Módulo de Gestión Administrativa

- Panel exclusivo para el personal de la guardería donde se pueden revisar, aprobar o rechazar inscripciones. También permite filtrar por edades, disponibilidad de cupos y fechas.

### Base de datos centralizada

- Almacena de forma segura la información de los niños inscritos y sus representantes. Permite búsquedas rápidas y genera reportes para el personal.

### Envío de correos automáticos

- Confirmación de recepción de solicitud

### Tecnologías y herramientas del proyecto:

- React.js: Biblioteca principal para construir la interfaz de usuario.
- Material UI o Tailwind CSS: Para estilos modernos y componentes visuales.
- React Hook Form + Yup: Para validar y gestionar los formularios de inscripción.

### Backend

- Spring Boot: Framework principal del backend para construir la API REST.

- Spring Data JPA: Para conectar y manipular la base de datos mediante entidades.
- Spring Security (opcional): Para controlar el acceso al sistema administrativo.
- Lombok: Para reducir la escritura de código repetitivo (getters, setters, etc.).

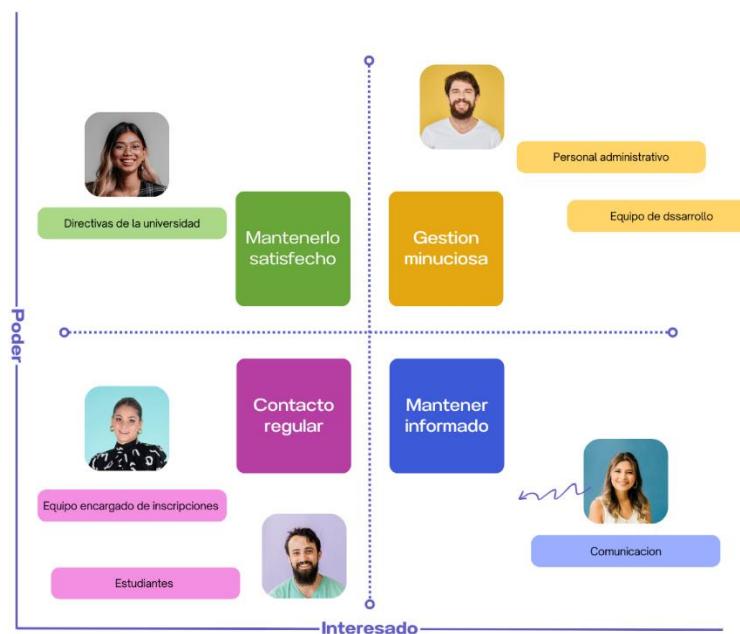
## Base de Datos

- MySQL: Como sistema de gestión de base de datos relacional.

## Herramientas de Desarrollo y Despliegue

- Postman: Para probar la API del backend.
- Git + GitHub: Para control de versiones y colaboración.
- Netlify / Vercel (Frontend) y Render / Railway / Heroku (Backend): Plataformas gratuitas para desplegar el sistema.

## Mapa stackholders



Matriz:

ID	Riesgo	Probabilidad	Impacto	Nivel de Riesgo	Estrategia de Mitigación
R1	Resistencia al cambio por parte del personal administrativo	Media	Alta	Alto	Capacitar al personal y mostrar beneficios del nuevo sistema.
R2	Fallos técnicos en el formulario digital	Alta	Alta	Crítico	Realizar pruebas antes del despliegue y contar con soporte técnico disponible.
R3	Pérdida de datos o errores en la base de datos	Media	Alta	Alto	Implementar copias de seguridad automáticas y validaciones en el ingreso.
R4	Acceso no autorizado a datos personales	Media	Alta	Alto	Usar autenticación, roles de acceso y cifrado de datos.
R5	Baja adopción por parte de estudiantes	Baja	Media	Medio	Crear una interfaz intuitiva y ofrecer ayuda técnica o tutoriales.

Cronograma:

Sprint	Duración	Objetivos / Actividades
Sprint 1	1 semana	<ul style="list-style-type: none"> <li>- Reunión inicial con stakeholders</li> <li>- Definición de requerimientos</li> <li>- Creación del backlog de producto</li> </ul>
Sprint 2	2 semanas	<ul style="list-style-type: none"> <li>- Diseño de interfaz del formulario</li> <li>- Maquetación inicial</li> <li>- Historias: UI básica, campos de entrada</li> </ul>

Sprint 3	2 semanas	- Conexión con base de datos - Validaciones del formulario - Historias: registro y validación
Sprint 4	2 semanas	- Gestión de usuarios y autenticación - Seguridad básica - Historias: inicio de sesión, roles
Sprint 5	2 semanas	- Pruebas del sistema - Corrección de errores - Historias: testing, feedback de usuarios
Sprint 6	1 semana	- Despliegue del formulario - Documentación final - Capacitación del personal administrativo

## Enlace JIRA

<https://andresfpv20001512-1749404639858.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?atlOrigin=eyJpIjoiMGFkNDc3MGY1ZjFNGVjNmI0YzljODNkYTRjMTM5NjQiLCJwIjoiaj9>

## Diseño del prototipo.

### Requerimientos funcionales y no funcionales

#### RF01 - Registro de Usuarios:

El sistema debe permitir a administradores crear nuevos administradores

#### RF02 - Autenticación y autorización:

El sistema debe permitir a los usuarios autenticarse mediante username y password, utilizando JWT para mantener sesiones seguras.

#### RF03 - Gestión de roles:

El sistema debe asignar roles (Administrador, Profesor, Tutor) para controlar el acceso a funcionalidades específicas.

**RF04 - Registro de niños:**

Los padres/tutores pueden registrar la información básica del niño (nombre, fecha de nacimiento, alergias, notas médicas).

**RF05 - Inscripción a cursos:**

El sistema debe permitir que los padres inscriban a los niños en cursos disponibles.

**RF06 - Gestión de cursos:**

Los administradores pueden crear, modificar y eliminar cursos, definiendo edad mínima/máxima, capacidad y horario.

**RF07 - Subida y gestión de documentos:**

Los acudientes pueden subir documentos requeridos para la inscripción, asociados a cada inscripción.

**RF08 - Consulta de estado de inscripción:**

Administradores pueden consultar el estado de la inscripción (pendiente, aprobada, rechazada).

**RF09 - Aprobación/Rechazo de inscripciones:**

Los administradores pueden aprobar o rechazar inscripciones realizadas.

**RF10 - Gestión de profesores:**

Los administradores pueden crear, modificar y eliminar profesores, incluyendo sus datos personales y asignaciones a cursos.

**RF11 - Generación de reportes:**

El sistema debe permitir a los administradores generar reportes sobre inscripciones, cursos, y estados de los alumnos en formatos descargables (por ejemplo, PDF o Excel).

#### RF12 - Historial de inscripciones:

Los usuarios pueden consultar el historial completo de inscripciones y cursos pasados.

### **Requerimientos No Funcionales (RNF)**

#### RNF01 - Seguridad:

El sistema debe utilizar autenticación basada en JWT y autorización por roles para proteger los datos y funcionalidades.

#### RNF02 - Usabilidad:

La interfaz debe ser fácil de usar e intuitiva para usuarios con conocimientos básicos en tecnología.

#### RNF03 - Rendimiento:

El tiempo de respuesta del sistema ante cualquier solicitud no debe superar los 3 segundos bajo carga normal.

#### RNF04 - Escalabilidad:

La arquitectura debe soportar un crecimiento gradual en el número de usuarios y datos sin afectar el rendimiento.

#### RNF05 - Logging y auditoría:

El sistema debe registrar eventos importantes (login, cambios en datos, inscripciones) para auditoría y seguridad.

#### RNF06 - Responsividad:

La interfaz debe adaptarse correctamente a diferentes tamaños de pantalla (móvil, tablet, desktop).

## RNF07 - Tolerancia a fallos:

El sistema debe manejar errores de forma controlada y mantener operativa la mayor parte del sistema en caso de fallos parciales.

## Historias de Usuario JIRA:

<https://andresfpv20001512-1749404639858.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?atlOrigin=eyJpIjoiMGFkNDc3MGY1ZjFINGVjNml0YzljODNkYTRjMTM5NjQiLCJwIjoiaiJ9>

The screenshot shows the JIRA interface with the URL <https://andresfpv20001512-1749404639858.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?atlOrigin=eyJpIjoiMGFkNDc3MGY1ZjFINGVjNml0YzljODNkYTRjMTM5NjQiLCJwIjoiaiJ9>. The left sidebar shows navigation links like 'Para ti', 'Recientes', 'Marcados como fav...', 'Aplicaciones', 'Planes', 'Proyectos', 'Recientes', 'Guarderia' (selected), 'Ver todos los proyectos', 'Equipos', and 'Más'. The main area displays the 'Guarderia' backlog with tabs for 'Resumen', 'Backlog' (selected), 'Tablero', 'Calendario', 'Cronograma', 'Páginas', 'Formularios', and '+'. Below the tabs is a search bar and a filter dropdown set to 'Epic'. A list of user stories is shown, each with a checkbox, title, status, and a 'Details' button. The stories are:

- SCRUM-6 Como desarrollador, quiero crear el servicio API para registrar inscri... FINALIZADA
- SCRUM-11 Como usuario, quiero una pantalla de login sencilla y segura, para... FINALIZADA
- SCRUM-9 Crear el servicio API para gestionar profesores (CRUD), para mante... TAREAS POR HACER
- SCRUM-12 Como administrador, quiero un panel de control que muestre inscri... EN CURSO
- SCRUM-13 Generación de reportes EN CURSO
- SCRUM-7 Crear el servicio API para login, para que los usuarios puedan aut... EN CURSO
- SCRUM-14 Creación de diagrama de secuencia FINALIZADA
- SCRUM-15 Creacion Diagrama de arquitectura FINALIZADA
- SCRUM-16 Creacion Diagrama de clases y componentes FINALIZADA

At the bottom, there is a 'Backlog (0 actividades)' summary and a 'Quickstart' button.

## Diagrama de clases

Para visualizar mejor Ingresar en el OneDrive

[https://laiberocol-my.sharepoint.com/:f/g/personal/apiptcan\\_estudiante\\_ibero\\_edu\\_co/Ejyi7BsnMbxAv0eZHQLJE0B6ezhEln3kxMqtEQEXnR1Xw?e=Us6Ns3](https://laiberocol-my.sharepoint.com/:f/g/personal/apiptcan_estudiante_ibero_edu_co/Ejyi7BsnMbxAv0eZHQLJE0B6ezhEln3kxMqtEQEXnR1Xw?e=Us6Ns3)

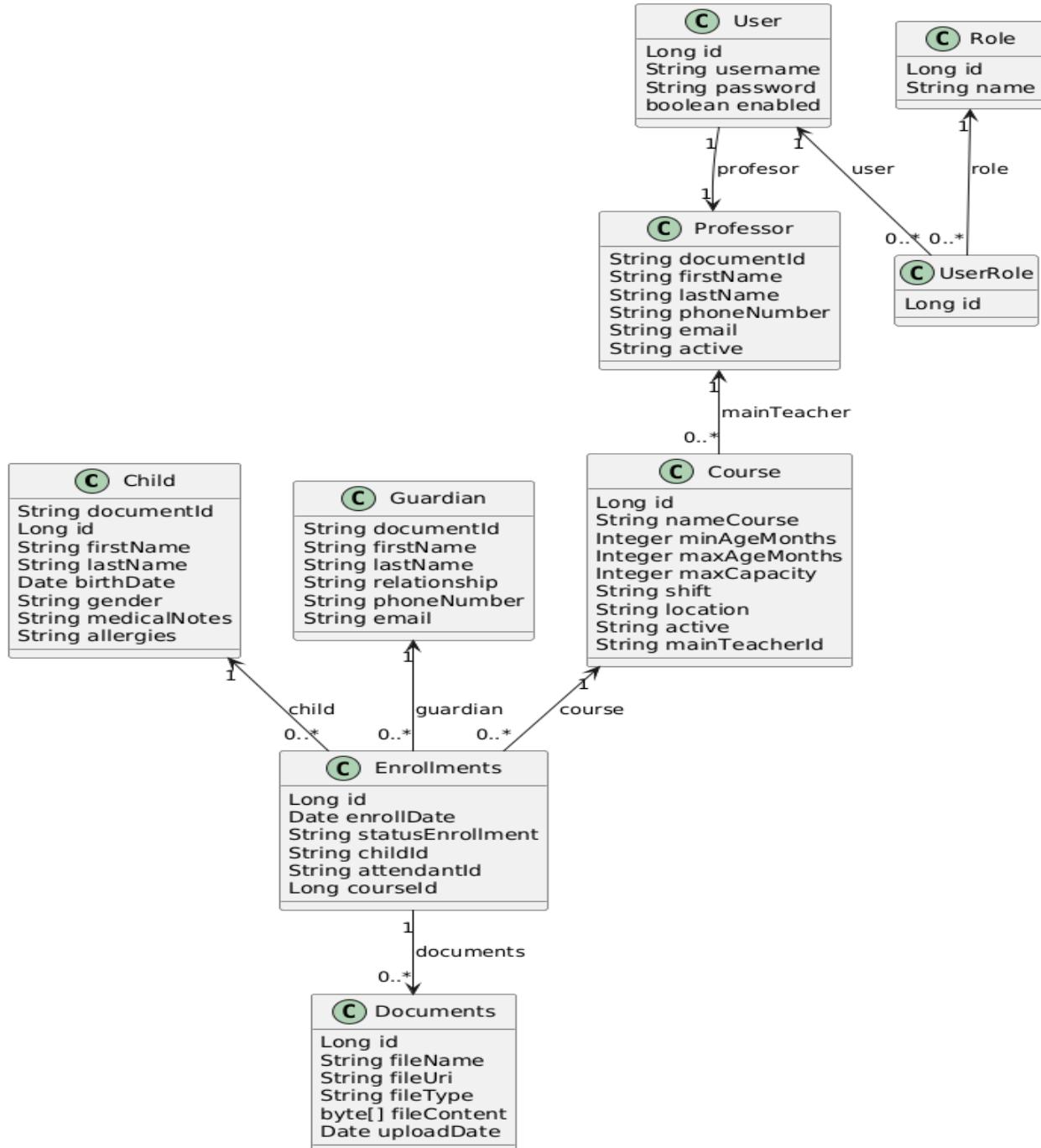


Diagrama de componentes

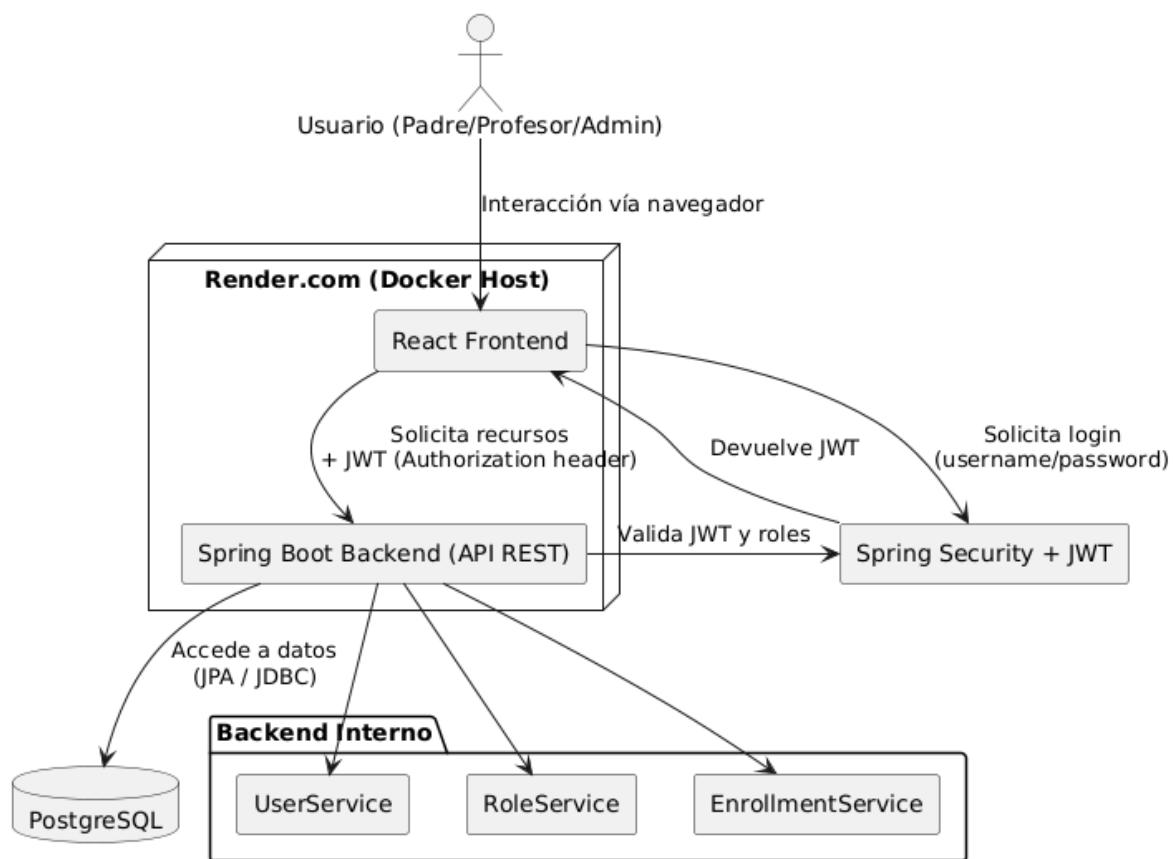
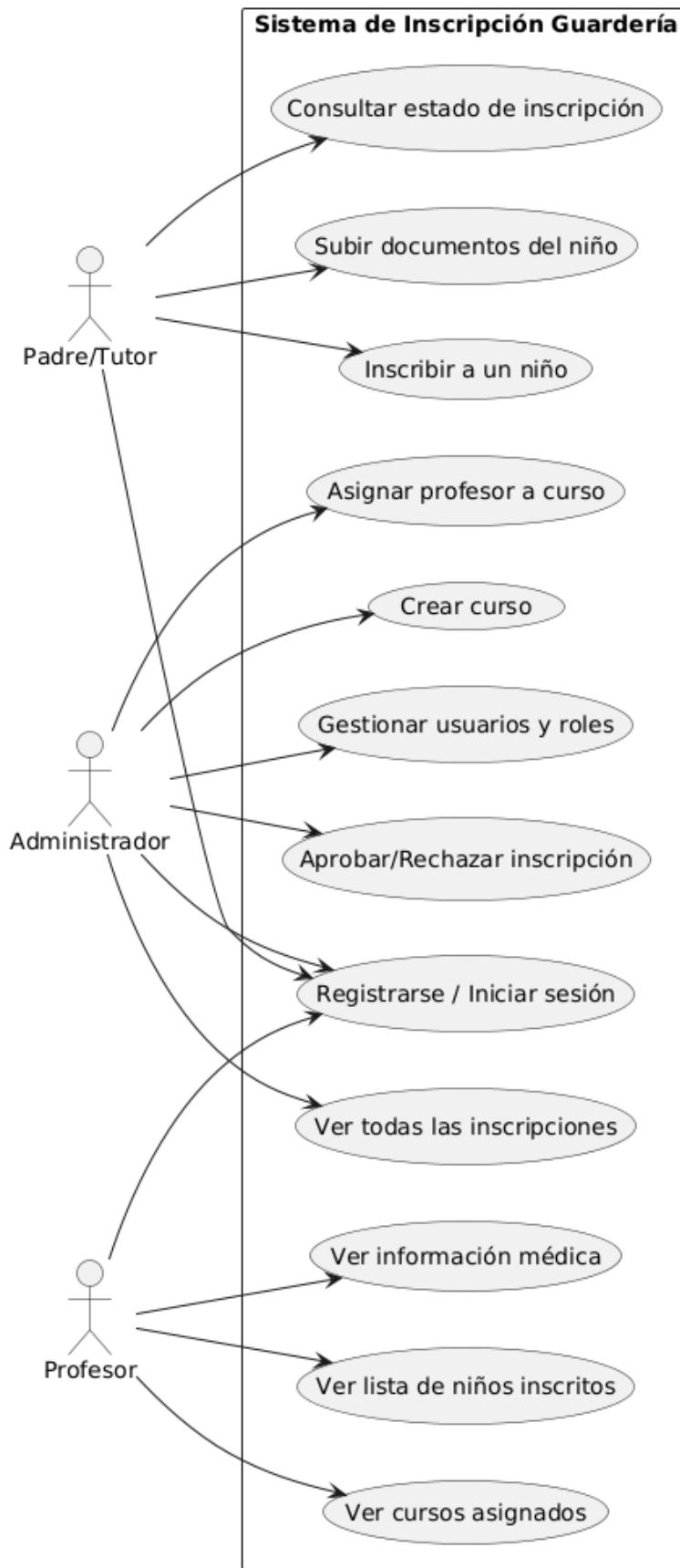
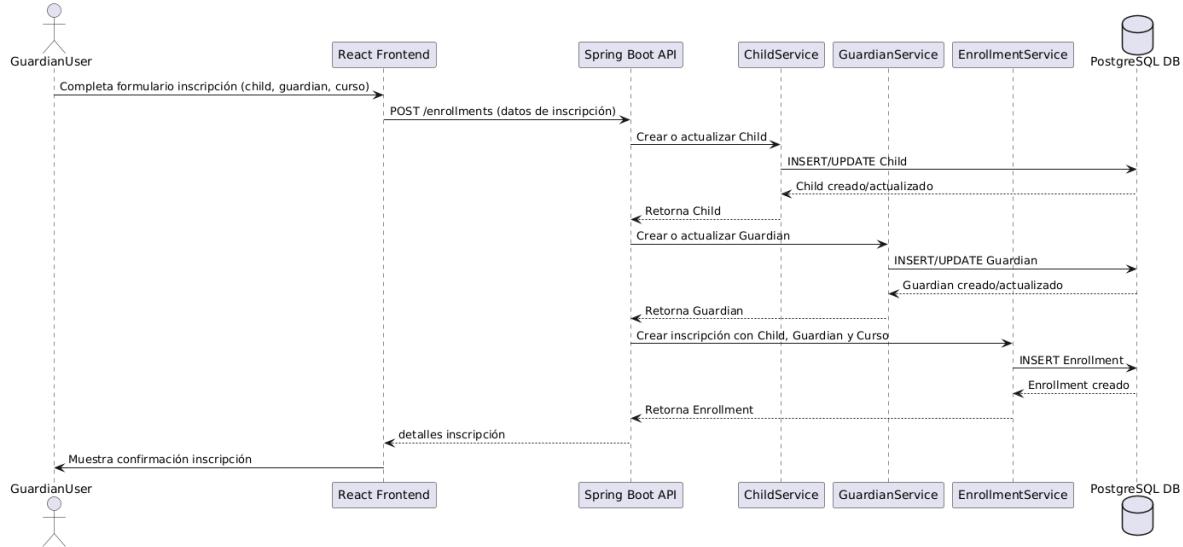


Diagrama de casos de uso

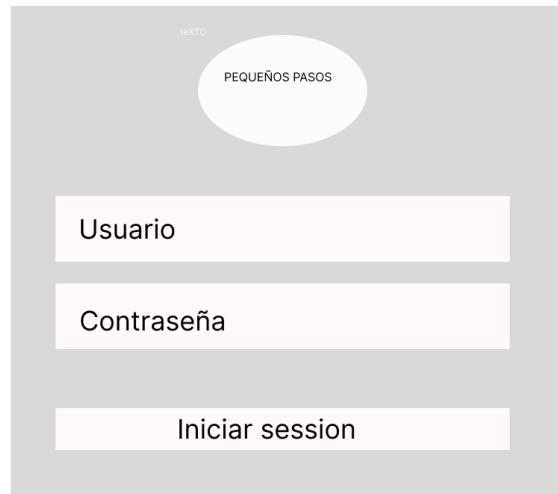


## Diagrama de secuencia proceso de inscripción



Desarrollo de WireFrames para visualizarlos mejor por favor ingresar al OneDrive

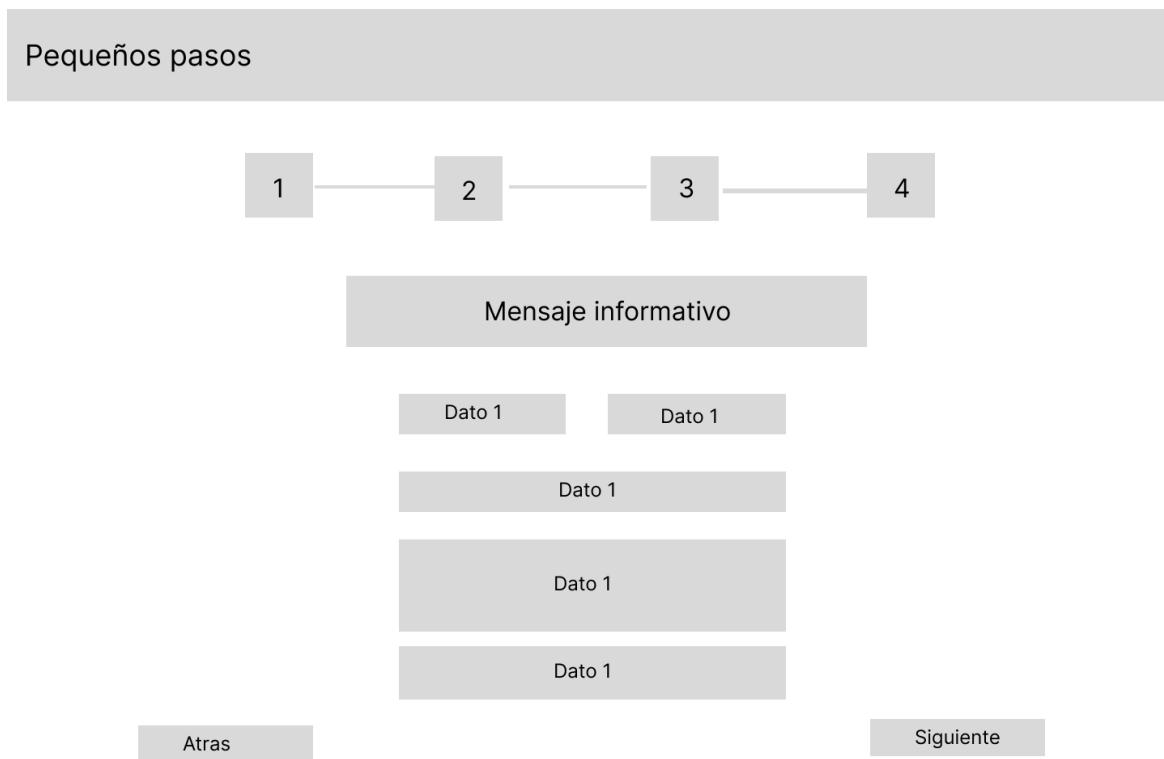
[https://laiberocol-my.sharepoint.com/:f/g/personal/apipecan\\_estudiante\\_ibero\\_edu\\_co/EiRiU9bW3NZKoFtBythe03UB3RXLsjK-HPzWIWUDMP1LEQ?e=URYfoG](https://laiberocol-my.sharepoint.com/:f/g/personal/apipecan_estudiante_ibero_edu_co/EiRiU9bW3NZKoFtBythe03UB3RXLsjK-HPzWIWUDMP1LEQ?e=URYfoG)



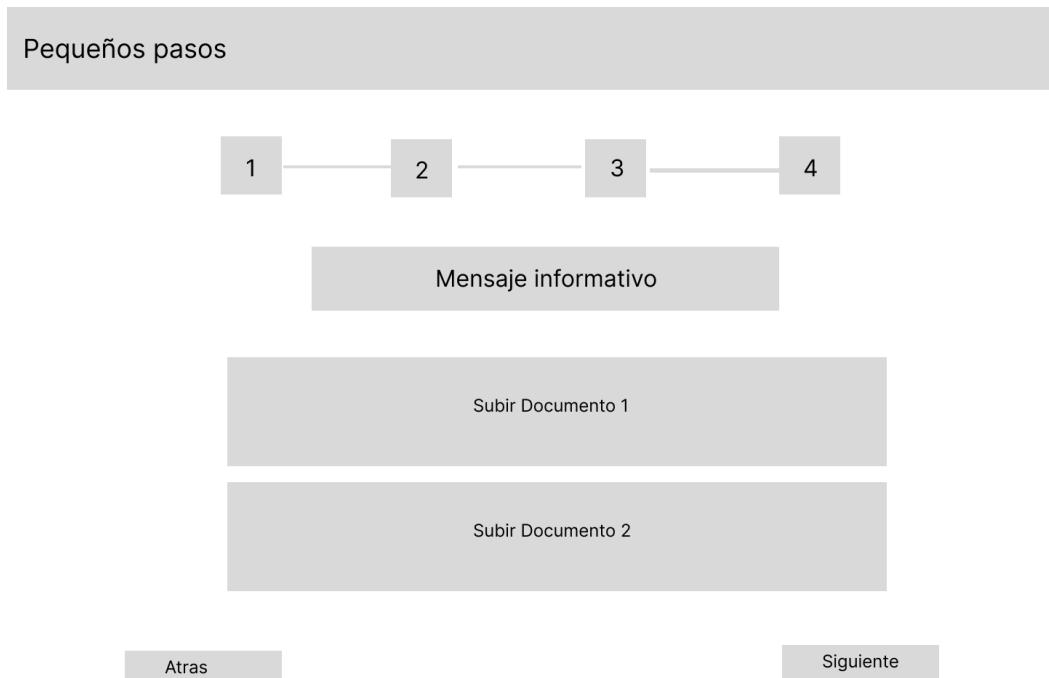
## Admin Page

A wireframe diagram of an admin dashboard. On the left, there's a sidebar titled "Pequeños pasos" containing four "Opcion 1" buttons. The main area has a header with "Bienvenido Nombre" and an "Opcion" button. Below the header is a large rectangular container with a title "Titulo Modulo".

## Pantalla principal del formulario



## Pantalla subir documentos



## Manual Técnico

### Resumen del Proyecto

Este proyecto consiste en el desarrollo de un sistema cliente-servidor para gestionar digitalmente el proceso de inscripción en una guardería. Se implementa una aplicación web que permite a los padres registrar a sus hijos mediante un formulario en línea, y a su vez proporciona al personal administrativo una interfaz para gestionar inscripciones y visualizar la información de forma estructurada y segura. El sistema busca mejorar la eficiencia, reducir errores y facilitar el acceso a los datos.

### Tecnologías Principales Utilizadas

#### Backend:

- Java 17
- Spring Boot 3
- Spring Security (con JWT)
- Spring Data JPA
- Base de datos Oracle (SQLDeveloper)

#### Frontend:

- React 18.3.1
- Material UI (MUI)
- React Hook Form (RHF) + Yup para validaciones
- Axios
- Redux Toolkit
- React Router Dom
- React Query

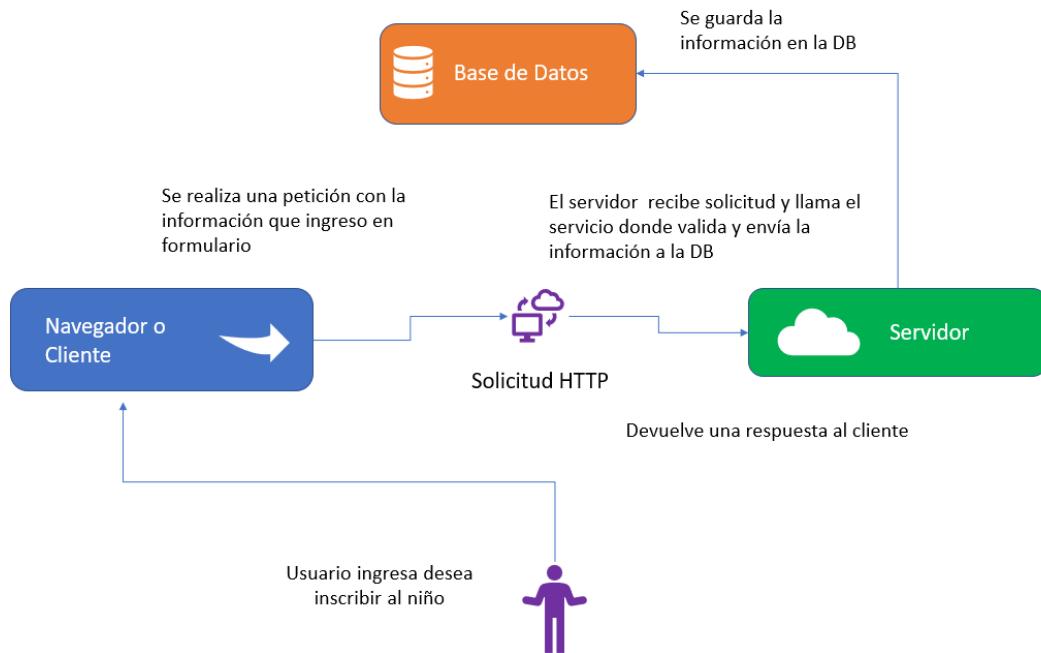
#### Despliegue:

- Render
- Docker
- PostgreSQL (Debido que render tiene solo servicio para DB PostgreSQL fue más fácil cambiarla)

## Arquitectura General

La aplicación sigue una arquitectura cliente-servidor desacoplada. El frontend React se comunica con el backend Spring Boot mediante API REST. La seguridad se implementa con Spring Security y JWT, protegiendo rutas y datos. Los formularios se validan en el cliente con RHF (React Hook Form) + Yup, y los datos se almacenan en una base de datos Oracle a través de Spring Data JPA. El flujo general garantiza una experiencia de usuario fluida y una gestión administrativa eficiente.

## Arquitectura del funcionamiento de la aplicación grafico



## Estructura de proyecto Backend (Spring Boot)

Dentro de la carpeta src/main/java, se encuentra el paquete principal del proyecto, que contiene la organización base de los distintos componentes de la API.

- **Auth** – Configuración de Spring Security Filtros o interceptores y servicio para gestionar el JWT
- **Constants** – Paquete donde para las constantes EJ. esta las constantes de URI de los endpoint
- **Controllers** - contiene los controladores REST que gestionan las solicitudes entrantes.

- **Dto** - define los objetos de transporte de datos entre capas.
- **Entity** - Las entidades que representan las tablas en la base de datos Oracle.
- **Repository** – Define las interfaces que extienden de JpaRepository utilizadas para realizar operaciones CRUD con la base de datos. Spring Data se encarga de su implementación automática.
- **Service** – Implementa la lógica de negocio de la aplicación. Los servicios intermedian entre los controladores y los repositorios, aplicando reglas de negocio y validaciones.

## **Estructura principal del proyecto Frontend (React)**

Dentro de la carpeta src/, se encuentra el paquete principal del proyecto, que contiene la organización base de los distintos componentes de la APP React.

- Api – Carpeta donde agrupa los llamados HTTP al API con axios
- Components – Carpeta que agrupa los componentes reutilizables de la aplicación,
- Config -incluye la configuración de librerías clave de la aplicación, como React Query
- Context -> Define y exporta contextos globales para el manejo del estado compartido entre componentes, cuando no se requiere Redux.
- Hooks - Carpeta que agrupa los custom hooks globales de la app
- Layout - Define los templates estructurales de la interfaz, como el Dashboard
- Routes - Rutas de navegación de la app
- Store – Configuración y gestión de slices de redux toolkit
- Views – Representa las páginas principales de la aplicación, como el login, el formulario de inscripción, la gestión de cursos, el panel administrativo, entre otros. Cada vista puede estar compuesta por múltiples componentes internos.

## **Seguridad**

El sistema implementa un esquema de seguridad basado en Spring Security 6 junto con JSON Web Tokens (JWT) para proteger los recursos del backend y controlar el acceso según roles definidos.

## **Procesamiento:**

La autenticación se realiza mediante JWT, que se genera en el login y se incluye en las solicitudes posteriores mediante la cabecera Authorization. El token contiene los datos del usuario y sus roles, y se valida en cada petición protegida por filtros personalizados.

## Flujo de Autenticación

- El usuario ingresa sus credenciales mediante el formulario de login.
- El backend valida las credenciales y genera un JWT si son correctas.
- El token es devuelto al cliente y almacenado en el frontend (localStorage).
- En cada solicitud protegida, el cliente incluye el token en la cabecera: (Authorization: Bearer <token generado>)

## Modelado de Datos

Entidades Principales

### **Teachers (Profesores):**

Guarda información básica de los profesores, incluyendo documento de identidad, nombre, teléfono y correo. Tiene un campo ACTIVE para indicar si el profesor está activo.

### **Courses (Cursos):**

Define los cursos disponibles en la guardería con atributos como nombre, rango de edad permitido (MIN\_AGE\_MONTHS, MAX\_AGE\_MONTHS), capacidad máxima, turno (SHIFT), ubicación y un profesor principal asignado.

### **Children (Niños):**

Registra la información personal de los niños inscritos, incluyendo documento, nombre, fecha de nacimiento, género y datos médicos relevantes (notas médicas y alergias).

### **Guardians (responsables):**

Contiene datos de los tutores o responsables legales de los niños, con información de contacto y parentesco.

**Enrollments (Inscripciones):**

Relaciona a un niño con un curso y un tutor, registrando la fecha de inscripción y el estado actual. Es la entidad central que vincula a niños, tutores y cursos.

**Documents (Documentos):**

Almacena archivos asociados a una inscripción, tales como documentos de identidad o certificados de vacunación. Puede almacenar el archivo directamente o solo la ruta.

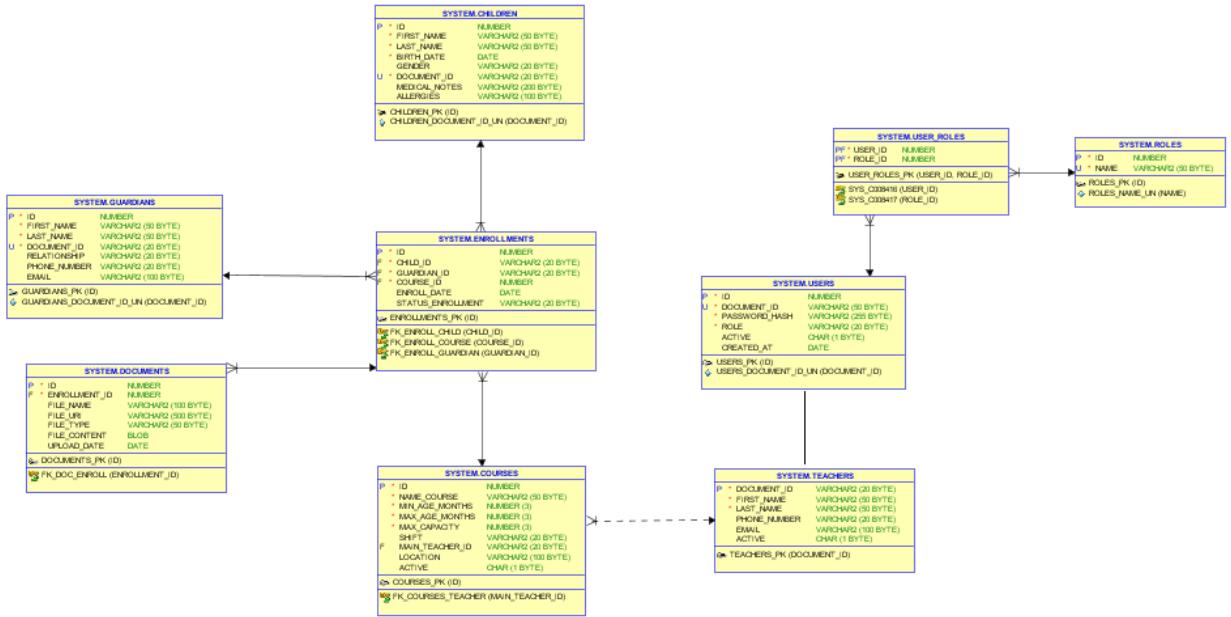
**Users (Usuarios):**

Registra los usuarios del sistema con credenciales y rol asignado (ADMIN, TEACHER, GUARDIAN, etc.), para la gestión de acceso.

**Roles y User\_Roles:**

Gestionan la asignación de roles a los usuarios mediante una relación muchos a muchos, permitiendo múltiples permisos por usuario.

**Diagrama ER (También adjunto en la documentación)**



## Detalles de API REST (Principales Endpoints)

Método	Endpoint URI	Descripción	Requiere Auth	Parámetros
<b>/authentication</b>				
<b>POST</b>	/authentication	Registrar un nuevo usuario	No	Body: RegisterAuth
<b>POST</b>	/authentication/login	Generar token JWT	No	Body: AuthCredentials

Método	Endpoint	Descripción	Requiere Auth	Parámetros
<b>/api/enrollment</b>				
<b>GET</b>	/api/enrollment	Obtener todas las inscripciones	Si	—
<b>GET</b>	/api/enrollment/{id}	Obtener inscripción por ID	Si	id: Long
<b>POST</b>	/api/enrollment	Crear nueva inscripción	Si	Body: Enrollment
<b>PUT</b>	/api/enrollment/{id}	Actualizar inscripción completa	Si	id, Body: Enrollment
<b>PATCH</b>	/api/enrollment/{id}/{status}	Actualizar estado de inscripción	Si	id: Long, status: String
<b>DELETE</b>	/api/enrollment/{id}	Eliminar inscripción	Si	id: Long
<b>GET</b>	/api/enrollment/{courseId}/students	Obtener estudiantes aceptados por curso	Si	courseId: Long

Método	Endpoint	Descripción	Requiere Auth	Parámetros
<b>/api/course</b>				
<b>GET</b>	/api/course/active	Obtener cursos activos	SI	
<b>GET</b>	/api/course/{id}	Obtener curso por ID	SI	id: Long
<b>POST</b>	/api/course	Crear un nuevo curso	SI	Body: Course
<b>DELETE</b>	/api/course/{id}	Eliminar curso	SI	id: Long

Método	Endpoint	Descripción	Requiere Auth	Parámetros
<b>/api/professor</b>				
<b>GET</b>	/api/professor	Obtener todos los profesores (paginado)	SI	Parámetros de paginación (Pageable)
<b>GET</b>	/api/professor/{id}	Obtener profesor por documento ID	SI	id: String
<b>POST</b>	/api/professor	Crear nuevo profesor	SI	Body: Professor
<b>PUT</b>	/api/professor/{id}	Actualizar profesor	SI	id, Body: Professor
<b>DELETE</b>	/api/professor/{id}	Eliminar profesor	SI	id: String

Método	Endpoint	Descripción	Requiere Auth	Parámetros
<b>/api/child</b>				
<b>GET</b>	/api/child	Obtener todos los niños	SI	—
<b>GET</b>	/api/child/{id}	Obtener niño por documento ID	SI	id: String
<b>POST</b>	/api/child	Crear nuevo niño	SI	Body: Child
<b>PUT</b>	/api/child/{id}	Actualizar información del niño	SI	id, Body: Child
<b>DELETE</b>	/api/child/{id}	Eliminar niño	SI	id: String

## Flujo de Funcionamiento del Sistema

El sistema de inscripción digital para la guardería opera bajo el siguiente flujo general:

### 1. Registro de Inscripción por Parte del Padre o Acudiente:

- El padre o acudiente accede al formulario de inscripción en línea.
- Completa la información personal del niño, incluyendo nombre, fecha de nacimiento, género, notas médicas y alergias.

- Selecciona el curso al que desea inscribir al niño, indicando la jornada correspondiente (mañana, tarde o jornada completa).
- Ingresa los datos del acudiente, como nombre, documento, parentesco, teléfono y correo electrónico.
- De manera opcional, puede adjuntar documentos relevantes (por ejemplo, documento de identidad, carné de vacunación, etc.).

## 2. Revisión y Gestión por Parte del Administrador:

- Un usuario con rol administrador accede al panel de gestión.
- Revisa las inscripciones pendientes y, según criterio, puede aprobar o rechazar las solicitudes.
- Tiene la capacidad de crear nuevos cursos, especificando parámetros como edad mínima/máxima, jornada, capacidad máxima, ubicación y profesor principal.
- Puede registrar profesores y asignarlos a los cursos disponibles, asegurando que cada grupo tenga un docente responsable.

## 3. Comunicación y Seguimiento:

- La información queda registrada en la base de datos Oracle, permitiendo su posterior consulta, auditoría o generación de reportes.

## Conclusión

El desarrollo e implementación del sistema digital de inscripción para la guardería ubicada en el barrio Decepaz representa un avance significativo en la modernización de sus procesos administrativos. Al sustituir los formularios físicos por una solución web segura, accesible y eficiente, se logró reducir los errores de transcripción, optimizar los tiempos de gestión y mejorar la experiencia tanto para los padres como para el personal.

El uso de tecnologías modernas como Spring Boot 3 con seguridad basada en JWT, una base de datos Oracle robusta y un frontend dinámico construido en React con herramientas como MUI, React Hook Form, Redux y Axios, permite que el sistema sea escalable, mantenable y fácil de usar.

Este sistema sienta las bases para una transformación digital completa, y se encuentra abierto a futuras ampliaciones como notificaciones automáticas, reportes estadísticos o integración con plataformas educativas.

**Pruebas las pruebas de integración, pruebas unitarias y automatizadas puede verlas en esta carpeta en .md o en el archivo que genera las pruebas e2e**

[https://laiberocol-my.sharepoint.com/:f/g/personal/apipican\\_estudiante\\_ibero\\_edu\\_co/Eps2LxZ5nMtEgsqHQRBJmTcBoxYCE369XZYUysukJEcEVA?e=n6ju6p](https://laiberocol-my.sharepoint.com/:f/g/personal/apipican_estudiante_ibero_edu_co/Eps2LxZ5nMtEgsqHQRBJmTcBoxYCE369XZYUysukJEcEVA?e=n6ju6p)

## Casos de pruebas

ID	Funcionalidad	Descripción	Entrada	Resultado Esperado
CP01	Inicio de Sesión	Inicio de sesión exitoso con credenciales válidas	Inicio de sesión exitoso con credenciales válidas	Acceso concedido, redireccionamiento a página principal

CP02	Inicio de Sesión	Inicio de sesión fallido por credenciales erróneas	Inicio de sesión fallido por credenciales erróneas	Mensaje de error indicando credenciales inválidas, acceso denegado
CP03	Formulario de Inscripción	Validación de campos obligatorios en el formulario	Validación de campos obligatorios en el formulario	Mensajes de validación que indican los errores y no permite enviar
CP04	Formulario de Inscripción	Envío correcto del formulario con todos los datos válidos	Envío correcto del formulario con todos los datos válidos	Formulario enviado exitosamente con mensaje de confirmación
CP05	Página Administrativa - Crear Curso	Creación exitosa de un nuevo curso con datos válidos	Creación exitosa de un nuevo curso con datos válidos	Curso creado y listado en la sección de cursos
CP06	Página Administrativa - Crear Curso	Intento de crear un curso con datos inválidos o incompletos	Intento de crear un curso con datos inválidos o incompletos	Mensaje de error que impide crear el curso
CP01	Inicio de Sesión	Inicio de sesión exitoso con credenciales válidas	Inicio de sesión exitoso con credenciales válidas	Acceso concedido, redireccionamiento a página principal

## Pruebas de usabilidad

### Objetivo

Evaluar la usabilidad del formulario de inscripción para niños en la guardería y la página administrativa, donde se gestionan cursos y administradores, asegurando que los usuarios pueden completar tareas clave de forma eficiente y satisfactoria.

## Alcance

- Formulario de inscripción: ingreso y envío de datos de niños.
- Página administrativa: creación de cursos para ofrecer y creación de administradores.
- Usuarios: padres o tutores para el formulario; personal administrativo para la gestión.

## Metodología

Tipo: Prueba Virtual moderada.

Técnicas: Test de tareas + observación directa + cuestionario breve de satisfacción.

Duración: 15 minutos por participante.

Herramientas: ZOOM

## Escenario y Tareas

Formulario de inscripción (usuarios padres/tutores)

Completar el formulario con datos personales del niño y tutor.

- Subir documentos requeridos (si aplica).
- Enviar el formulario y recibir confirmación.

## Página administrativa (usuarios administrativos)

- Crear un nuevo curso para ofrecer en la guardería.
  - Modificar información de un curso existente.
- 
- Crear un nuevo usuario administrador.
  - Indicadores de éxito:

## Resultados y Observaciones

- Observaciones del Formulario de Inscripción
- Mejorar la visualización del logo del sistema.
- Mejorar los mensajes de ayuda para los procesos de inscripción, por ejemplo, en la inclusión.
- Los campos del formulario deberían tener el indicador de obligatoriedad (\*).
- Incluir horarios específicos para el paso de selección del curso.
- Evaluar la inclusión de una ventana resumen antes de finalizar el formulario con la información ingresada.
- Sería útil incluir un enlace o botón que dirija a un tutorial o chat para resolver dudas.

### Puntos Buenos del Formulario

- El formulario tiene un formato secuencial que facilita la navegación y comprensión.
- Diseño amigable e intuitivo.
- Los mensajes de ayuda u orientación son un buen detalle para guiar al usuario.

### Observaciones de la Página Administrativa

- Los iconos o botones para gestión de inscripciones deberían tener tooltips que expliquen su función.
- Evaluar la posibilidad de capturar más información sobre docentes y cursos.
- Mejorar los filtros en los módulos para buscar información por edad, jornada o estado (activo/inactivo)

### Puntos Buenos de la Página Administrativa

- La sección para revisar inscripciones permite visualizar la información necesaria para tomar decisiones.
- La generación de reportes aporta valor al sistema.
- El diseño es amigable, con colores suaves representativos de la guardería.

## Conclusiones

La aplicación es intuitiva para la mayoría de las tareas, especialmente en el formulario de inscripción. En la página administrativa, se identificaron pequeñas barreras para usuarios menos experimentados, sobre todo en la gestión de roles y acciones de guardado. Las observaciones señalan áreas claras de mejora para optimizar la experiencia.

## Recomendaciones

- Añadir tooltips y ayudas contextuales en la página administrativa.
- Mejorar la visibilidad de elementos clave como botones “Guardar” y mensajes de confirmación.
- Revisar terminología y añadir indicadores de obligatoriedad en el formulario.
- Considerar la implementación de un resumen antes de enviar el formulario y recursos de ayuda accesibles (tutoriales, chat).

## Referencias

Gómez Gutiérrez, H. (2013). Bloque IV. En Física I: (2 ed.). Cengage Learning.

Labarca Briones, R. (2014). Capítulo 6. Integral de Riemann y Capítulo 7. Aplicaciones de la integral. En Cálculo en una variable. Editorial Universidad de Santiago de Chile.

Pipicano Villa Andres Felipe 2025 Propuesta proyecto (Manuscrito no publicado). Universidad Iberoamericana, Curso Análisis y diseño de software

Báez Pérez, C. I. & Suárez Zarabanda, M. I. (2013). Proceso de desarrollo de software: basado en la articulación de RUP y CMMI priorizando su calidad.. Universidad de Boyacá. Capítulo 1 página 11 a 16

Omaña, M. (2012). Manufactura esbelta: una contribución para el desarrollo de software con calidad. Página de la 14 a la 18.