

LabGestor - Sistema de documentación para un laboratorio

Desarrollado por:



David Gonzalez



Andrea Niño

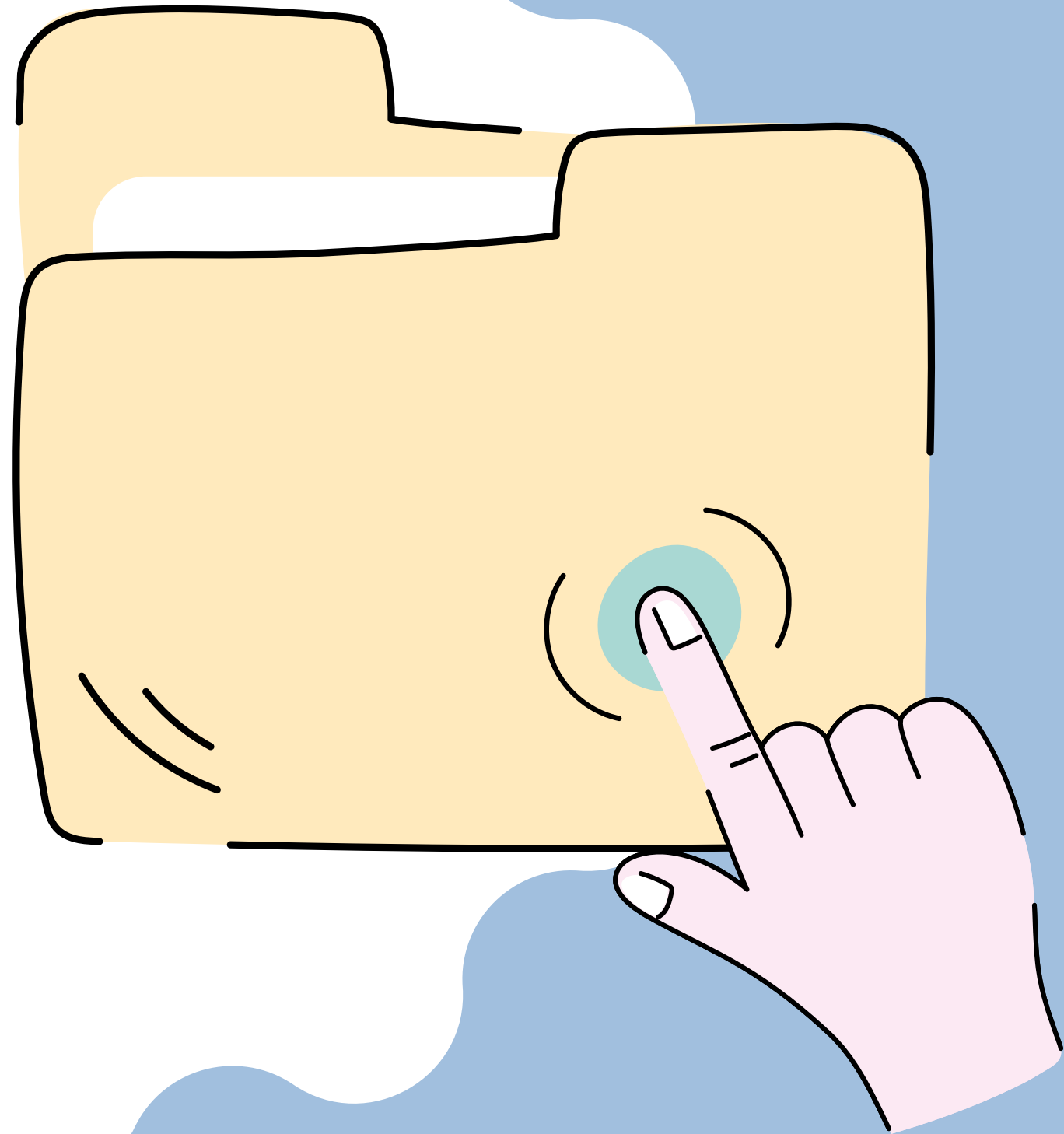


Nicolás Becerra



Santiago Quitian

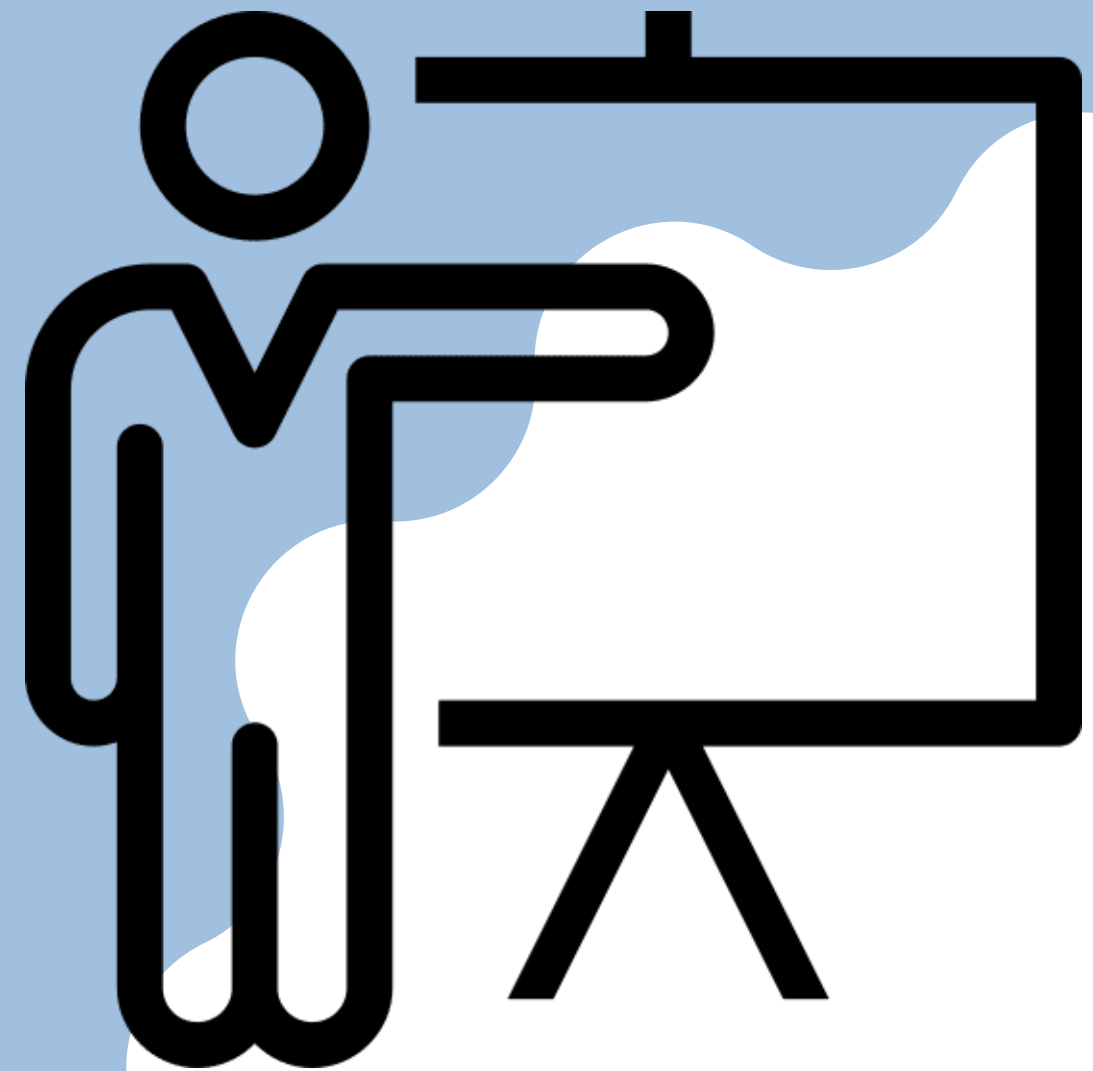
Contenidos



- Introducción
- Objetivo general
- Objetivos específicos
- Alcance y Roles
- Fases de vida del software
- Link de la plataforma
- Costos

Introducción

El sistema de información "LabGestor" se ha diseñado para automatizar la documentación que en la actualidad se ingresa del forma manual en el laboratorio de microbiología de la empresa VITECO S.A.



Objetivo general

El objetivo principal del sistema es digitalizar y automatizar las tareas que actualmente se registran de manera manual y obstaculizan el flujo de trabajo del área.

Objetivos específicos



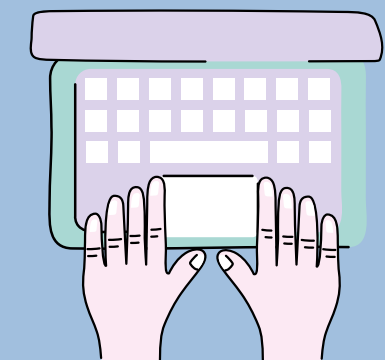
Crear módulos de software que faciliten la documentación de procedimientos estándar para el análisis



Elaborar una base de datos exhaustiva que almacene información sobre registros, procesos, usuarios y análisis de productos.




Diseñar una interfaz de usuario intuitiva que facilite la navegación y el uso del software.



Construir casos de uso precisos, requisitos funcionales, no funcionales y clases bien estructuradas.

Alcance y Roles

El sistema de información estará dirigido exclusivamente a los usuarios del área de microbiología, el sistema tendrá 2 roles (Administrador y Analista) contará con 4 módulos principales los cuales son modulo de usuarios e inicio de sesión, módulo de clientes, módulo de fabricantes y módulo de productos el cual contiene sub módulos.



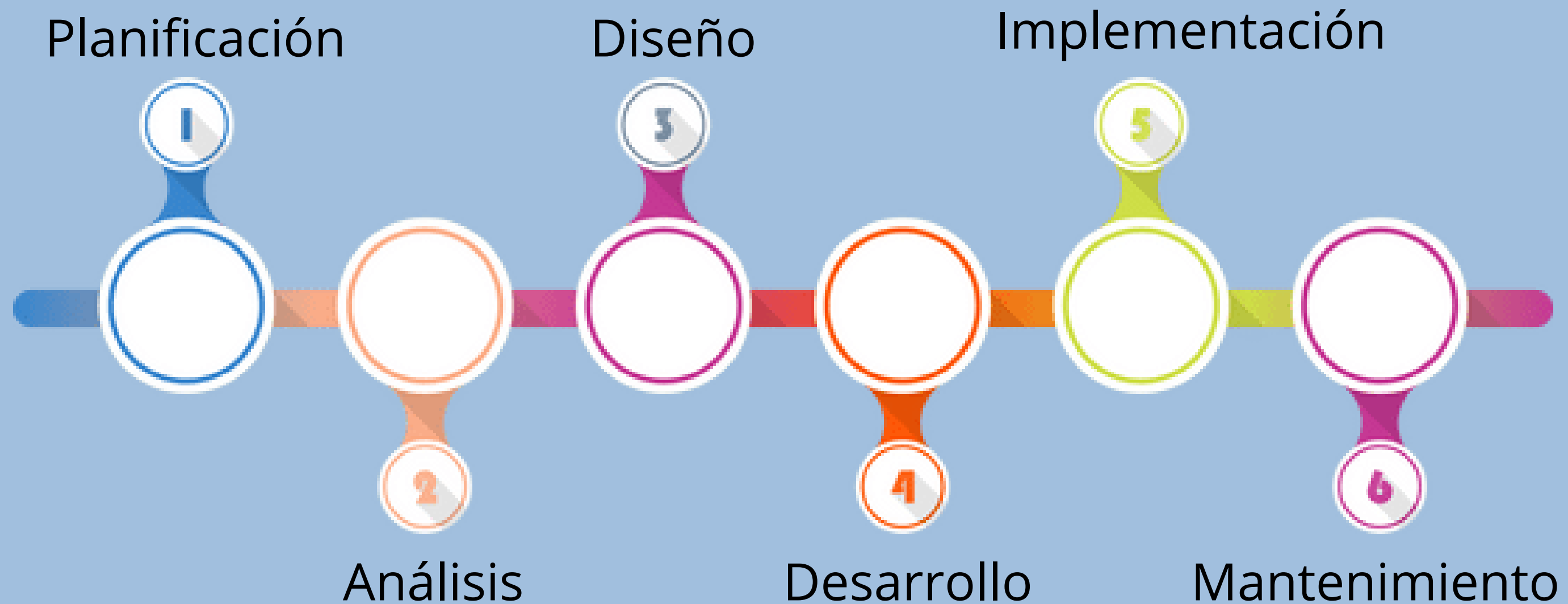
Analista



Administrador

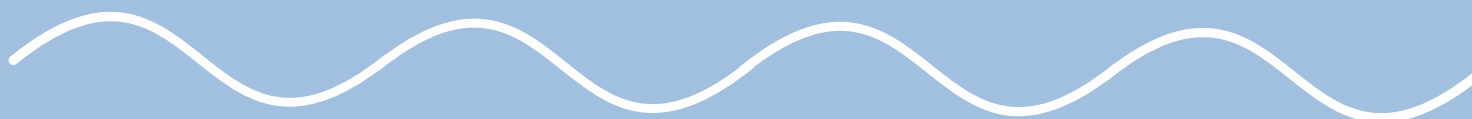


Ciclo de vida del software



Planificación

En esta fase inicial, se realizó un planteamiento del problema exhaustivo, se definieron los objetivos del proyecto, alcance, propuesta técnica y una sólida Justificación del proyecto.



Planteamiento del problema

El laboratorio de microbiología de la empresa VITECO S.A. enfrenta desafíos significativos en la gestión y eficiencia de sus procesos de documentación. Actualmente, tareas cruciales como el diligenciamiento de bitácoras de monitoreo se realizan de forma manual, lo que impacta negativamente en la eficiencia del área. Este enfoque manual genera retrasos y aumenta la probabilidad de errores.

Además, el laboratorio experimenta dificultades en el control de las funciones del personal, lo que afecta el orden y la responsabilidad en los procesos internos. También se ha identificado una desconexión entre el laboratorio y el inventario de la empresa, lo que puede llevar a incertidumbre y reprocesos adicionales.

Análisis

En la fase de Análisis, se realizaron entrevistas a 2 personas de la empresa que permitieron identificar flujos de trabajo. Esto se plasmo en requerimientos funcionales y no funcionales detallados, y la definición de las Reglas de Negocio, sirviendo como base para los primeros Mockups visuales del sistema.

Requisitos Funcionales (RF)

Los requisitos funcionales describen las funciones y servicios específicos que el sistema LabGestor debe proporcionar a sus usuarios. Están organizados por módulos para una mejor comprensión.

Módulo de Registro e Ingreso de Usuarios


Este módulo gestiona todo lo relacionado con la administración y autenticación de los usuarios del sistema.

- RF-USU-001: Registrar Nuevo Usuario: El sistema debe permitir el registro de un nuevo usuario solicitando Cédula, Nombres, Apellidos, Correo electrónico y Rol.
- RF-USU-002: Validar Formato de Datos de Registro: El sistema debe verificar que los datos ingresados durante el registro cumplan con los formatos esperados.



Fase de Diseño

La fase de diseño se centró en la ilustrados en diagramas necesarios como el MER, ER, diagrama de clases, casos de uso entre otros. Se diseñó la base de datos en PostgreSQL. Se eligió la arquitectura para el backend y para el frontend.



Bienvenido

Ingresa tus credenciales

Nombre de Usuario

Contraseña

☐ Mostrar contraseña

[¿Olvidaste tu Contraseña?](#)

Diagrama ER

El diagrama ER representa el diagrama Mer de una manera mas visual y entendible, muestra todas las tablas que contiene la base de datos la cual fue creada en PostgreSQL

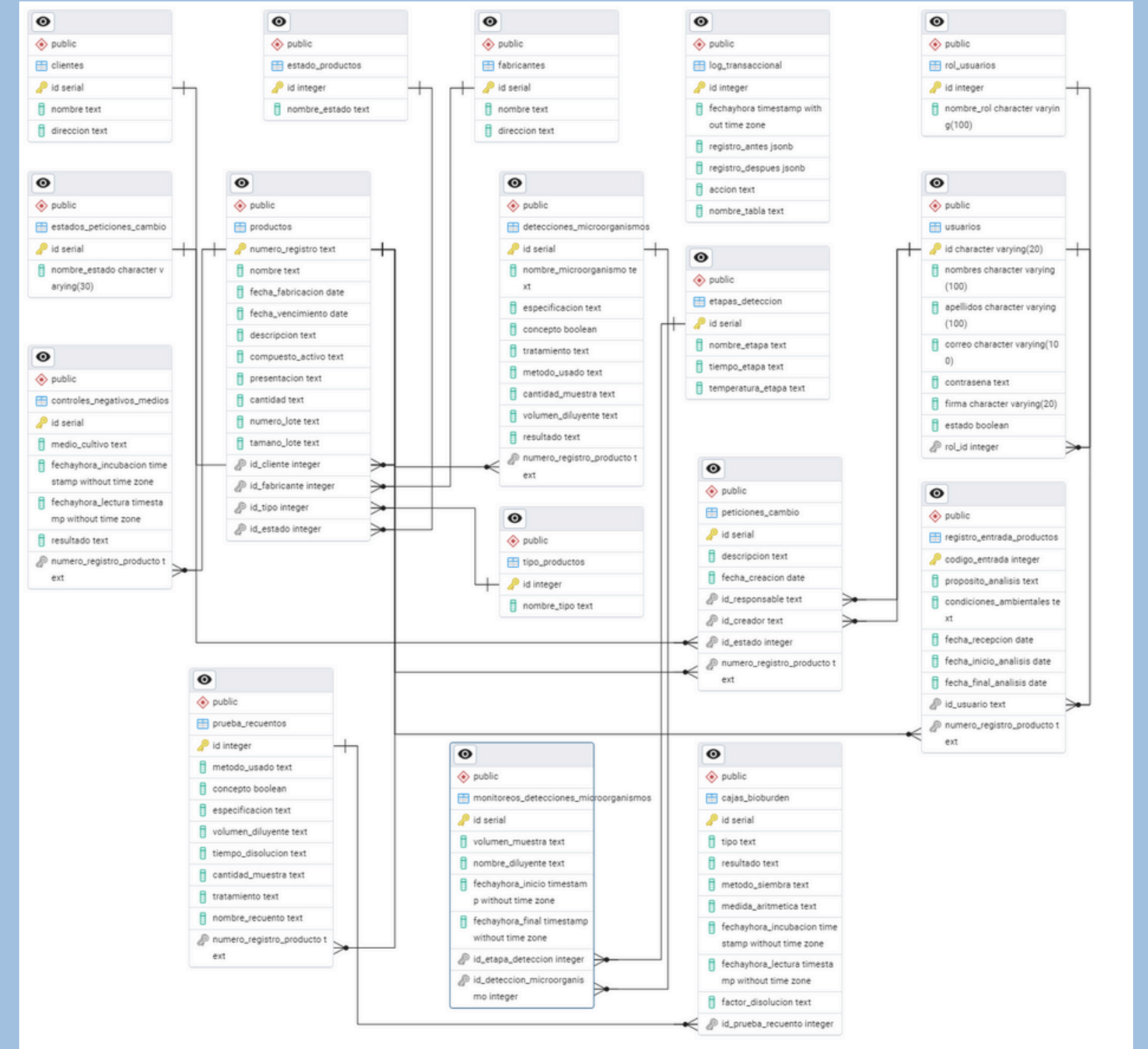
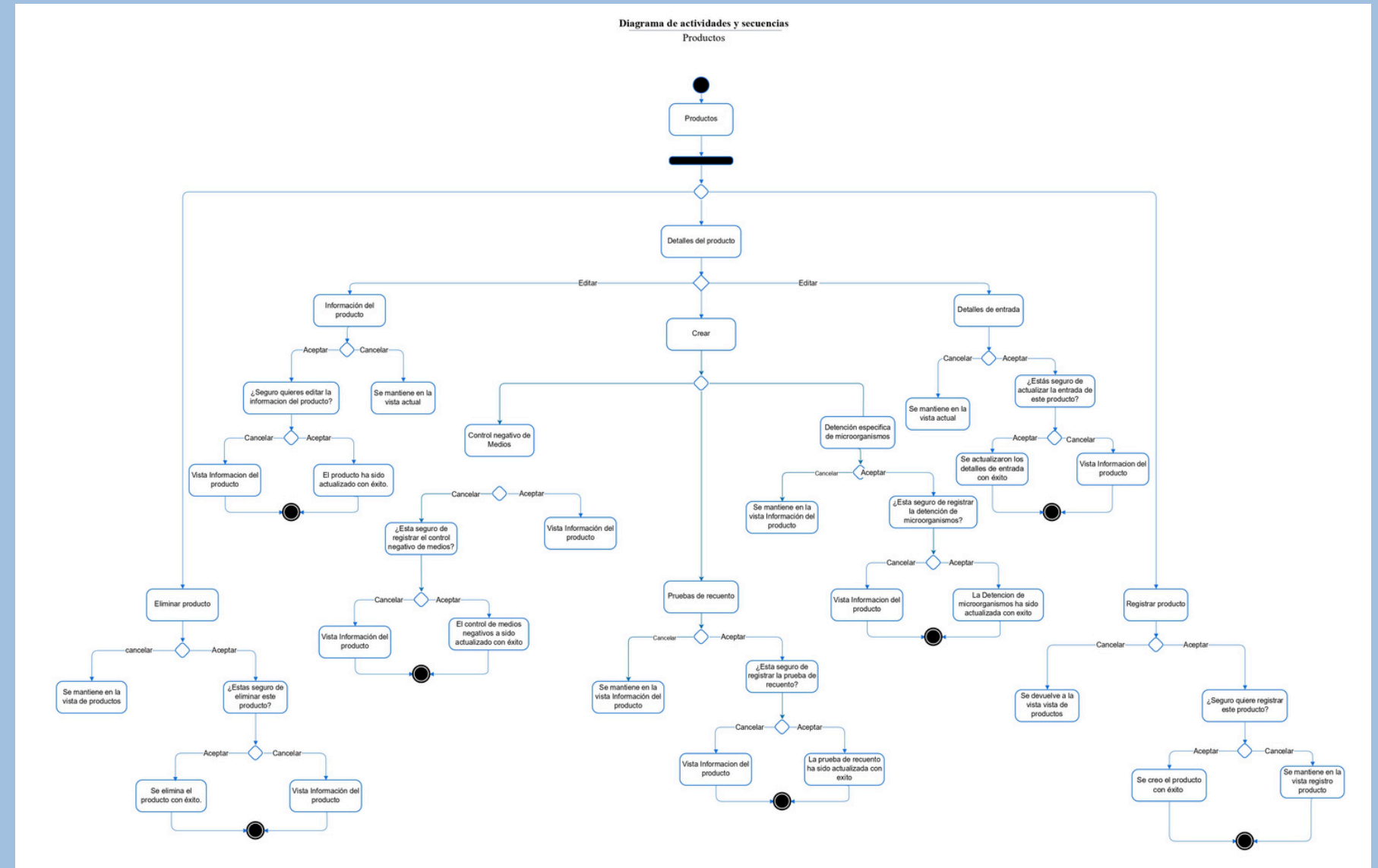


Diagrama de actividades

Muestra la respuesta del sistema dependiendo de la decisión que el usuario toma y permite entender el flujo del sistema de una manera mas clara.



Casos de uso

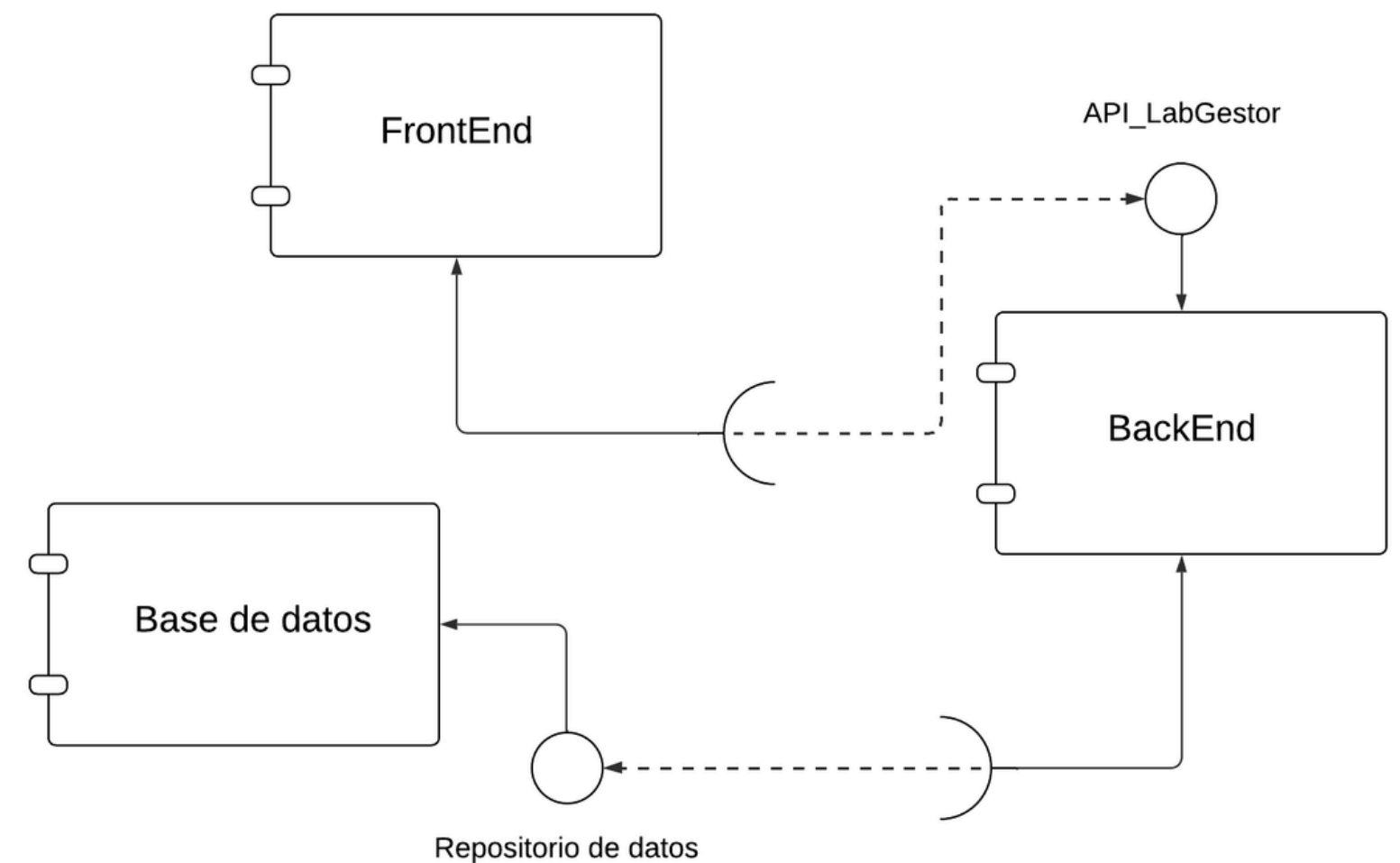
Permite visualizar las interacciones entre los usuarios (actores) y el sistema, mostrando qué puede hacer el sistema desde la perspectiva del usuario.



Diagrama de componentes

Representa la estructura física y lógica de un sistema de software y muestra cómo estos componentes se comunican entre si, siendo útil para entender la arquitectura del software.

Diagrama de Componentes LabGestor



Fase de Desarrollo

- Arquitectura
- Front End
- Back End
- PostgreSQL



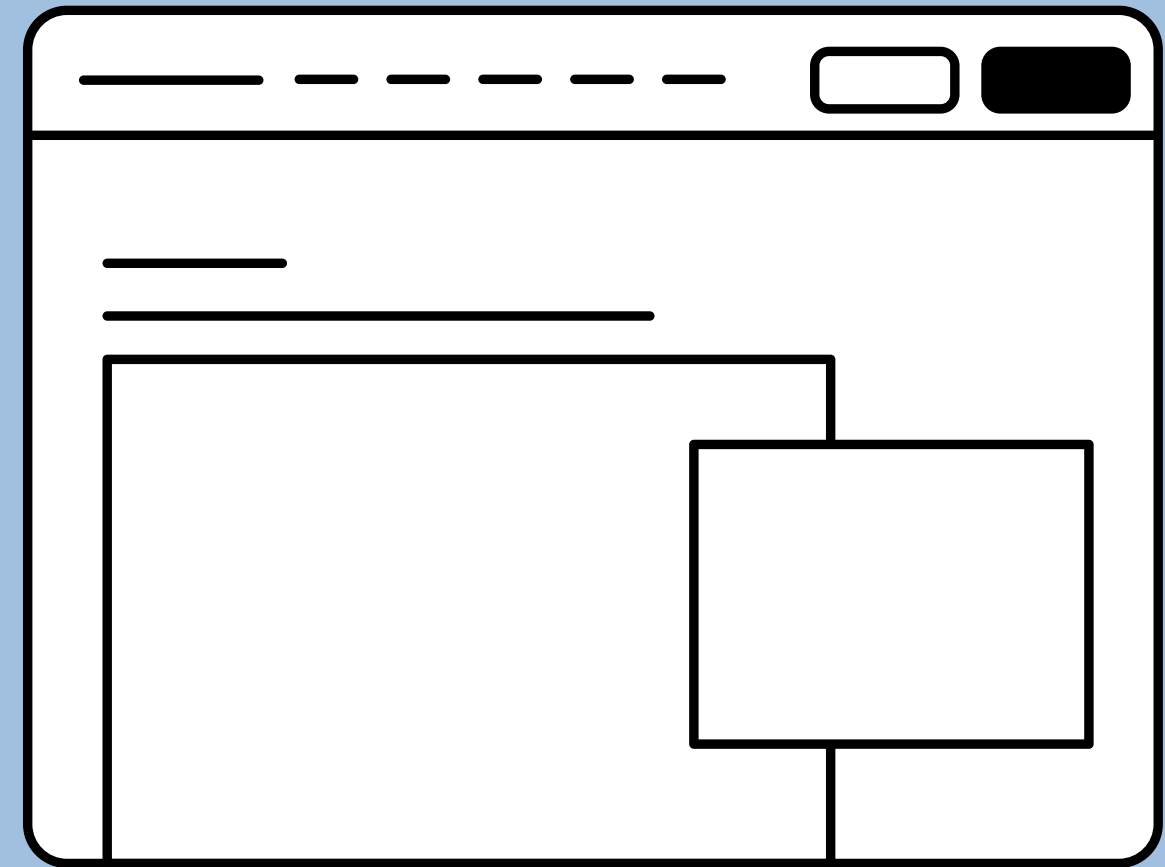
Arquitectura

Para el backend se uso una arquitectura Clean, para el frontend se uso una arquitectura por feature o módulos y para la arquitectura del sistema en general una arquitectura Cliente-Servidor



FrontEnd

Para el frontend se uso Astro,
JavaScript, React entre otras
tecnologías como AdGrid,
SweetAlert2 y React-Hook-
Form



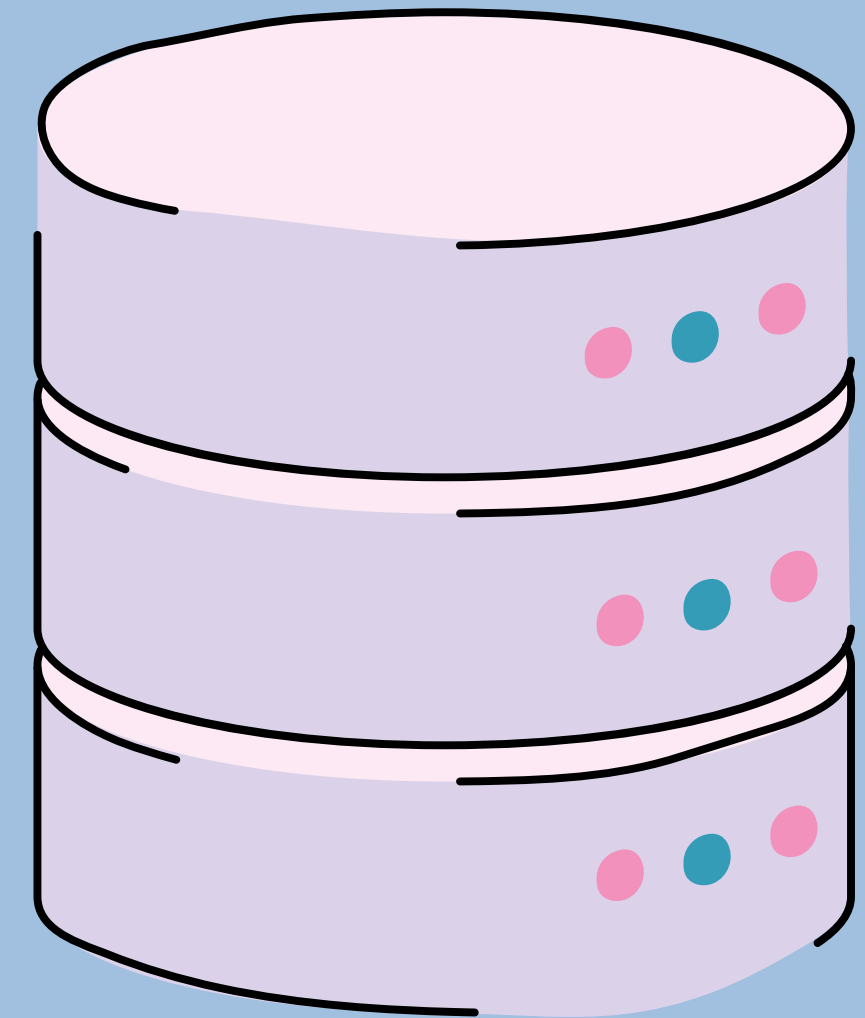
BackEnd

Para el Backend se uso Go como lenguaje, echo para simplificar el proceso de vinculación de cargas útiles de solicitudes HTTP y GORM para conectar la base de datos.



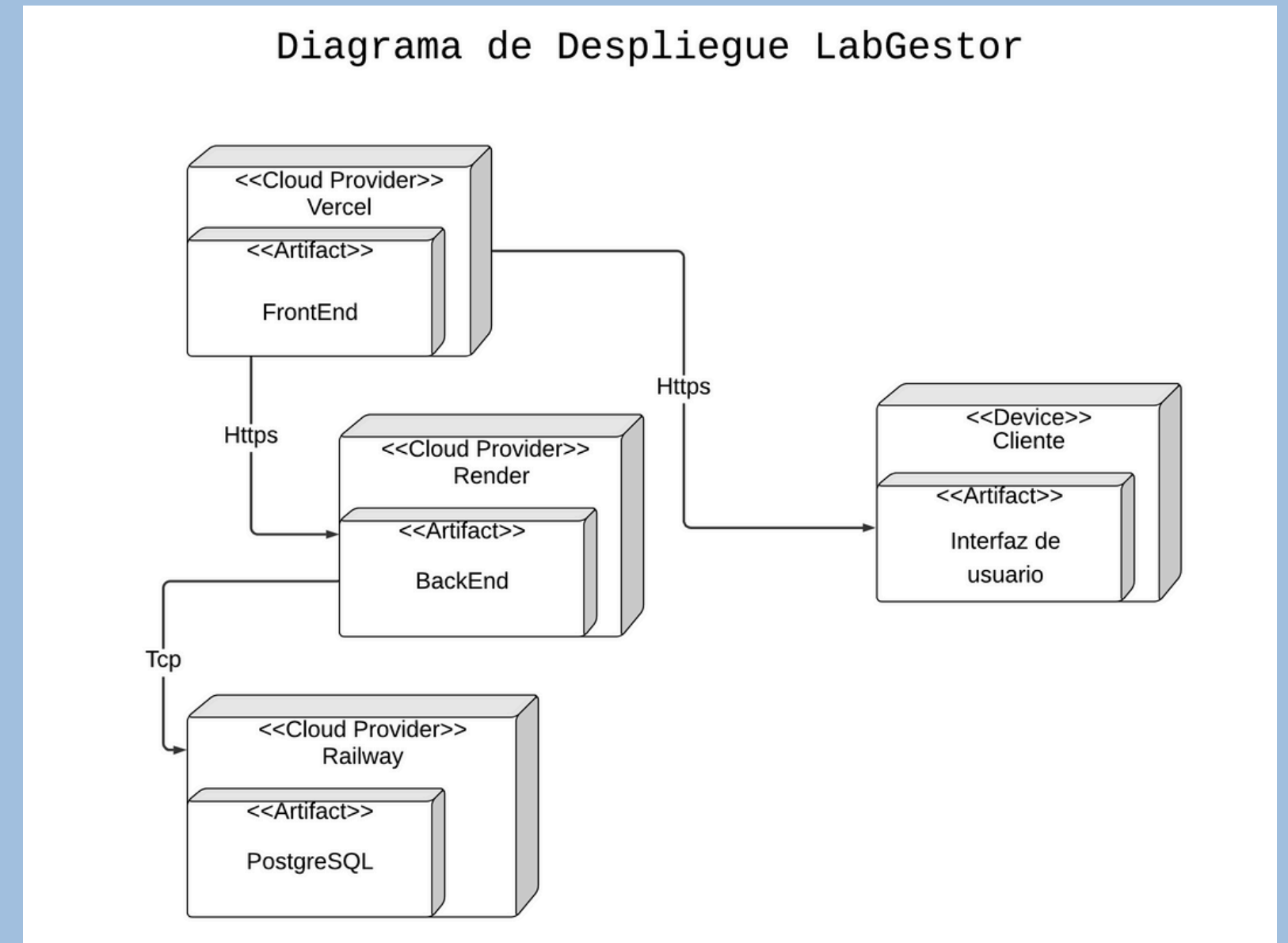
PostgreSQL

Se eligió esta base de datos por capacidad para manejar grandes volúmenes de datos con alta concurrencia, algo esencial para la información sensible del laboratorio.



Fase de Implementación

En la fase de despliegue e implantación, se uso Railway para la base de datos PostgreSQL, Render para el backend en Go y Vercel para el frontend en Astro con React.



Fase de Mantenimiento

Se abordó el punto de Garantía en el cual el equipo de desarrollo se comprometió a tener bajo constante observación el sistema durante 6 meses, en ese tiempo los errores del sistema serán solucionados sin costo alguno



Costos

Personal	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Costo Total
Desarrollar Full Stack	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 9.600.000
Desarrollador FrontEnd	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 9.600.000
Desarrollador Backend	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 9.600.000
Diseñador web y analista	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 1.600.000	\$ 9.600.000
Total							\$ 38.400.000
Hardware							
Computador Desarrollador Full Stack	\$ 1.900.000	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 1.900.000
Computador Desarrollador FrontEnd	\$ 1.800.000	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 1.800.000
Computador Desarrollador BackEnd	\$ 1.800.000	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 1.800.000
Computador Diseñador web y analista	\$ 1.800.000	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 1.800.000
Total							\$ 7.300.000
Software							
Figma	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
AdGrid	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
Astro	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
React	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
Golang	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
Hosting	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 40.000	\$ 40.000
Postgress	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
GitHub	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
LucidChart	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -	\$ -
Total							\$ 40.000
Servicios							
Energia Electrica	\$ 30.000	\$ 30.000	\$ 30.000	\$ 30.000	\$ 30.000	\$ 30.000	\$ 180.000
Internet	\$ 60.000	\$ 60.000	\$ 60.000	\$ 60.000	\$ 60.000	\$ 60.000	\$ 360.000
Transporte	\$ 30.000	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 30.000
Total							\$ 570.000
Total del Software							\$ 46.310.000

Gracias

