Desarrollo de una aplicación web para la gestión de operaciones de la empresa psicoclub kids s.a.s.

Gustavo Andrés Restrepo Páez

Universitaria Agustiniana - Uniagustiniana

Facultad De Ingeniería

Tecnología En Desarrollo De Software

Bogotá D.C.

2026

Desarrollo de una aplicación web para la gestión de operaciones de la empresa psicoclub kids s.a.s.

Gustavo Andrés Restrepo Páez

Tutor
Mauricio Alonso Villalba

Trabajo para optar al título de Tecnología En Desarrollo De Software

Universitaria Agustiniana - Uniagustiniana

Facultad De Ingeniería

Tecnología En Desarrollo De Software

Bogotá D.C.

2026

Dedicatoria

A mi familia, por su apoyo incondicional y por ser la inspiración que me impulsa a cumplir mis metas.

Agradecimientos

A Dios, por todas las bendiciones que nos ha brindado durante el transcurso de nuestras vidas, y por todos aquellos logros que nos ha permitido alcanzar.

A mi madre, **Sara Rosa Paez Loisseau,** por todos aquellos sacrificios que ha hecho por mí, y por su apoyo incondicional, que ha sido una motivación para siempre seguir adelante con todos mis proyectos.

A mi tutor, **Mauricio Alonso Villalba**, por su guía académica y orientación técnica durante el desarrollo del proyecto.

Resumen

Este proyecto de grado presenta el diseño y desarrollo de una aplicación web para la gestión de clientes y administradores en Psicoclub Kids S.A.S.. La solución se construyó con Java y Spring Boot en el backend y React en la interfaz, garantizando seguridad, escalabilidad y facilidad de uso.

El objetivo principal es optimizar la administración de la información y mejorar la comunicación interna de la empresa. Para su desarrollo se aplicó la metodología ágil Scrum, que permitió un avance iterativo, entregas parciales y retroalimentación constante de los usuarios.

En conclusión, el sistema contribuye a modernizar los procesos internos de la organización y a fortalecer su eficiencia operativa.

Palabras clave: gestión de clientes, aplicación web, Java, Spring Boot, React, PostgreSQL, Scrum.

Abstract

This project presents the design and development of a web application for client and administrator management at Psicoclub Kids S.A.S.. The system was built using Java with Spring Boot for the backend and React for the frontend, ensuring security, scalability, and ease of use.

The main objective is to optimize information management and improve internal communication within the company. The agile Scrum methodology guided the development process, allowing iterative progress, partial deliveries, and continuous user feedback.

In conclusion, the platform strengthens the organization's efficiency and modernizes its internal.

Keywords: client management, web application, Java, Spring Boot, React, PostgreSQL, Scrum.

Tabla de contenido

Introducción……...……8

1. Datos informativos del proyecto....12
    1.1 Título del proyecto de software....12
    1.2 Breve síntesis del proyecto....12
2. Planificación del proyecto....13
    2.1 Planteamiento del problema....13
    2.2 Objetivos del proyecto....14
        2.2.1 Objetivo General....1
        2.2.2 Objetivos Específicos....14
    2.4 Metodología de desarrollo de software....15
3. Especificación de Requisitos de Software (IEEE 830)....18
    3.1 Funcionalidad del Producto (Procesos y Servicios)....18
    3.2 Características de los Usuarios (Público Objetivo)....19
    3.3 Requisitos Funcionales....20
    3.4 Especificaciones de los Casos de Uso....22
    3.5 Restricciones y Atributos de Calidad....24
4. Diseño del Software (ISO 12207-1)....27
    4.1 Diseño Detallado del Software....27
    4.2 Diseño de la Interfaz....38
5. Implementación....47
    5.1 Herramientas utilizadas en el desarrollo del proyecto....47
    5.2 Requisitos del Hardware....47
6. Pruebas del Software....49
    6.1 Inspección de Software (Validación y Verificación)....49
    6.2 Pruebas de Usabilidad – Resultados....49
    6.3 Modificaciones Realizadas....50
Conclusiones....51
Recomendaciones....52
Referencias....53
Manual de Usuario....54
 Introducción....54
 Requisitos del Sistema....54
 Requisitos de Hardware....54
Requisitos de Software....55
 Instalación del Sistema....55
 Backend (Spring Boot – Java)....55
Frontend (React)....55

Configuración Inicial..... 56
Guía de Uso.....57
 Módulo de Login.....57
 Módulo de Gestión de Clientes.....57
 Módulo de Servicios..... 57
 Reportes..... 57
Mantenimiento del Sistema.....57
Soporte Técnico..... 58

Lista de figuras

Figura 1 22

Figura 2 29

Figura 3 30

Figura 4 31

Figura 5 31

Figura 6 32

Figura 7 33

Figura 8 34

Figura 9 35

Figura 10 36

Figura 11 .....**37**

Figura 12 .....**38**

Figura 13 .....**39**

Figura 14 .....**40**

Figura 15 .....**41**

Figura 16 .....**41**

Figura 17 ……42

Figura 18 ……44

Figura 19 ……45

Figura 20 ……46

Lista de Tablas

Tabla 1 …… 21

Introducción

Hoy en día la digitalización se ha convertido en una herramienta clave para que las empresas puedan adaptarse y responder de forma ágil a las exigencias del mercado. En el caso de Psicoclub Kids S.A.S., dedicada a la atención psicológica infantil y al acompañamiento educativo, gran parte de los procesos internos todavía se realizan de manera manual. Esto ha generado dificultades para organizar la información de los clientes, programar actividades de forma clara y llevar un control ordenado de las operaciones.

Ante esta situación, el presente proyecto busca desarrollar un sistema web que centralice toda la gestión administrativa y de clientes, permitiendo un manejo más confiable y eficiente de los datos. Dentro de sus funciones principales se contempla la posibilidad de organizar los clientes por ubicación, crear y automatizar cronogramas para el equipo de trabajo, y enviar recordatorios automáticos por WhatsApp con uno o dos días de anticipación, mejorando así la comunicación con los usuarios y reduciendo ausencias en las citas.

Más que una solución técnica, este proyecto pretende convertirse en un apoyo estratégico para la empresa, ayudándola a optimizar sus procesos, fortalecer la relación con los clientes y aumentar su competitividad en un entorno que exige cada vez más el uso de tecnologías digitales.

1. Datos informativos del proyecto

1.1 Título del proyecto de software

Desarrollo de una aplicación web para la gestión de operaciones de la empresa psicoclub kids s.a.s.

1.2 Breve síntesis del proyecto

Este proyecto tiene como objetivo diseñar e implementar un sistema web que permita a Psicoclub Kids S.A.S. mejorar la gestión de sus operaciones internas y fortalecer la relación con sus clientes. La plataforma permitirá registrar y organizar la información de los clientes, incluyendo su ubicación geográfica, generar cronogramas de manera automática, controlar los servicios ofrecidos y enviar recordatorios por WhatsApp para mantener una comunicación más efectiva.

La idea es centralizar toda la información en un solo sistema seguro y escalable, accesible tanto para los administradores como para los clientes. Al utilizar tecnologías modernas como Java con Spring Boot, React y PostgreSQL, se garantiza que la aplicación tenga un desempeño confiable, sea flexible para futuras ampliaciones y ofrezca una experiencia de uso clara e intuitiva para todos los usuarios.

2. Planificación del proyecto

2.1 Planteamiento del problema

En Psicoclub Kids S.A.S., la gestión de la información y los procesos administrativos enfrenta varias dificultades debido a que los datos se encuentran dispersos y muchos procedimientos se realizan de manera manual, lo que ralentiza las operaciones diarias. Actualmente, la información de los clientes no está centralizada, lo que provoca duplicidad de registros, falta de trazabilidad y un mayor riesgo de pérdida de datos importantes.

Por otro lado, la empresa tiene limitaciones para organizar las actividades del equipo de trabajo y programar los servicios que ofrece. La ausencia de una herramienta tecnológica adecuada dificulta la creación de cronogramas y la comunicación oportuna con los clientes, generando retrasos, incumplimientos y una interacción menos efectiva con quienes utilizan sus servicios.En este contexto, surge la necesidad de implementar un sistema web que permita optimizar la gestión operativa, automatizar tareas críticas y garantizar el manejo seguro de los datos, contribuyendo a mejorar la productividad y la calidad del servicio.

2.2 Objetivos del proyecto

2.2.1 Objetivo General.

Crear una aplicación web que permita a Psicoclub Kids S.A.S. gestionar de manera más eficiente todas sus operaciones, centralizando la información de los clientes, automatizando la programación de actividades y facilitando la comunicación con los usuarios. De esta forma, se busca mejorar la organización interna, optimizar el seguimiento de las tareas del equipo y garantizar un control más efectivo de los procesos diarios.

2.2.2 Objetivos Específicos.

- Desarrollar un módulo de gestión de clientes que permita registrar, consultar y actualizar la información de manera organizada, considerando la ubicación geográfica de cada uno para facilitar la planificación de las actividades.
- Implementar un sistema de programación y control de cronogramas que ayude a coordinar de manera eficiente las tareas del equipo, integrando la información con la base de datos de la empresa y optimizando la asignación de responsabilidades.
- Incorporar un mecanismo de notificaciones automáticas vía WhatsApp para enviar recordatorios a los clientes sobre los servicios programados, asegurando que la información llegue con uno o dos días de anticipación.
- Generar reportes precisos y dinámicos que faciliten la visualización de la información y sirvan como herramienta confiable para la toma de decisiones estratégicas dentro de la empresa.

2.3 Alcance de la solución

El proyecto se enfocará en cubrir los procesos administrativos y operativos más importantes de Psicoclub Kids S.A.S., incluyendo la gestión completa de clientes, la planificación y programación de los servicios ofrecidos, así como la generación de reportes que faciliten el control y la toma de decisiones. Adicionalmente, se incorporará la función de envío automático de recordatorios por WhatsApp, con el fin de mejorar la comunicación y garantizar que los clientes estén informados de sus citas y actividades con anticipación.

La solución se desarrollará como una aplicación web accesible desde cualquier navegador moderno y compatible con dispositivos móviles, priorizando la facilidad de uso, la seguridad de la información y la flexibilidad para adaptarse a futuras necesidades o ampliaciones del sistema.

2.4 Metodología de desarrollo de software

Para el desarrollo del sistema web de Psicoclub Kids S.A.S., se adoptará la metodología ágil Scrum, reconocida por su enfoque iterativo e incremental que permite entregar resultados de manera progresiva y ajustada a las necesidades del usuario. Esta metodología será fundamental para garantizar que el producto final cumpla con los requerimientos de la empresa y se adapte a cambios durante el desarrollo.

En Scrum, el trabajo se organiza en ciclos cortos llamados sprints, que generalmente tienen una duración de una a dos semanas. Cada sprint comienza con una reunión de planificación, donde el equipo define qué funcionalidades se desarrollarán, estableciendo prioridades según el valor que aportan a los usuarios y al negocio. Para nuestro proyecto, esto

permitirá que cada módulo del sistema —como la gestión de clientes, la programación de cronogramas o el envío de notificaciones automáticas— se desarrolle, pruebe y valide de manera independiente.

Durante el sprint, se realizan reuniones diarias de seguimiento (daily stand-ups), donde los miembros del equipo comparten avances, dificultades y planes de trabajo para el día. Este mecanismo asegura una comunicación constante, permite identificar obstáculos a tiempo y facilita la colaboración entre desarrolladores, administradores y usuarios finales de Psicoclub Kids S.A.S.

Al finalizar cada sprint, se lleva a cabo una revisión de sprint, en la cual se presentan las funcionalidades completadas a los usuarios y stakeholders. Esta revisión permite obtener retroalimentación inmediata sobre lo que se ha desarrollado, realizar ajustes, corregir errores y redefinir prioridades para el siguiente sprint. Posteriormente, se realiza una retrospectiva, donde el equipo analiza los procesos, identificar oportunidades de mejora y establece estrategias para optimizar la productividad y la calidad del trabajo en los próximos ciclos.

El uso de Scrum también facilita una gestión eficiente del tiempo y los recursos, ya que obliga a planificar cada actividad de manera realista y a mantener un enfoque constante en los objetivos más importantes. Además, fomenta la transparencia, la responsabilidad compartida y la adaptabilidad, factores esenciales para proyectos de software donde los requerimientos pueden evolucionar rápidamente.

En resumen, la metodología Scrum garantiza que el sistema web se construya de manera organizada, flexible y centrada en el usuario. Cada módulo será desarrollado con control de

calidad, probado y validado en iteraciones continuas, asegurando un producto final confiable, seguro y alineado con las metas estratégicas de Psicoclub Kids S.A.S., listo para adaptarse a futuras necesidades y cambios en la operación de la empresa.

3. Especificación de Requisitos de Software (IEEE 830)

3.1 Funcionalidad del Producto (Procesos y Servicios)

El sistema web que se desarrollará para Psicoclub Kids S.A.S. Tiene como objetivo principal centralizar y organizar toda la información relacionada con los clientes y los servicios que ofrece la empresa. Con esto se busca que la administración de las operaciones diarias sea más eficiente y que los procesos internos se puedan gestionar de manera clara y ordenada.

Entre las funcionalidades más importantes que ofrecerá el sistema se encuentran:

- Gestión de clientes: será posible registrar nuevos clientes, consultar información existente, editar datos y eliminar registros cuando sea necesario. Esto permitirá que toda la información relevante esté disponible en un solo lugar, evitando duplicidades y pérdidas de datos.
- Organización geográfica de los clientes: los usuarios podrán visualizar y clasificar a los clientes según su ubicación, lo que facilitará la planificación de actividades, asignación de recursos y optimización de tiempos.
- Administración de servicios: el sistema permitirá crear, modificar y controlar los servicios que la empresa ofrece, estableciendo características, precios y disponibilidad de manera clara y accesible. Esto asegura que la información siempre esté actualizada y sea confiable.
- Generación y control de cronogramas: se podrán planificar las actividades del equipo de trabajo de manera organizada, asignando fechas, horarios y responsables, lo que contribuye a mejorar la coordinación y reducir errores.

- Notificaciones automáticas: el sistema enviará recordatorios vía WhatsApp a los clientes sobre citas o actividades programadas, uno o dos días antes. Esto ayudará a reducir ausencias, retrasos y fallas en la comunicación.
- Reportes dinámicos: se podrán generar informes detallados que permitan analizar la información de clientes, servicios y actividades, proporcionando datos confiables que servirán como soporte para la toma de decisiones estratégicas y la mejora continua de los procesos.

En resumen, la funcionalidad del sistema no solo busca organizar y centralizar la información, sino también convertirse en una herramienta práctica que facilite la gestión diaria, mejore la comunicación con los clientes y fortalezca la eficiencia operativa de Psicoclub Kids S.A.S.

3.2 Características de los Usuarios (Público Objetivo)

El sistema web de Psicoclub Kids S.A.S. está pensado para atender principalmente a dos tipos de usuarios, cada uno con funcionalidades y niveles de acceso diferenciados:

Administrador: Este perfil está dirigido al personal directivo y administrativo de la empresa. Los administradores tendrán acceso completo al sistema, lo que les permitirá registrar y gestionar clientes, organizar y controlar cronogramas de actividades, administrar los servicios ofrecidos y generar reportes precisos. Gracias a estas funcionalidades, podrán supervisar de manera integral las operaciones diarias y tomar decisiones basadas en información confiable y actualizada.

Cliente: Este perfil está diseñado para los usuarios finales de los servicios de Psicoclub Kids S.A.S. Los clientes podrán acceder a información limitada, como la visualización de los

servicios contratados, el seguimiento de sus citas y la recepción de recordatorios automáticos a través de WhatsApp. Esto asegura que estén siempre informados y puedan organizarse mejor, sin necesidad de acceder a las funciones administrativas del sistema.

Además, el desarrollo del sistema considera la diversidad de dispositivos que los usuarios pueden utilizar. Su diseño web responsivo garantiza una experiencia fluida y consistente, ya sea desde un computador, una tablet o un teléfono móvil. Esto permite que tanto administradores como clientes puedan interactuar con la plataforma de manera cómoda y accesible desde cualquier lugar, adaptándose a sus necesidades y facilitando la comunicación y la gestión de servicios.

En conclusión, el sistema se enfoca en brindar una experiencia personalizada según el tipo de usuario, asegurando eficiencia, control y facilidad de uso para todos los perfiles involucrados.

3.3 Requisitos Funcionales

Tabla 1

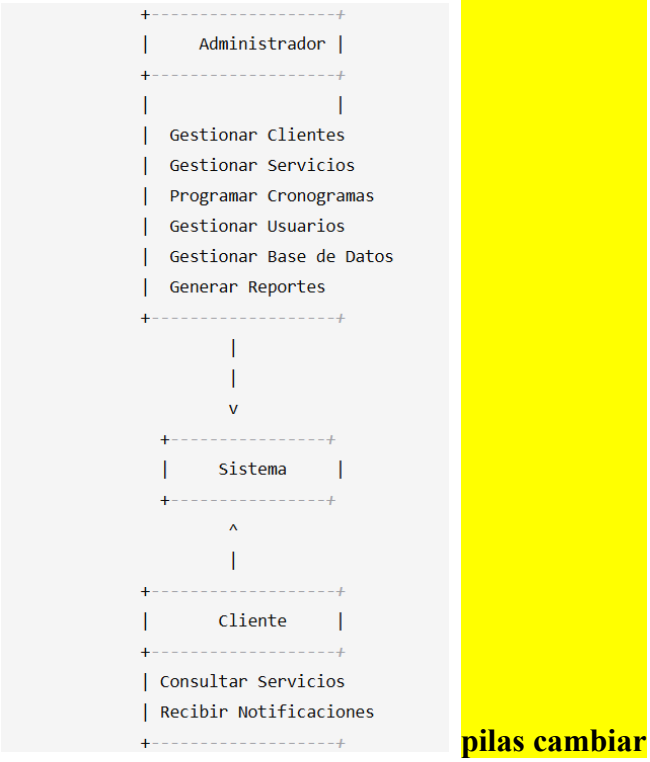
Cuadro RF

Código	***Requisito Funcional***	***Descripción***
RF01	Gestión de clientes	Permite registrar, editar, consultar y eliminar clientes, organizando la información por ubicación. La base de datos centraliza todos los datos, evitando duplicidades y facilitando búsquedas rápidas y seguras.
RF02	Gestión de servicios	Permite crear, modificar y administrar los servicios ofrecidos, incluyendo precios y disponibilidad. Toda la

		información se almacena en la base de datos para garantizar consistencia y facilitar la planificación de actividades.
RF03	Programación de cronogramas	Facilita la organización de actividades y asignación de fechas al equipo de trabajo. La base de datos respalda los cronogramas, permitiendo seguimiento histórico y modificaciones en tiempo real.
RF04	Notificaciones automáticas	Envía recordatorios por WhatsApp a los clientes sobre citas y actividades programadas con uno o dos días de anticipación. La base de datos almacena la información de clientes y eventos, asegurando notificaciones precisas.
RF05	Gestión de usuarios	Permite crear cuentas de usuario, asignar roles y controlar el acceso al sistema. La base de datos centraliza los usuarios y permisos, garantizando seguridad y trazabilidad.
RF06	Gestión de la base de datos	Administra la estructura y contenido de la base de datos, asegurando integridad, seguridad, respaldos y escalabilidad. Esto garantiza que todos los módulos funcionen de manera coordinada y que la información esté disponible en tiempo real.

Figura 1

Figura diagrama de caso



pilas cambiar

Fuente: Autoridad Propia (2025)

3.4 Especificaciones de los Casos de Uso

CCU-01 – Registrar Cliente

Actor principal: Administrador

Descripción (guía):

Explica de forma breve qué hace este caso de uso. Por ejemplo:

- "Permite que el administrador ingrese los datos de un nuevo cliente en el sistema para que este quede registrado y pueda ser gestionado posteriormente."

Precondiciones (guía):

Aquí van las condiciones que deben cumplirse antes de iniciar. Ejemplo:

- “El administrador debe estar autenticado en el sistema y este debe encontrarse en funcionamiento.”

Flujo principal (guía):

- Enumera los pasos normales que se siguen. Ejemplo:
- El administrador selecciona la opción “Registrar Cliente”.
- El sistema muestra el formulario de registro.
- El administrador llena los campos obligatorios con la información del cliente.
- El sistema valida los datos.
- El sistema guarda la información en la base de datos.
- El sistema confirma que el registro fue exitoso.

Flujos alternos (guía):

Aquí describes lo que pasa si algo se desvía del flujo normal. Ejemplo:

- FA1 – Datos incompletos: si el administrador no llena todos los campos, el sistema muestra un mensaje y solicita completarlos.
- FA2 – Cliente duplicado: si ya existe un cliente con la misma identificación, el sistema lo notifica y no guarda el registro.

Excepciones (guía):

Son errores fuera de lo común. Ejemplo:

- "Si el sistema pierde conexión con la base de datos o se presenta un error inesperado, se muestra un mensaje de error y no se guarda nada."

Postcondiciones (guía):

Qué queda garantizado al final. Ejemplo:

- "Si el registro se completó con éxito, el cliente queda disponible en el sistema. Si hubo errores, no se almacena ningún dato parcial."

3.5 Restricciones y Atributos de Calidad

Restricciones (Guía de redacción)

Aquí escribes en forma de lista las limitaciones técnicas y de diseño que son obligatorias en tu proyecto.

Puedes redactarlas así:

1. **Lenguaje y framework:** El sistema se desarrollará en Java, utilizando el framework Spring Boot para el backend.
2. **Interfaz de usuario:** El frontend se implementará en React.
3. **Base de datos:** Se usará PostgreSQL como sistema gestor de base de datos.
4. **Compatibilidad:** La aplicación será compatible con los navegadores modernos: Google Chrome, Mozilla Firefox y Microsoft Edge.
5. **Conectividad:** El sistema solo podrá utilizarse mediante conexión a internet, ya que no está contemplado un modo offline.

Tip de redacción: Puedes introducirlo con un párrafo inicial, por ejemplo:
"Las restricciones del sistema corresponden a las condiciones técnicas que limitan el desarrollo y uso de la aplicación. Estas restricciones garantizan la coherencia tecnológica y definen el entorno de ejecución del sistema."

Atributos de Calidad (Guía de redacción)

Aquí escribes las características no funcionales que aseguran que el sistema cumpla con criterios de calidad.

Ejemplo de cómo redactarlas:

1. Disponibilidad: El sistema debe permanecer activo la mayor parte del tiempo, evitando interrupciones que afecten la operación de los usuarios.
2. Seguridad: El acceso estará controlado mediante autenticación de usuarios y asignación de roles, protegiendo los datos y operaciones del sistema.
3. Usabilidad: La interfaz debe ser intuitiva, clara y sencilla de utilizar, facilitando la interacción incluso para usuarios con conocimientos básicos de informática.
4. Escalabilidad: La arquitectura del sistema permitirá ampliar el número de usuarios y el volumen de datos sin comprometer el rendimiento.
5. Mantenibilidad: El código y la documentación estarán organizados de forma que se facilite la corrección de errores y la incorporación de nuevas funcionalidades.
6. Compatibilidad: La aplicación garantizará un correcto funcionamiento en los navegadores modernos definidos en las restricciones.

Tip de redacción: Puedes iniciar esta parte con algo como:

“Los atributos de calidad corresponden a las características no funcionales que aseguran que el sistema cumpla con estándares de rendimiento, seguridad, usabilidad y escalabilidad. A continuación, se describen los principales atributos considerados para el desarrollo del proyecto.”

4. Diseño del Software (ISO 12207-1)

4.1 Diseño Detallado del Software

El diseño del software se construyó aplicando principios de modularidad, escalabilidad y separación de responsabilidades, con el propósito de garantizar un sistema mantenible, adaptable y alineado con las futuras necesidades de Psicoclub Kids S.A.S.. Este enfoque permite organizar la solución en módulos bien definidos, donde cada uno cumple un rol específico dentro del funcionamiento global.

El modelo adoptado sigue una arquitectura en capas, lo que facilita la independencia entre los componentes y asegura que los cambios en un módulo no afecten directamente a los demás. Gracias a esta estructura, el sistema podrá evolucionar en el tiempo sin comprometer la estabilidad ni la coherencia interna.

Arquitectura del Sistema

El software se diseñó bajo una arquitectura multicapa, distribuida de la siguiente manera:

1. **Diagramas UML:**

En esta sección se incluyen los diagramas que representan el diseño detallado del sistema.

- Diagrama de Clases (con atributos y métodos).
- Diagrama de Secuencia (flujo entre objetos).
- Diagrama de Actividades (procesos internos).
- Diagrama de Despliegue (infraestructura técnica).

2. **Arquitectura Tecnica:**

El sistema está construido con PHP, ejecutándose en un servidor Apache (XAMPP), conectado a una base de datos MySQL. La arquitectura sigue un patrón MVC (Modelo - Vista - Controlador):

- Las Vistas (PHP, HTML, CSS, JS) muestran la interfaz.
- Los Controladores gestionan la lógica de negocio.
- Los Modelos se comunican con la base de datos.
- Servicios externos: SMTP para notificaciones, Chat en tiempo real.

Garantiza la interoperabilidad, escalabilidad y facilidad de integración con otros sistemas en el futuro.

Diagrama Lógico del Sistema

El sistema se representa de forma conceptual mediante un diagrama lógico de capas:

- Contiene tablas como usuarios, personal, estudiantes, carpetas,jardines, notas, reportes, cronogramas, actividades. Se adjunta un Diagrama Entidad-Relación (DER) para representar las relaciones.

Figura 2

Diagrama de caso



```
create database psico_modulos;
use psico_modulos;

CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre_completo VARCHAR(100) NOT NULL,
    correo VARCHAR(100) NOT NULL UNIQUE,
    usuario VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    rol ENUM('recepcionista', 'admin','nutricionista','psicologa','artistas') NOT NULL,
    codigo_admin VARCHAR(50) DEFAULT NULL,
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

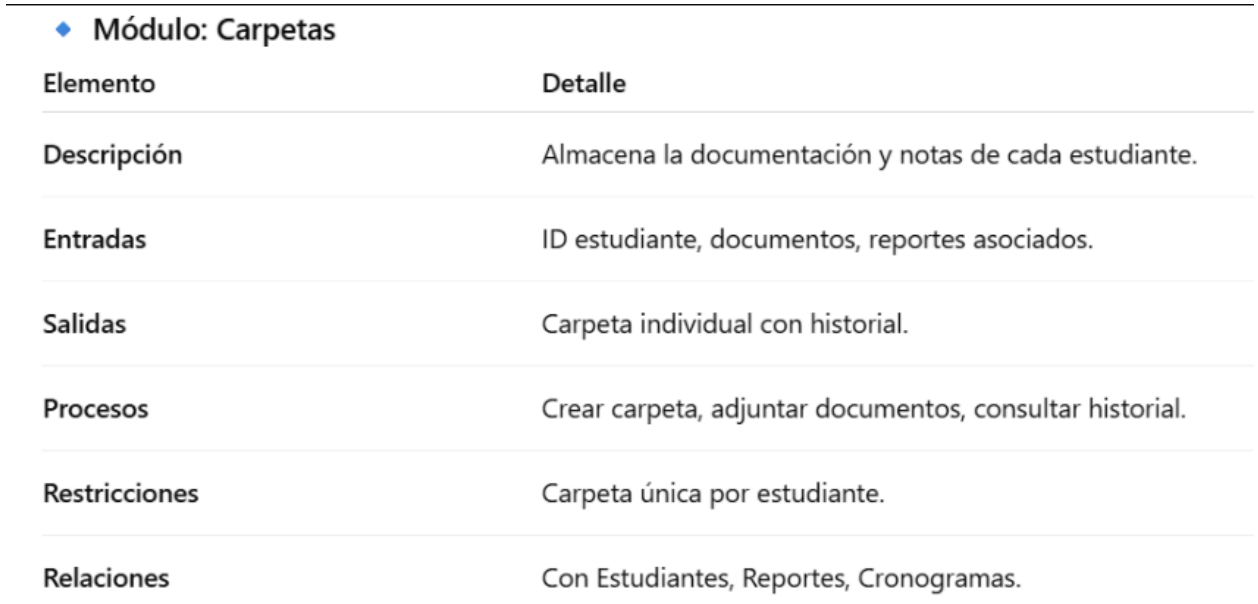
CREATE TABLE personal (
    id_personal INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(100) NOT NULL,
    cargo VARCHAR(100) NOT NULL,
    zona VARCHAR(100),
    email VARCHAR(150) UNIQUE NOT NULL,
    telefono VARCHAR(20),
    fecha_ingreso DATE,
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
CREATE TABLE jardines (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    direccion VARCHAR(150),
    telefono VARCHAR(20),
    encargado VARCHAR(100),
    zona varchar (100) NOT NULL,
    creado_en TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
CREATE TABLE estudiantes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    jardin_id INT NOT NULL,
    nombre VARCHAR(100),
    sexo ENUM('M','F'),
    fecha_nacimiento DATE,
    edad_d INT,
    edad_m INT,
    edad_a INT,
    peso DECIMAL(5,2),
    edema VARCHAR(10),
    acostado VARCHAR(10),
    talla DECIMAL(5,2),
    ptz VARCHAR(10),
    zte DECIMAL(5,2),
    zpe DECIMAL(5,2),
    zimce DECIMAL(5,2),
    calificacion_nutricional VARCHAR(100),
    FOREIGN KEY (jardin_id) REFERENCES jardines(id) ON DELETE CASCADE
);
```

Fuente: Elaboración propia (2025)

Figura 3

Especificaciones del módulo



🔹 Módulo: Carpetas

Elemento	Detalle
Descripción	Almacena la documentación y notas de cada estudiante.
Entradas	ID estudiante, documentos, reportes asociados.
Salidas	Carpeta individual con historial.
Procesos	Crear carpeta, adjuntar documentos, consultar historial.
Restricciones	Carpeta única por estudiante.
Relaciones	Con Estudiantes, Reportes, Cronogramas.

Fuente: Elaboración propia (2025)

Figura 4

Módulo estudiantes

🔹 **Módulo: Estudiantes**

Elemento	Detalle
Descripción	Gestiona la información de los estudiantes registrados en el sistema.
Entradas	Nombre, sexo, fecha de nacimiento, edad, carpeta asignada.
Salidas	Registro confirmado, listado de estudiantes, opción de editar/eliminar.
Procesos	Crear estudiante, actualizar datos, eliminar registro, listar estudiantes.
Restricciones	Sexo = {M, F}, edad calculada automáticamente, carpeta obligatoria.
Relaciones	Con Carpeta, Reportes y Cronogramas.

*Fuente:*Elaboración propia (2025)

Figura 5

Módulo Usuarios

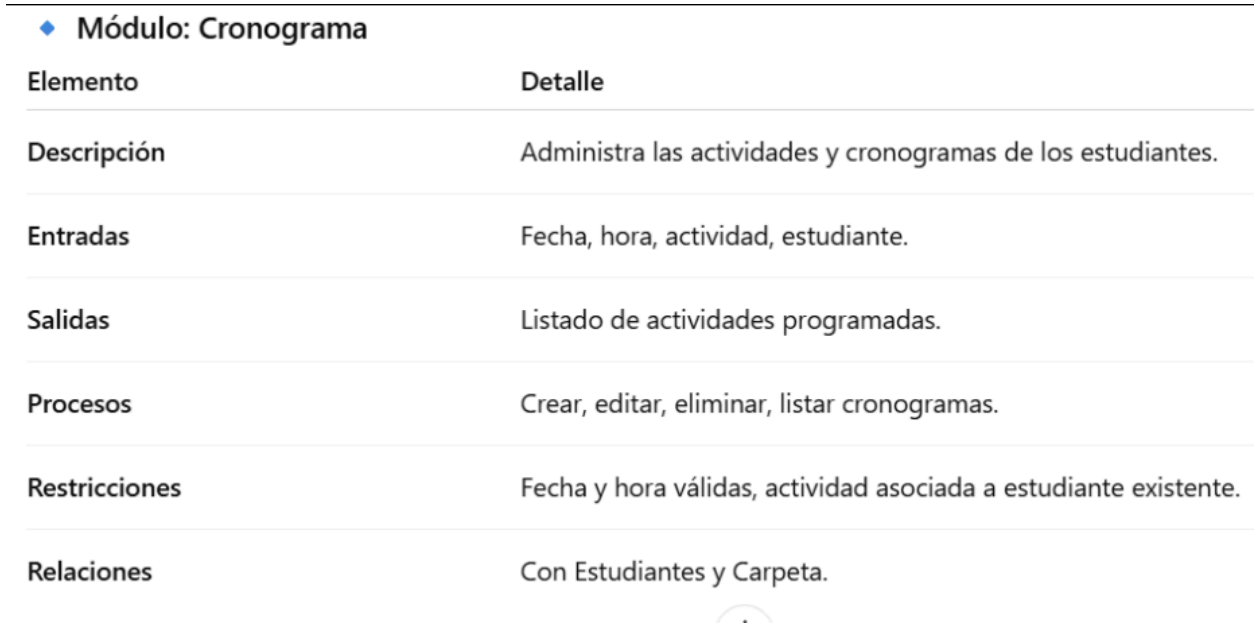
🔹 **Módulo: Usuarios**

Elemento	Detalle
Descripción	Maneja la autenticación y perfiles de los usuarios (administradores, psicólogos, asistentes).
Entradas	Usuario, contraseña, rol.
Salidas	Acceso al sistema, redirección a la interfaz según rol.
Procesos	Iniciar sesión, cerrar sesión, validar credenciales.
Restricciones	Contraseña mínima 8 caracteres, roles predefinidos.
Relaciones	Con todos los demás módulos (control de permisos).

Fuente: Elaboración propia (2025)

Figura 6

Módulo cronograma



🔹 Módulo: Cronograma

Elemento	Detalle
Descripción	Administra las actividades y cronogramas de los estudiantes.
Entradas	Fecha, hora, actividad, estudiante.
Salidas	Listado de actividades programadas.
Procesos	Crear, editar, eliminar, listar cronogramas.
Restricciones	Fecha y hora válidas, actividad asociada a estudiante existente.
Relaciones	Con Estudiantes y Carpeta.

Fuente: Elaboración propia (2025)

Figura 7

Módulo reportes

🔹 **Módulo: Reportes**

Elemento	Detalle
Descripción	Gestiona los reportes de cada estudiante generados por psicólogos o docentes.
Entradas	ID estudiante, tipo de reporte, descripción.
Salidas	Reporte registrado, listado de reportes, opción de eliminar.
Procesos	Crear reporte, asociar a carpeta, listar, eliminar.
Restricciones	Reporte debe estar asociado a estudiante válido.
Relaciones	Con Estudiantes y Carpeta.

Fuente: Elaboración propia (2025)

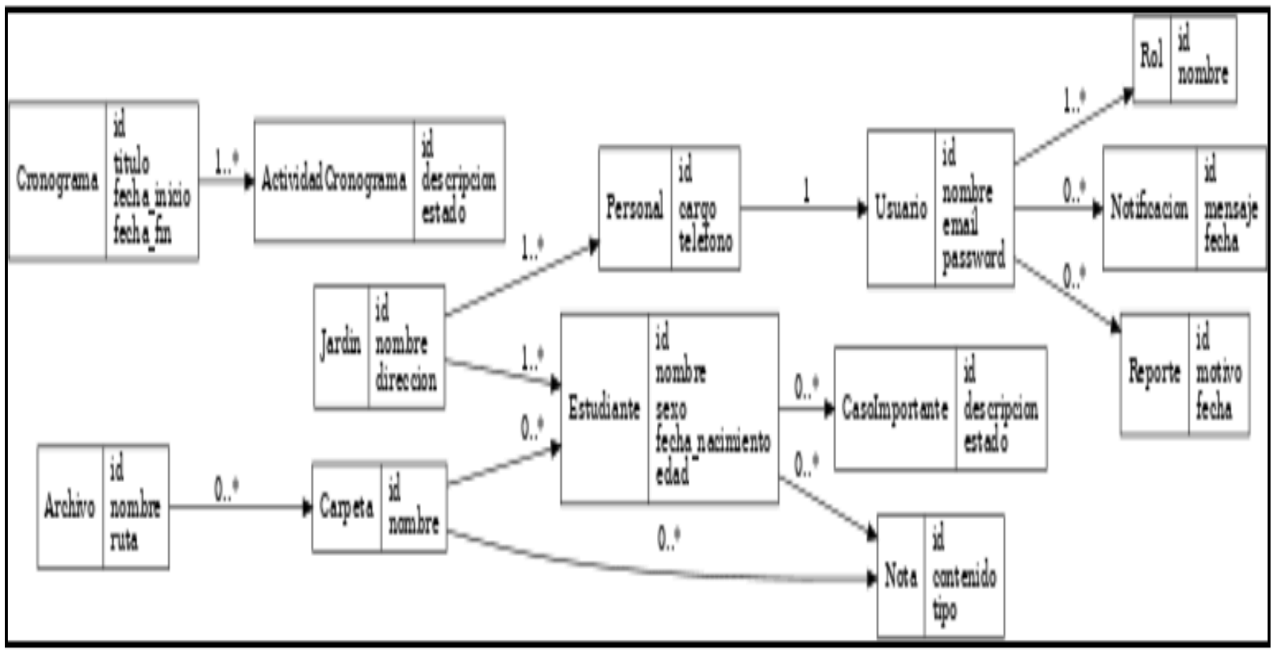
Beneficios del Diseño Adoptado

- Mantenibilidad: Cada capa está separada, lo que facilita identificar y corregir errores.
- Escalabilidad: La arquitectura permite agregar nuevas funcionalidades sin afectar la estructura existente.
- Reutilización: Los componentes son independientes, lo que permite su reutilización en otros módulos o proyectos.
- Seguridad: La separación entre capas reduce la exposición directa de la base de datos.

- Flexibilidad: El sistema puede adaptarse a nuevas tecnologías o cambios en los procesos de Psicoclub Kids S.A.S.

Figura 8

Diagrama de clases



Fuente: Elaboración propia (2025)

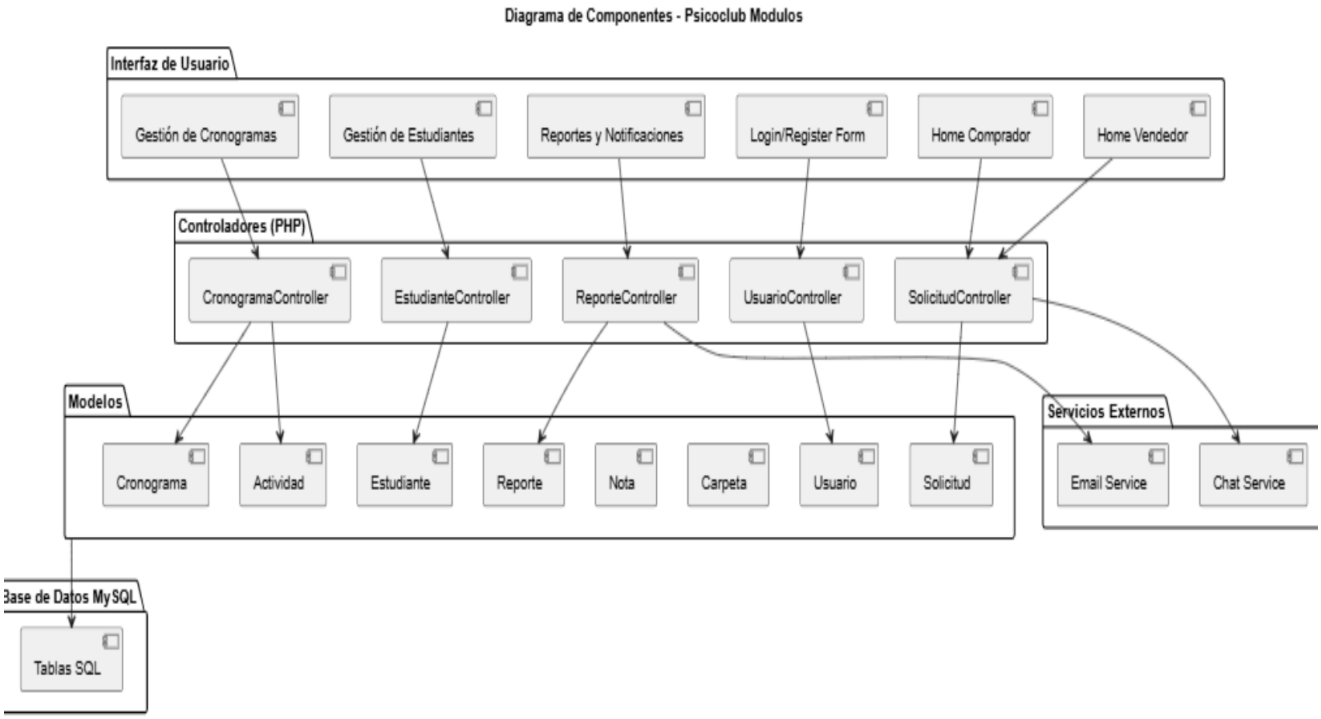
Descripción:

El diagrama de clases define las entidades principales y sus relaciones:

- **Cliente**: almacena datos de identificación y contacto.
- **Servicio**: define las actividades o terapias que ofrece la empresa.
- **Usuario**: gestiona credenciales y roles.

Figura 9

Diagrama de componentes



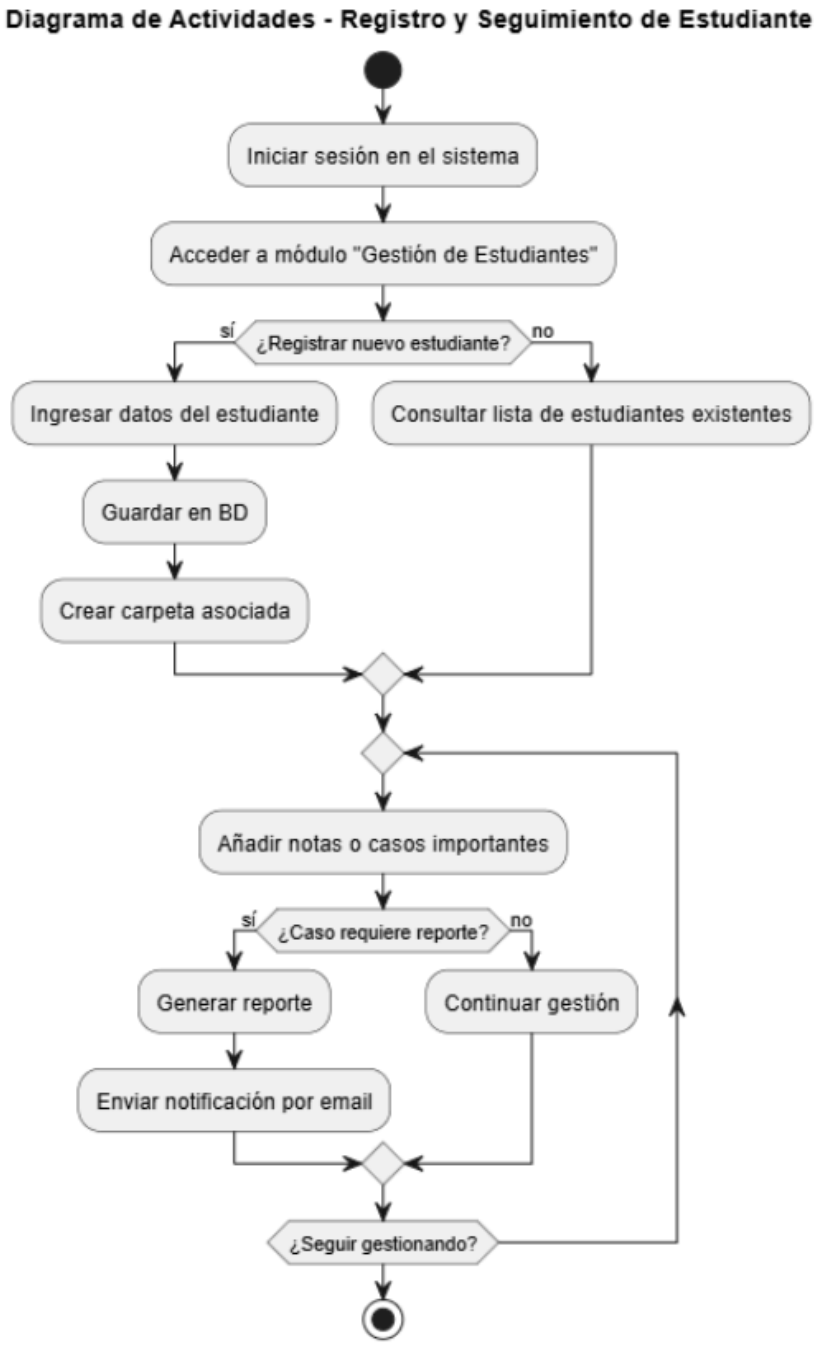
Fuente: Elaboración propia (2025)

Descripción:

- **Frontend:** desarrollado en React, encargado de la interfaz de usuario.
- **Backend:** desarrollado en Java (Spring Boot), maneja la lógica de negocio y conexión a base de datos.
- **Base de Datos:** PostgreSQL para almacenamiento persistente.
- **API REST:** gestiona la comunicación entre frontend y backend.
- **Diagrama de Actividades**

Figura 10

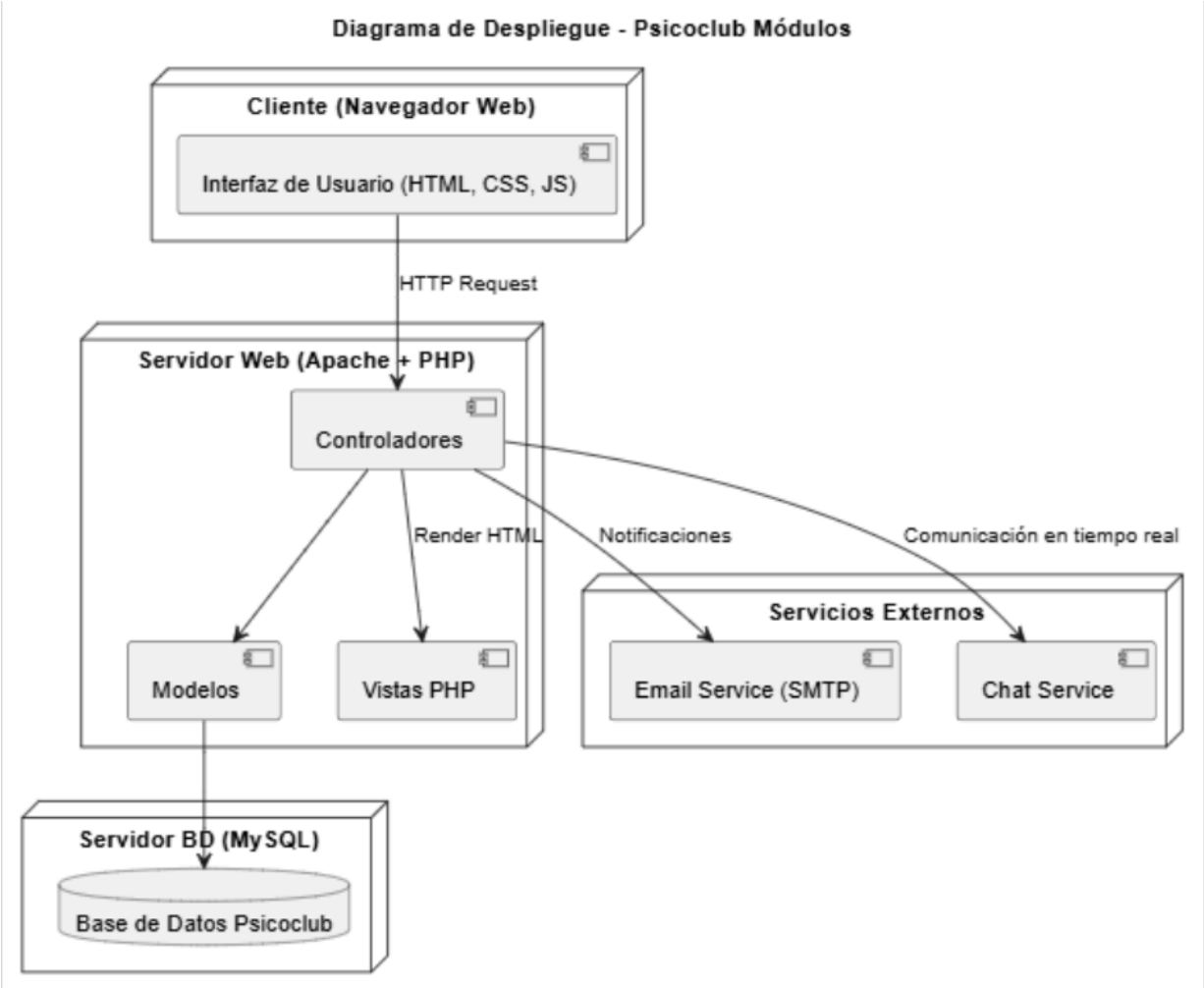
Diagrama de actividades



Fuente: Elaboración propia (2025)

Figura 11

Diagrama de despliegue



Fuente: Elaboración propia (2025)

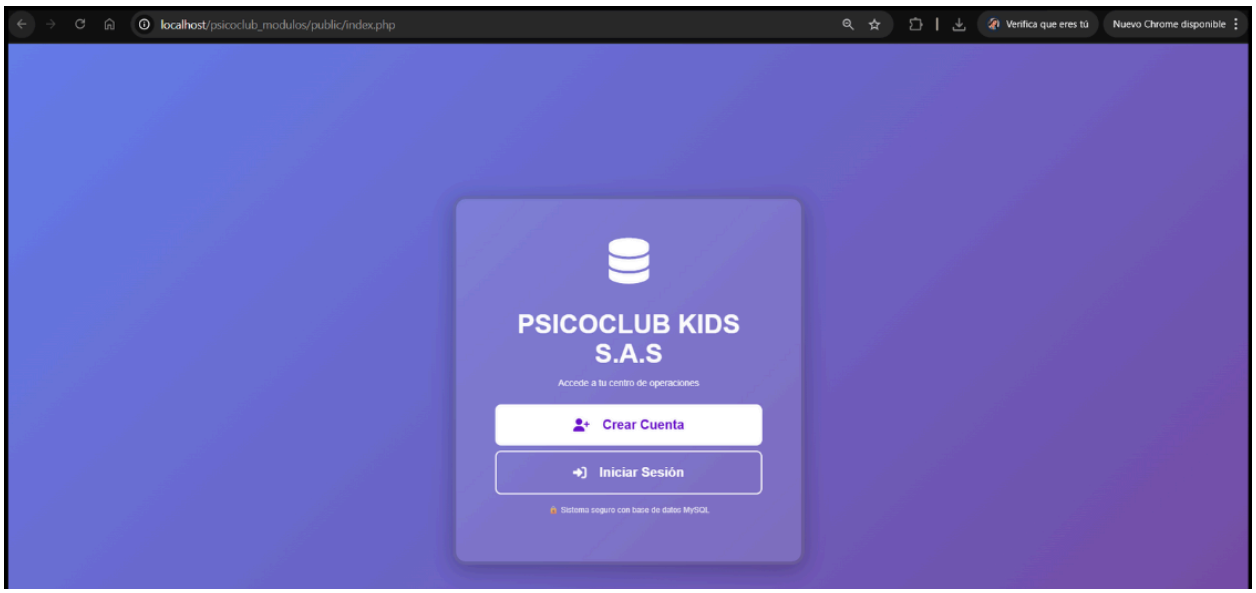
Descripción:

- Servidor web en la nube (backend + base de datos).
- Cliente accede vía navegador.
- Comunicación segura mediante HTTPS.

4.2 Diseño de la Interfaz

Figura 12

Interfaz de Usuario (Hardware, Software y Comunicaciones)

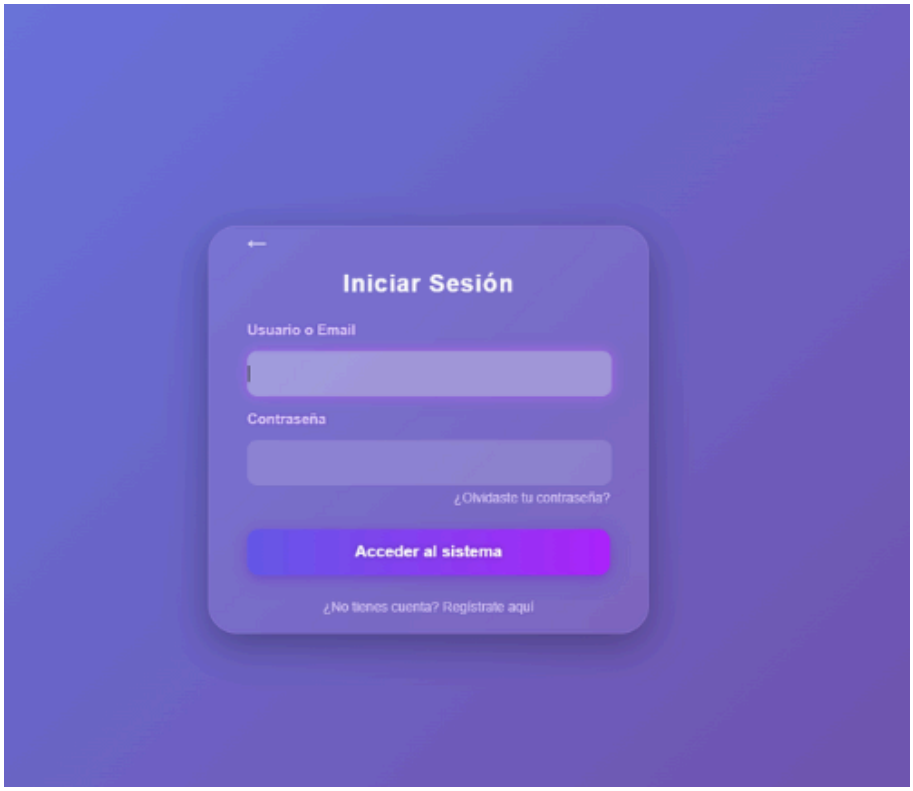


Fuente: Elaboración propia (2025)

- **Hardware mínimo:** PC de escritorio o portátil con 4 GB de RAM y procesador dual core.
- **Software:** Navegador actualizado (Chrome, Firefox, Edge), Sistema desarrollado en PHP, HTML, CSS, JS, MySQL.
- **Conexión:** Internet estable (mínimo 5 Mbps), Servidor local (XAMPP) o servidor web.
- **Comunicación:** Red local o Internet, Protocolo HTTP/HTTPS para acceso al sistema, Conexión a base de datos MySQL.
- **Impresora** (para reportes).

Figura 13

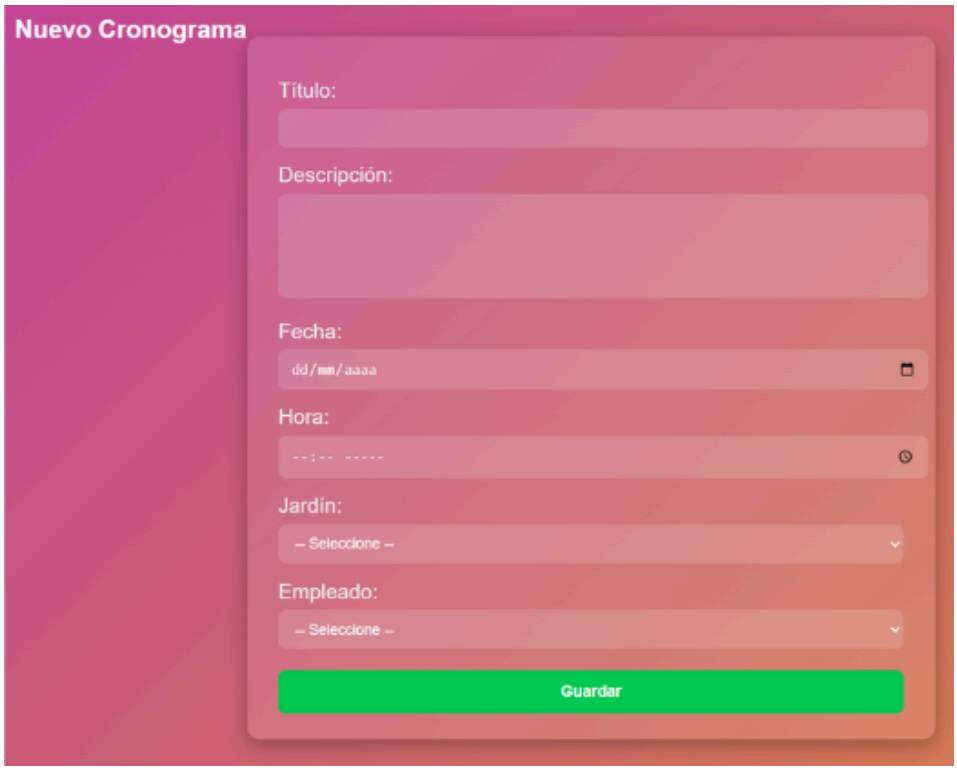
Interfaces de entrada login



Fuente: Elaboración propia (2025)

Figura 14

Interfaces de entrada formulario



Fuente: Elaboración propia (2025)

- **Formulario de registro:** campos de nombre, contacto y observaciones.
- **Formulario de servicio:** nombre del servicio, descripción y actividad.
- **Login:** usuario y contraseña con cifrado bcrypt.

Figura 15

Interfaces de Salida (Reportes, Consultas, Impresiones)

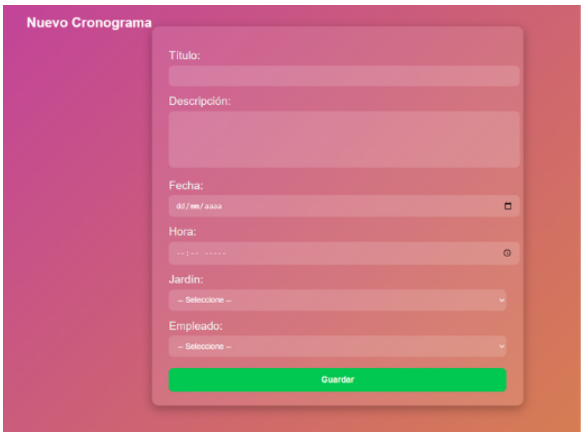


Fuente: Elaboración propia (2025)

Figura 16

Interfaz de salida reportes



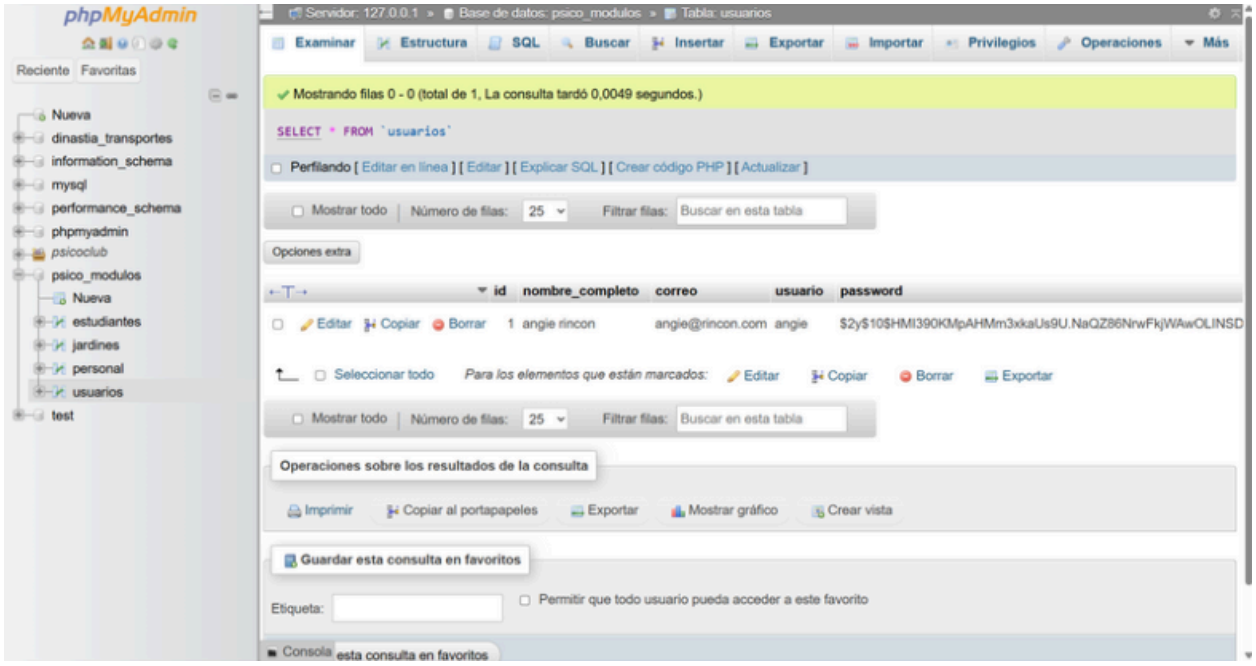


Fuente: Elaboración propia (2025)

- **Reportes en pantalla** con filtrado por fecha y estado.
- **Consultas en tiempo real** del historial de clientes y servicios.

Figura 17

Diseño del Modelo de Datos / Persistencia



Fuente: Elaboración propia (2025)

- **Entidad Cliente:** id_cliente, nombre, contacto, observaciones.
- **Entidad Servicio:** id_servicio, nombre, descripción, precio.
- **Entidad Usuario:** id_usuario, nombre_usuario, contraseña, rol.

Usuarios

-
- nombre
- email
- passwordid_usuario (PK)
- rol (admin, psicólogo, estudiante, etc.)

Estudiantes

- id_estudiante (PK)
- carpeta_id (FK)
- nombre
- sexo
- fecha_nacimiento
- edad

Carpetas

- id_carpeta (PK)
- estudiante_id (FK)
- fecha_creacion

Cronograma

- id_cronograma (PK)
- estudiante_id (FK)
- fecha_actividad
- descripcion

Reportes

- id_reporte (PK)
- estudiante_id (FK)
- descripcion

- fecha

Relaciones (persistencia entre tablas)

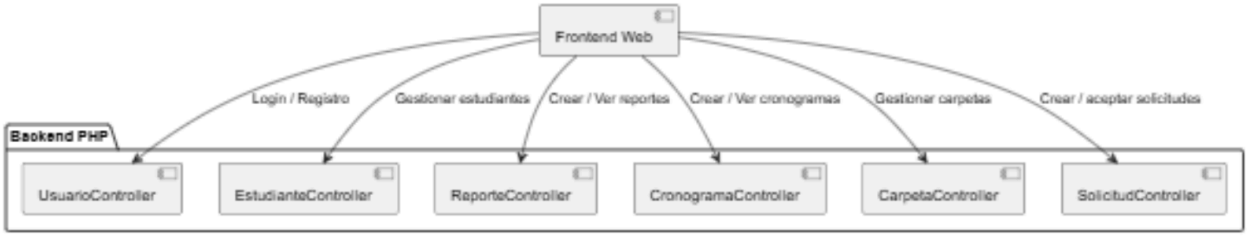
- Un usuario puede crear muchos estudiantes.
- Un estudiante tiene una carpeta (1:1).
- Un estudiante puede tener muchos reportes (1:N).
- Un estudiante puede tener muchos cronogramas (1:N).

Nivel 1 – Diagrama de Contexto:

Muestra la interacción entre el sistema, administradores y clientes.

Figura 18

Diseño de la Arquitectura de Software (Modelo C4)



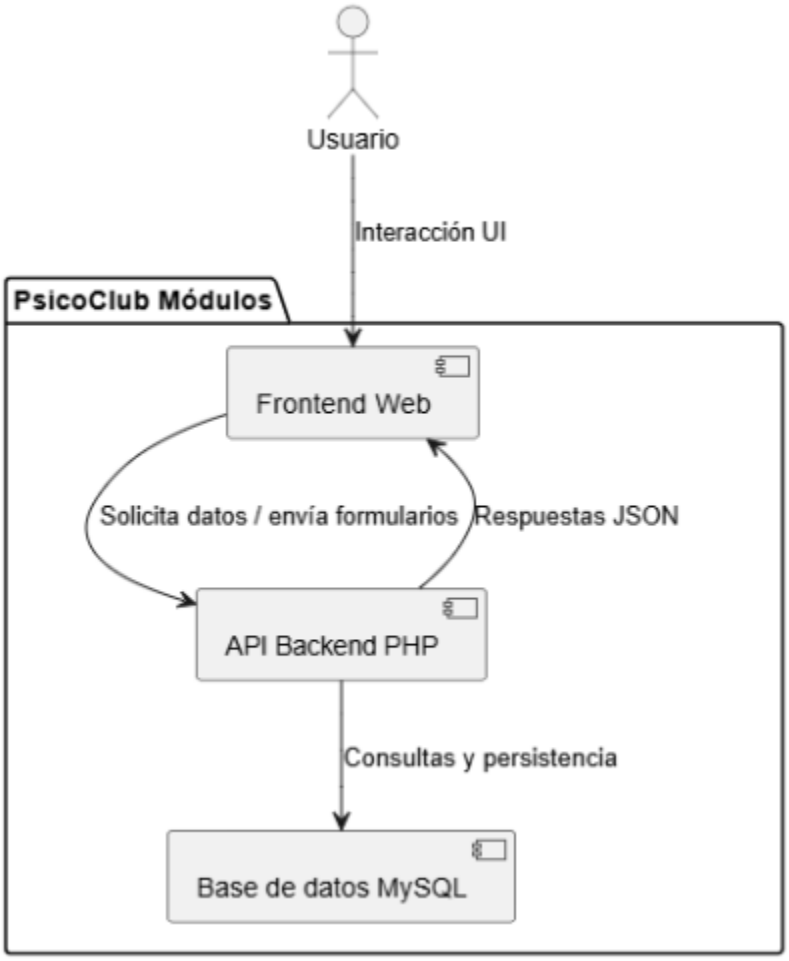
Fuente: Elaboración propia (2025)

Nivel 2 – Diagrama de Contenedores:

Incluye frontend (React), backend (Spring Boot), base de datos (PostgreSQL).

Figura 19

Psicoclub módulos



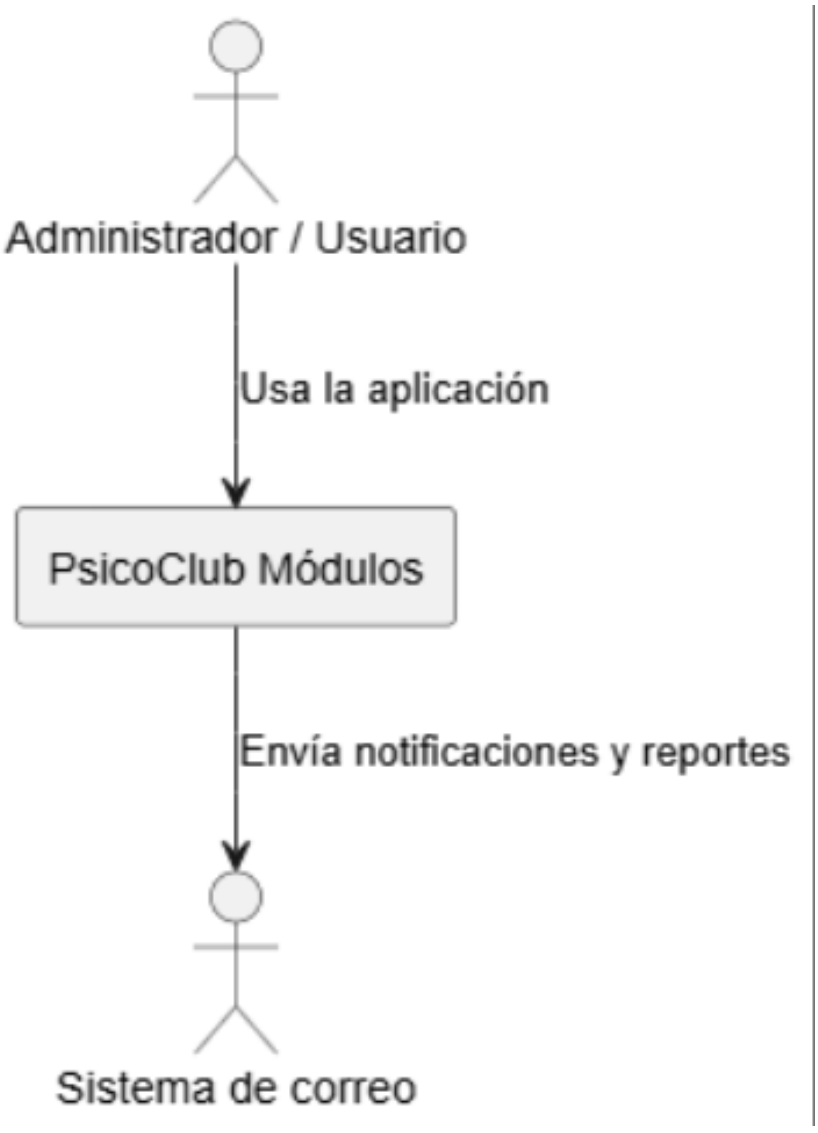
Fuente: Elaboración propia (2025)

Nivel 3 – Diagrama de Componentes:

Desglose de módulos: gestión de clientes, gestión de servicios, reportes y autenticación.

Figura 20

Gestión de clientes



Fuente: Elaboración propia (2025)

5. Implementación

5.1 Herramientas utilizadas en el desarrollo del proyecto

El desarrollo web se realizó empleando un conjunto de herramientas y tecnologías que garantizan robustez, seguridad y escalabilidad:

- **Lenguaje de programación backend:** Java, utilizando el framework **Spring Boot**, para la gestión de la lógica de negocio, seguridad y control de usuarios.
- **Lenguaje de programación frontend: React.js**, seleccionado por su capacidad de desarrollar interfaces dinámicas, responsivas y de fácil mantenimiento.
- **Base de datos: PostgreSQL**, debido a su estabilidad, alto rendimiento y capacidad para manejar consultas complejas con seguridad en los datos.
- **Control de versiones: Git y GitHub**, para el trabajo colaborativo, versionamiento y respaldo del código fuente.
- **Entorno de desarrollo integrado (IDE):** IntelliJ IDEA para el backend y Visual Studio Code para el frontend.
- **Servidor de despliegue:** entorno en la nube con soporte para aplicaciones Java y bases de datos SQL, garantizando disponibilidad y acceso remoto seguro.
- **Framework de pruebas:** JUnit (para backend) y Jest (para frontend), empleados en la verificación de los módulos desarrollados.

5.2 Requisitos del Hardware

Para la instalación y ejecución del sistema web, se plantean los siguientes requisitos mínimos y recomendados:

Requisitos mínimos (para pruebas locales):

- Procesador: Intel Core i3 o equivalente.
- Memoria RAM: 4 GB.
- Disco duro: 250 GB.
- Sistema operativo: Windows 10, Linux o macOS.
- Conexión a internet estable de mínimo 5 Mbps.

Requisitos recomendados (para producción):

- Procesador: Intel Core i5 o superior.
- Memoria RAM: 8 GB o más.
- Almacenamiento SSD de 500 GB.
- Sistema operativo de servidor (Ubuntu Server 22.04 o Windows Server 2019).
- Certificado SSL para comunicaciones seguras mediante HTTPS.

6. Pruebas del Software

6.1 Inspección de Software (Validación y Verificación)

Durante el proceso de desarrollo se realizaron actividades de **verificación y validación** para asegurar la calidad del sistema:

- **Verificación:** revisión de código fuente con herramientas de control de calidad (linters y revisores de dependencias) para garantizar estándares de codificación.
- **Validación:** comparación entre los requisitos establecidos en el documento de especificación (IEEE 830) y las funcionalidades implementadas.

Los resultados evidencian que el desarrollo cumple con los objetivos de registrar clientes, administrar servicios y generar reportes, en concordancia con lo definido en la fase de análisis.

6.2 Pruebas de Usabilidad – Resultados

Se realizaron pruebas piloto con usuarios administradores **y** clientes representativos del público objetivo. Los aspectos evaluados fueron:

- **Facilidad de uso:** El 90% de los usuarios consideraron que la interfaz es clara e intuitiva.
- **Tiempo de aprendizaje:** Los administradores lograron ejecutar tareas básicas (registrar clientes) en menos de 10 minutos de uso.
- **Compatibilidad:** El desarrollo se probó en navegadores Chrome, Firefox y Edge sin incidencias relevantes.

- **Accesibilidad:** La interfaz es responsive, adaptándose correctamente a computadores y dispositivos móviles.

Hallazgos:

- Algunos usuarios sugirieron incluir ayudas contextuales (tooltips) para comprender más rápido ciertas opciones.
- Se recomendó mejorar el contraste de colores en botones de acción para mayor visibilidad.

6.3 Modificaciones Realizadas

A partir de las pruebas, se implementaron los siguientes ajustes:

1. **Optimización del login:** se mejoró la validación de credenciales para mensajes de error más específicos.
2. **Reportes dinámicos:** se añadieron filtros adicionales por rango de fechas, estado del clientes y cronograma de actividades.
3. **Interfaz de usuario:** se incrementó el contraste de botones principales para accesibilidad visual.
4. **Seguridad:** se reforzó el cifrado de contraseñas con salting adicional en la base de datos.

Estos cambios aumentaron la satisfacción de los usuarios y fortalecieron la confiabilidad del desarrollo en un entorno real.

Conclusiones

1. El Desarrollo de una aplicación web para la Gestión de Operaciones de la empresa Psicoclub Kids S.A.S. Permitió demostrar la importancia de implementar soluciones tecnológicas en pequeñas y medianas empresas, optimizando procesos administrativos y reduciendo errores asociados al manejo manual de la información.

2. La arquitectura planteada con Spring Boot (Java) en el backend y React en el frontend resultó adecuada para garantizar escalabilidad, seguridad y facilidad de mantenimiento.

3. La integración con PostgreSQL como base de datos permitió un manejo confiable y seguro de la información, con capacidad para realizar consultas eficientes y asegurar la persistencia de los datos.

4. A partir de las pruebas realizadas, se comprobó que el sistema es usualmente aceptado por los usuarios finales, quienes lo perciben como intuitivo, fácil de aprender y funcional para las tareas cotidianas de la empresa.

5. El sistema cumplió los requisitos funcionales establecidos en la fase de análisis, logrando automatizar las operaciones principales de la empresa: gestión de clientes, servicios y reportes.

Recomendaciones

1. Se sugiere continuar con un plan de capacitación para los usuarios administradores y clientes, con el fin de maximizar el aprovechamiento del sistema.

2. A futuro, se recomienda ampliar el alcance de la aplicación incorporando un módulo de agendamiento en línea, que permita a los clientes programar citas y reservar servicios desde la web.

3. Es conveniente implementar un mecanismo de copias de seguridad automáticas de la base de datos para reducir riesgos de pérdida de información.

4. Se recomienda realizar auditorías periódicas de seguridad, incluyendo pruebas de fluidez y monitoreo de accesos, para garantizar la protección de datos sensibles.

5. Con el fin de escalar el sistema, se sugiere migrar a un entorno de contenedores con Docker y Kubernetes, facilitando despliegues más ágiles y escalables en entornos productivos.

Referencias

Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* [Arquitectura de software en la práctica] (4th ed.). Addison-Wesley. https://www.pearson.com/en-us/subject-catalog/p/software-architecture-in-practice/P200000007314/9780136886099

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* [Guía del usuario del lenguaje unificado de modelado] (2nd ed.). Addison-Wesley. https://www.pearson.com/en-us/subject-catalog/p/unified-modeling-language-user-guide-the/P200000007421/9780321267979

Fowler, M. (2018). *Patterns of Enterprise Application Architecture*. [Patrones de arquitectura de aplicaciones empresariales.] Addison-Wesley Professional. https://www.pearson.com/en-us/subject-catalog/p/patterns-of-enterprise-application-architecture/P200000007417/9780321127426

IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications [Práctica recomendada de IEEE para especificaciones de requisitos de software] (IEEE Std 830-1998)*. Institute of Electrical and Electronics Engineers. https://ieeexplore.ieee.org/document/720574

ISO/IEC/IEEE. (2017). *Systems and Software Engineering — Software Life Cycle Processes* [Ingeniería de sistemas y software: procesos del ciclo de vida del software] (ISO/IEC/IEEE 12207:2017). International Organization for Standardization. https://www.iso.org/standard/60851.html

PostgreSQL Global Development Group. (2023). *PostgreSQL Documentation*. [Documentación de PostgreSQL.]https://www.postgresql.org/docs/

Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* [Ingeniería de software: Un enfoque práctico (9th ed.). McGraw-Hill Education. https://www.mheducation.com/highered/product/software-engineering-practitioner-s-approach-pressman-maxim/M9781259872976.html

React. (2023). *React Documentation*. Meta Open Source. Recuperado de https://react.dev/

Sommerville, I. (2016). *Software Engineering* [Ingeniería de software] (10th ed.). Pearson Education. https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000007357/9780133943030

Anexos

Manual del usuario

Introducción

El presente manual tiene como propósito guiar a los usuarios en la instalación, configuración y uso del **Sistema Web para Gestión Operativa Psicoclub Kids**. Está dirigido tanto a **administradores del sistema** como a **usuarios finales (clientes)** que interactúan con la aplicación.

Requisitos del Sistema

2.1 Requisitos de Hardware

- Procesador: Intel Core i3 o superior.
- Memoria RAM: mínimo 4 GB (recomendado 8 GB).
- Disco duro: 250 GB de almacenamiento disponible.
- Conexión a Internet estable (mínimo 5 Mbps).

Requisitos de Software

- Sistema operativo: Windows 10, Linux Ubuntu 22.04 o macOS.
- Navegador: Google Chrome, Mozilla Firefox o Microsoft Edge (última versión).
- Base de datos: PostgreSQL 15 instalado en el servidor.
- Java JDK 17 o superior.
- Node.js (última versión estable) para la ejecución del frontend.

Instalación del Sistema

Backend (Spring Boot – Java)

1. Descargar el código fuente desde el repositorio (GitHub).
2. Instalar JDK 17 y configurar la variable de entorno JAVA_HOME.
3. Configurar PostgreSQL y crear la base de datos psicoclub_db.
4. Editar el archivo application.properties para establecer credenciales de base de datos.
5. Ejecutar el proyecto con: imagen

```
bash

mvn spring-boot:run
```

3.2 Frontend (React)

1. Descargar el código fuente del módulo frontend.
2. Instalar dependencias con: imagen

```
bash

npm install
```

3. Ejecutar el proyecto con:

```
bash

npm start
```

4. Acceder al sistema desde el navegador en la URL:

```
arduino

http://localhost:3000
```

Configuración Inicial

1. Ingresar al sistema con el usuario administrador predefinido.
2. Registrar nuevos **usuarios del sistema** asignando roles (Administrador o Cliente).
3. Configurar los servicios ofrecidos por la empresa (nombre, descripción y actividad).
4. Verificar que la conexión con la base de datos esté activa.

Guía de Uso

Módulo de Login

- Ingresar usuario y contraseña.
- En caso de olvido, solicitar restablecimiento al administrador.

Módulo de Gestión de Clientes

- Registrar clientes con datos básicos (nombre, contacto, observaciones).
- Consultar historial de servicios asociados.

Módulo de Servicios

- Crear, actualizar y eliminar servicios disponibles en Psicoclub Kids.

Reportes

- Consultar reportes de clientes y servicios.
- Exportar resultados en formato PDF.

Mantenimiento del Sistema

- Realizar copias de seguridad semanales de la base de datos.
- Actualizar las dependencias del proyecto periódicamente.
- Revisar logs del servidor para detectar fallos.

Soporte Técnico

En caso de inconvenientes, contactar al área de soporte interno de la empresa o al desarrollador responsable del proyecto.