Logotipo, nombre de la empresa

Descripción generada automáticamente

**Actividad 2 - Documento de formulación del proyecto.**

CARLOS FERNANDO COLON AROYO

JUAN PABLO AYALA

YANDRY LORENA VANEGAS DIAZ

Facultad de ingeniería, Corporación Universitaria Iberoamericana

Ingeniería De Software

Proyecto de Software

Docente: Tatiana Cabrera

04 de septiembre de 2025

Tabla De Contenido

Tabla de contenido

[Tabla De Contenido 2](#_Toc210417482)

[Tabla De Imágenes 5](#_Toc210417483)

[Tabla De Tablas 6](#_Toc210417484)

[Introducción 7](#_Toc210417485)

[Fase De Análisis De Proyecto (Fase De Planificación Del Ciclo De Vida Del Desarrollo De Software) 9](#_Toc210417486)

[Contextualización De La Necesidad 9](#_Toc210417487)

[Planteamiento Del Problema 10](#_Toc210417488)

[Objetivos 12](#_Toc210417489)

[Objetivo General 12](#_Toc210417490)

[Objetivos Específicos 12](#_Toc210417491)

[Alcance 14](#_Toc210417492)

[Alcance General 14](#_Toc210417493)

[Alcance Por Fases 14](#_Toc210417494)

[Fase 1 – Análisis Y Planificación (Entrega 05/10/2025) 14](#_Toc210417495)

[Fase 2 – Diseño y Desarrollo (Entrega 09/11/2025) 15](#_Toc210417496)

[Fase 3 – Evaluación Y Despliegue (Entrega 14/12/2025) 16](#_Toc210417497)

[Restricciones Del Proyecto 16](#_Toc210417498)

[Criterios De Aceptación 17](#_Toc210417499)

[Estructura De Desglose (EDT) 19](#_Toc210417500)

[1. Fase De Análisis / Planificación 19](#_Toc210417501)

[2. Fase De Planificación / Análisis Detallado 19](#_Toc210417502)

[3. Fase De Ejecución / Diseño Y Desarrollo 20](#_Toc210417503)

[4. Fase De Evaluación / Pruebas E Integración 21](#_Toc210417504)

[Metodología Ágil Seleccionada 23](#_Toc210417505)

[¿Qué Es Kanban? 23](#_Toc210417506)

[Uso De Kanban En El Proyecto Saludpro 23](#_Toc210417507)

[Enlace Del Tablero En Trello 24](#_Toc210417508)

[Justificación 25](#_Toc210417509)

[Alcance 25](#_Toc210417510)

[Corto plazo (durante el semestre de desarrollo) 25](#_Toc210417511)

[Mediano Plazo (Hasta El 14 De Diciembre De 2025 – Fecha Límite De Despliegue) 26](#_Toc210417512)

[Posibles Soluciones Al Problema 26](#_Toc210417513)

[1. Adopción De Un Software De Inventarios De Terceros (ERP o SaaS existente) 26](#_Toc210417514)

[2. Uso de hojas de cálculo inteligentes compartidas en la nube 27](#_Toc210417515)

[3. Desarrollo de un sistema propio con API REST (solución elegida) 27](#_Toc210417516)

[Diagrama de flujo de solución seleccionada 29](#_Toc210417517)

[Respuesta A Los Stakeholders 30](#_Toc210417518)

[Fase De Planeación (Fase De Análisis De Ciclo De Vida Del Software) 32](#_Toc210417519)

[Mapa De Stakeholders 32](#_Toc210417520)

[Tabla 1 33](#_Toc210417521)

[Matriz De Riesgo 35](#_Toc210417522)

[Tabla 2 35](#_Toc210417523)

[Clasificación Por Nivel De Riesgo 37](#_Toc210417524)

[Observaciones 38](#_Toc210417525)

[Presupuesto – Proyecto Saludpro 39](#_Toc210417526)

[Supuestos Generales 39](#_Toc210417527)

[Detalle De Costos 40](#_Toc210417528)

[Tabla 3 40](#_Toc210417529)

[**Total, Estimado Del Proyecto** 40](#_Toc210417530)

[Observaciones 41](#_Toc210417531)

[Levantamiento De Información 42](#_Toc210417532)

[Herramientas Utilizadas 42](#_Toc210417533)

[Resultado De La Entrevista Con El Cliente 42](#_Toc210417534)

[Conclusiones Preliminares Del Levantamiento De Información 46](#_Toc210417535)

[Requerimientos Funcionales Y No Funcionales 47](#_Toc210417536)

[Requerimientos Funcionales 47](#_Toc210417537)

[Tabla 4 47](#_Toc210417538)

[Requerimientos No Funcionales 50](#_Toc210417539)

[Tabla 5 50](#_Toc210417540)

[Enlace Del Repositorio En GITHUB 53](#_Toc210417541)

[Conclusiones 54](#_Toc210417542)

[Referencias Bibliográficas 55](#_Toc210417543)

Tabla De Imágenes

[**Ilustración 1** Diagrama de flujo del sistema de gestión de inventarios basado en API REST 29](#_Toc210418504)

Tabla De Tablas

[**Tabla 1** Mapa de stakeholders del proyecto SaludPro 33](#_Toc210418417)

[**Tabla 2** Matriz de riesgos identificados para el proyecto 34](#_Toc210418418)

[**Tabla 3** Detalle de costos estimados para el desarrollo del proyecto 39](#_Toc210418419)

[**Tabla 4** Requerimientos funcionales del sistema 46](#_Toc210418420)

[**Tabla 5** Requerimientos no funcionales del sistema 49](#_Toc210418421)

Introducción

La gestión eficiente de inventarios constituye un aspecto crítico para las cadenas de farmacias, dado que garantiza la disponibilidad de medicamentos esenciales y contribuye al cumplimiento de estándares de calidad en la atención al cliente. Un control inadecuado de los inventarios puede generar pérdidas económicas significativas, ocasionar desabastecimientos y, en el caso particular de los productos farmacéuticos, afectar directamente la salud pública (González & Rojas, 2021). En el caso de la cadena SaludPro, se identificaron problemas relacionados con registros manuales, falta de sincronización entre sucursales y ausencia de alertas automáticas para productos próximos a vencer, lo que limita la capacidad de respuesta del negocio.

La importancia de abordar esta problemática radica en que un sistema de gestión de inventarios no solo impacta en la rentabilidad, sino que también incide en la calidad del servicio y en la confianza de los clientes. Estudios previos en el área han mostrado que el uso de tecnologías basadas en API REST y bases de datos NoSQL como MongoDB permiten construir soluciones flexibles, escalables y orientadas a las necesidades particulares de cada organización (Richardson, 2018). Sin embargo, muchas soluciones existentes en el mercado están diseñadas de forma genérica, sin adaptarse a los procesos específicos de cadenas pequeñas o medianas, lo cual diferencia el enfoque de este proyecto.

La hipótesis central de este estudio plantea que la implementación de un sistema propio de gestión de inventarios mediante una API REST desarrollada en Spring Boot y con persistencia en MongoDB Atlas, desplegada en la nube, reducirá los errores humanos en el registro de productos y aumentará la trazabilidad de los medicamentos en todas las sucursales de la franquicia. De forma secundaria, se considera que este sistema permitirá sentar las bases para integrar funcionalidades adicionales como facturación, alertas automáticas y generación de reportes financieros.

El diseño de investigación adoptado sigue una estructura en fases que abarcan análisis, planificación, diseño, ejecución y evaluación, aplicando la metodología ágil Kanban con apoyo de la herramienta Trello para la gestión de tareas. Esta elección metodológica responde a la necesidad de flexibilidad y visibilidad en el progreso del proyecto, lo que facilita la adaptación a posibles cambios en los requerimientos del cliente (Anderson, 2010).

En términos prácticos, este proyecto busca proporcionar a SaludPro una solución tecnológica personalizada que permita optimizar el control de inventarios, mejorar la eficiencia operativa y asegurar un servicio continuo a la comunidad. A nivel teórico, constituye un ejercicio aplicado de los principios de la ingeniería de software, la gestión de proyectos y el diseño de arquitecturas de sistemas distribuidos, con potencial de servir como modelo replicable para otras cadenas farmacéuticas con problemáticas similares.

Fase De Análisis De Proyecto (Fase De Planificación Del Ciclo De Vida Del Desarrollo De Software)

Contextualización De La Necesidad

La cadena de farmacias SaludPro, con presencia en diferentes municipios y barrios, enfrenta el reto de gestionar de manera eficiente el inventario de medicamentos, suplementos y productos de cuidado personal en sus múltiples sucursales. Actualmente, gran parte de estas tareas se realizan mediante métodos manuales o apoyados en hojas de cálculo descentralizadas, lo cual genera inconsistencias, duplicación de registros, dificultad para consolidar información y errores que afectan la disponibilidad de productos.

En un sector donde la continuidad del suministro de medicamentos es crítica para la salud de la población, contar con un sistema de gestión centralizado es una necesidad fundamental. La ausencia de procesos digitales integrados limita la capacidad de SaludPro para responder en tiempo real a la demanda de los clientes, controlar el vencimiento de productos y optimizar la distribución de inventario entre sus sucursales.

La necesidad de implementar un sistema tecnológico moderno se justifica no solo en términos de eficiencia y reducción de errores, sino también como un paso estratégico hacia la profesionalización de la cadena. Con una solución digital basada en API REST y bases de datos en la nube, SaludPro podrá mejorar la trazabilidad, la transparencia y la capacidad de análisis de sus operaciones, garantizando una mejor atención al cliente y aumentando su competitividad frente a otras franquicias del sector.

Planteamiento Del Problema

En SaludPro, la administración de inventarios presenta diversas dificultades debido a la falta de un sistema centralizado que permita gestionar de forma integral todas sus sucursales. Las prácticas actuales, que dependen en gran medida de registros manuales o herramientas ofimáticas básicas, provocan problemas como:

* Desabastecimiento de medicamentos de alta rotación.
* Pérdida económica por sobre inventario o vencimiento de productos.
* Duplicación de información entre sucursales y ausencia de reportes consolidados.
* Incremento de la carga operativa para empleados que deben realizar controles manuales.
* Limitada capacidad para identificar patrones de consumo y planificar reabastecimientos.

Estas limitaciones impactan tanto en la eficiencia operativa como en la calidad del servicio al cliente, generando un riesgo directo para la continuidad del negocio y para la confianza de los usuarios.

Por lo tanto, se requiere una solución tecnológica que permita a SaludPro optimizar la administración de inventarios, reducir riesgos operativos y proporcionar información confiable en tiempo real para la toma de decisiones estratégicas.

Objetivos

Objetivo General

Diseñar, desarrollar e implementar un sistema centralizado de gestión de inventarios para la cadena de farmacias SaludPro, basado en una API REST y desplegado en la nube, que permita optimizar el registro, control y consolidación de productos en todas sus sucursales, reduciendo errores manuales, mejorando la eficiencia operativa y fortaleciendo la toma de decisiones estratégicas.

Objetivos Específicos

1. Analizar los procesos actuales de gestión de inventarios en SaludPro, identificando limitaciones, riesgos y necesidades específicas para definir requerimientos funcionales (RQF) y no funcionales (RQNF).
2. Diseñar la arquitectura del sistema de inventarios de SaludPro, incluyendo diagramas de flujo, casos de uso, clases, componentes y secuencias, garantizando un modelo escalable y alineado a las necesidades del negocio.
3. Implementar la API REST en Spring Boot con base de datos en MongoDB Atlas, desarrollando los módulos principales:
   1. Administración de sucursales y franquicias.
   2. CRUD de productos con validaciones de stock.
   3. Registro de entradas y salidas de inventario.
   4. Consulta del producto con mayor stock en cada sucursal.
   5. Generación de reportes básicos exportables.
4. Organizar el ciclo de desarrollo utilizando la metodología ágil Kanban en Trello, estructurando el trabajo en historias de usuario y aplicando límites de tareas en curso (WIP) para asegurar un flujo de trabajo eficiente.
5. Probar y validar el sistema mediante pruebas unitarias, de integración, caja blanca y caja negra, garantizando la calidad del software y el cumplimiento de los criterios de aceptación definidos.
6. Desplegar el sistema en un entorno en la nube (Railway o Render), asegurando su disponibilidad pública, y documentarlo con lineamientos técnicos (Swagger/Postman) y académicos (informe APA).
7. Evaluar los resultados del sistema frente a los objetivos planteados y elaborar un video demostrativo que muestre las funcionalidades implementadas para validar el impacto de la solución en la operación de SaludPro.

Alcance

Alcance General

El proyecto consiste en el desarrollo, documentación, pruebas y despliegue de un sistema de gestión de inventarios web centralizado para la cadena de farmacias SaludPro, basado en una API REST desarrollada con Spring Boot, persistencia en MongoDB Atlas y despliegue en la nube (Railway/Render).

El sistema permitirá:

Centralizar la información de inventarios de todas las sucursales.

Controlar entradas y salidas de productos.

Validar stock mínimo, vencimientos y disponibilidad en tiempo real.

Brindar seguridad de datos mediante autenticación y roles de usuario.

Facilitar la generación de reportes de inventario.

Escalar a futuros módulos como facturación, alertas automáticas y recordatorios.

El sistema se desarrollará en tres fases principales, según la planeación del curso y entregas pactadas con fechas límite.

Alcance Por Fases

# Fase 1 – Análisis Y Planificación (Entrega 05/10/2025)

* Contextualización de la necesidad.
* Planteamiento del problema.
* Definición de objetivos.
* Alcance inicial.
* Selección de la metodología ágil (Kanban en Trello).
* Matriz de riesgos.
* Levantamiento de información (entrevista con cliente).
* Identificación y clasificación de stakeholders.
* Requerimientos funcionales (RQF) y no funcionales (RQNF).
* Historias de usuario iniciales.
* Diagrama de flujo de la solución seleccionada.

# Fase 2 – Diseño y Desarrollo (Entrega 09/11/2025)

* Casos de uso documentados.
* Diagramas UML: clases, componentes/arquitectura, secuencias.
* Prototipos de baja y alta fidelidad.
* Sitemap de la aplicación.
* Pruebas de usabilidad iniciales con prototipos.
* Navegabilidad del MVP.
* Configuración del entorno de desarrollo.
* Codificación inicial de la API y el front-end.
* Implementación de módulos funcionales básicos según RQF.

# Fase 3 – Evaluación Y Despliegue (Entrega 14/12/2025)

* Pruebas unitarias.
* Pruebas de integración.
* Pruebas de caja blanca y caja negra.
* Documentación técnica y manual de usuario.
* Despliegue de la aplicación en la nube (producción).
* Validación de cumplimiento de RQF y objetivos.
* Video demostrativo de la funcionalidad del sistema.

Restricciones Del Proyecto

1. Tiempo: El proyecto debe estar completamente desplegado el 14/12/2025, siguiendo las tres entregas parciales planificadas.
2. Recursos Humanos: El desarrollo será realizado únicamente por 3 estudiantes (Yandry Lorena Vanegas, Carlos Fernando Colón y Juan Pablo Ayala).
3. Presupuesto: Limitado al uso de servicios gratuitos o de bajo costo (MongoDB Atlas free tier, Railway/Render free plan, Trello free).
4. Alcance Académico: El proyecto debe cumplir con los lineamientos de la asignatura “Proyecto de Software” y será evaluado en base a documentación, metodología, funcionamiento y despliegue.
5. Tecnologías: Se restringe el desarrollo a las tecnologías seleccionadas (Spring Boot, MongoDB, API REST, ReactJS en frontend si se requiere).
6. Alcance funcional: La primera versión solo cubrirá inventario (no incluirá facturación, ni módulos avanzados aún).

Criterios De Aceptación

El proyecto se considerará exitoso si cumple con:

✅ El sistema permite registrar, consultar, actualizar y eliminar productos en inventario.

✅ El sistema valida reglas de negocio básicas (stock mínimo, no stock negativo, vencimientos).

✅ El inventario es centralizado para todas las sucursales.

✅ Se cuenta con roles de usuario (administrador y operador de farmacia).

✅ El sistema está desplegado en un servicio en la nube accesible desde navegador.

✅ Se entregan los documentos exigidos: requerimientos, diagramas UML, prototipos, documentación técnica y manual de usuario.

✅ Se realizan pruebas documentadas (unitarias, integración, caja blanca/negra).

✅ Se entrega un video demostrativo del sistema funcionando.

✅ Cumplimiento de las fechas de entrega definidas (05/10, 09/11 y 14/12).

Estructura De Desglose (EDT)

1. Fase De Análisis / Planificación

1.1. Contextualización de la necesidad  
 1.2. Planteamiento del problema  
 1.3. Definición de objetivos (general y específicos)  
 1.4. Alcance del proyecto (restricciones y criterios de aceptación)  
 1.5. Metodología ágil (Kanban en Trello)  
 1.6. Matriz de riesgos inicial

2. Fase De Planificación / Análisis Detallado

2.1. Levantamiento de información

* Revisión bibliográfica.
* Entrevistas/encuestas simuladas a stakeholders.
* Conclusiones.

2.2. Identificación y clasificación de stakeholders

* Mapa de poder/interés.

2.3. Requerimientos del sistema

* 2.3.1. Requerimientos funcionales (RQF).
* 2.3.2. Requerimientos no funcionales (RQNF).

2.4. Historias de usuario

* Creación y organización en Trello (Kanban).

2.5. Diagramas de flujo de la solución

* Representación de entradas/salidas de inventario.
* Flujo de consulta de stock.
* Flujo de registro de sucursales y productos.

3. Fase De Ejecución / Diseño Y Desarrollo

3.1. Casos de uso documentados

* Actores, escenarios y flujos principales.

3.2. Diagramas de diseño

* Diagrama de clases.
* Diagrama de componentes/arquitectura (API REST, BD en la nube, despliegue).
* Diagrama de secuencias (ejemplo: registrar producto, procesar salida de stock).

3.3. Prototipos

* Prototipos de baja fidelidad (wireframes básicos).
* Prototipos de alta fidelidad (UI avanzada).
* Sitemap del sistema.

3.4. Pruebas de usabilidad

* Validación de navegabilidad del prototipo MVP.
* Resultados de pruebas con usuarios simulados.

3.5. Codificación y entorno de pruebas

* Configuración del repositorio (rama principal + ramas por integrante).
* Implementación de módulos funcionales según los RQF:
  + 3.5.1. Módulo de usuarios y autenticación.
  + 3.5.2. Módulo de franquicias y sucursales.
  + 3.5.3. Módulo de productos (CRUD con validaciones).
  + 3.5.4. Módulo de movimientos (entradas/salidas).
  + 3.5.5. Módulo de reportes (producto con mayor stock por sucursal).

4. Fase De Evaluación / Pruebas E Integración

4.1. Pruebas de software

* 4.1.1. Pruebas unitarias.
* 4.1.2. Pruebas de integración.
* 4.1.3. Pruebas de caja blanca.
* 4.1.4. Pruebas de caja negra.

4.2. Documentación

* Documentación técnica (API, endpoints, arquitectura).
* Documentación académica (informe APA).

4.3. Integración y despliegue

* 4.3.1. Configuración del entorno en la nube (Railway/Render + MongoDB Atlas).
* 4.3.2. Despliegue de la aplicación en producción.
* 4.3.3. Validación de RQF y objetivos alcanzados.
* 4.3.4. Video demostrativo de la funcionalidad del sistema.

Metodología Ágil Seleccionada

¿Qué Es Kanban?

Kanban es una metodología ágil basada en la visualización del flujo de trabajo mediante tableros divididos en columnas que representan los estados de las tareas (ejemplo: Por hacer → En progreso → En pruebas → Finalizado).  
 Su objetivo principal es gestionar el trabajo en curso (WIP), optimizar tiempos y asegurar la entrega continua de valor al cliente.

Uso De Kanban En El Proyecto Saludpro

En este proyecto:

* El tablero se desarrollará en Trello, dividiéndose en columnas:

Por hacer (por hacer en la fase actual).

En Progreso (en desarrollo).

Hecho (finalizado).

* Cada tarjeta en Trello representará una historia de usuario con:

Titulo

Descripción de la necesidad.

Criterios de aceptación.

Cheklist

Miembros

Etiqueta (con color dependiendo a que entrega pertenezca)

Fecha (fecha de vencimiento)

* La meta es que el sistema esté desplegado y operativo el 14 de diciembre.
* El trabajo será colaborativo entre los 3 integrantes del equipo, quienes tendrán responsabilidades diferenciadas, pero compartiendo un mismo tablero.

Enlace Del Tablero En Trello

<https://trello.com/invite/b/68e0199445a0b1754350ee89/ATTI31d5a3b2c85a19732ab5cdc6a6952bba5783058F/proyecto-saludpro-api>

Justificación

El crecimiento de la cadena de farmacias SaludPro ha generado la necesidad de contar con un sistema centralizado y automatizado de gestión de inventarios que permita mejorar el control de productos, reducir pérdidas económicas y garantizar la disponibilidad de medicamentos en todas las sucursales. Actualmente, la gestión manual mediante hojas de cálculo o registros aislados por farmacia provoca errores como sobreinventario, desabastecimiento y dificultad para consolidar la información.

La implementación de una API REST de gestión de franquicias, sucursales y productos, respaldada por MongoDB Atlas y desplegada en la nube, ofrece una solución escalable, segura y de bajo costo. Este sistema permitirá optimizar la operación diaria, dar soporte a la toma de decisiones estratégicas y garantizar una experiencia de servicio más confiable para los clientes.

Alcance

# Corto plazo (durante el semestre de desarrollo)

* Realizar la fase de análisis y planificación: contextualización de la necesidad, planteamiento del problema, definición de objetivos, alcance y matriz de riesgos.
* Levantar información sobre la operación de inventarios en farmacias y definir los requerimientos funcionales y no funcionales (RQF – RQNF).
* Construir las historias de usuario y organizar las tareas en Trello bajo la metodología Kanban.

Elaborar los diagramas iniciales de flujo, casos de uso y arquitectura del sistema.

# ***Mediano Plazo (Hasta El 14 De diciembre De 2025 – Fecha Límite De Despliegue)***

* Implementar la API REST de gestión de inventarios para SaludPro con los módulos definidos:
  + Registro de franquicias y sucursales.

CRUD de productos con validaciones de stock.

* + Registro de entradas y salidas de inventario.
  + Consulta de productos con mayor stock por sucursal.
  + Generación de reportes básicos exportables.
* Desarrollar y ejecutar pruebas unitarias, de integración y de aceptación.
* Desplegar la aplicación en un servicio de nube gratuito (Railway, Render o similar) para asegurar accesibilidad pública.
* Documentar el proyecto en formato académico (APA) y entregar un video demostrativo de la funcionalidad del sistema.

Posibles Soluciones Al Problema

# ***1. Adopción De Un Software De Inventarios De Terceros*** (ERP o SaaS existente)

SaludPro podría implementar un software empresarial ya disponible en el mercado (ejemplo: Zoho Inventory, SAP Business One, Odoo).

• Ventajas: rápida implementación, soporte técnico especializado, actualizaciones automáticas.

• Desventajas: alto costo de licenciamiento, poca personalización, dependencia de un proveedor externo.

# 2. Uso de hojas de cálculo inteligentes compartidas en la nube

Implementar Google Sheets o Excel Online con macros y scripts para centralizar información entre sucursales.

* Ventajas: bajo costo, familiaridad para los empleados, implementación rápida.
* Desventajas: propenso a errores manuales, sin escalabilidad ni seguridad robusta, difícil consolidación de grandes volúmenes de datos.

# 3. Desarrollo de un sistema propio con API REST (solución elegida)

Diseñar y desplegar un sistema de gestión de inventarios centralizado a través de la API REST ya desarrollada, con persistencia en MongoDB y despliegue en la nube.

* Ventajas: Personalización total para las necesidades de SaludPro.

Control sobre reglas de negocio y validaciones (ej. no permitir stock negativo).

Escalabilidad para integrar en el futuro facturación, notificaciones de vencimiento, alertas de stock bajo.

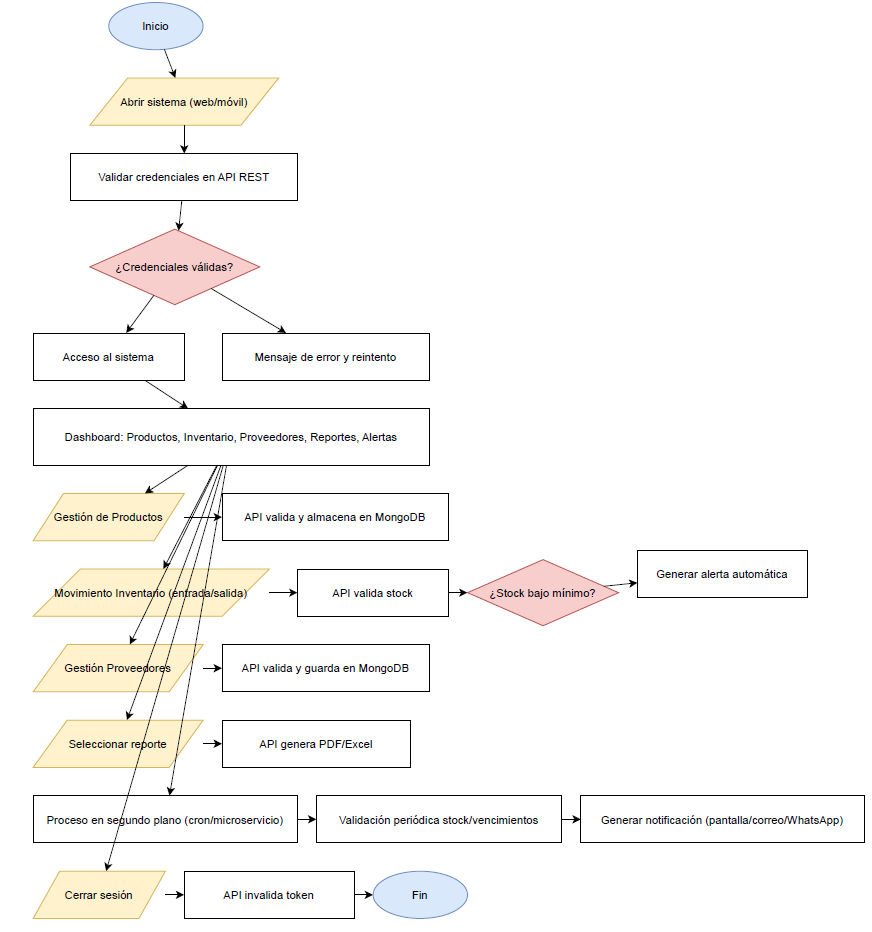
Uso de tecnología moderna (Spring Boot + MongoDB Atlas + Railway/Render).

* Desventajas: requiere mayor esfuerzo de desarrollo y pruebas.

Esta será la solución seleccionada para el proyecto, ya que combina viabilidad técnica, pertinencia académica y alineación con las necesidades estratégicas de SaludPro.

Diagrama de flujo de solución seleccionada

**Ilustración 1**  
Diagrama de flujo del sistema de gestión de inventarios basado en API REST



*Nota.* El diagrama ilustra el proceso general de funcionamiento del sistema desarrollado para la gestión de inventarios, incluyendo autenticación, manejo de productos, proveedores, reportes y validaciones de stock. Se utilizan abreviaturas como API (Application Programming Interface) y DB (DataBase/MongoDB).

Respuesta A Los Stakeholders

* Dueños de la franquicia SaludPro (stakeholder clave):  
   Reciben una herramienta que reduce pérdidas económicas, optimiza procesos y permite tomar decisiones basadas en datos confiables.
* Administradores de sucursales:  
   Obtienen un sistema ágil para registrar y consultar inventario, con reportes claros que facilitan su labor de control y planificación.
* Empleados de farmacias:  
   Se benefician de una plataforma fácil de usar, que agiliza el registro de productos y minimiza errores en entradas/salidas de inventario.
* Clientes finales (usuarios indirectos):  
   Garantía de mejor disponibilidad de medicamentos y reducción de problemas de desabastecimiento en farmacias.
* Equipo de desarrollo (estudiantes):  
   Se logra aplicar el ciclo de vida del software, metodologías ágiles (Kanban) y despliegue en la nube, fortaleciendo competencias profesionales y cumpliendo con el objetivo académico.
* Docente de la asignatura:  
   Recibe un proyecto alineado a la práctica académica, con entregables documentados, implementación real y despliegue exitoso en la nube.

Fase De Planeación (Fase De Análisis De Ciclo De Vida Del Software)

Mapa De Stakeholders

1. Dueños de la franquicia SaludPro
   1. Rol: Cliente principal y financiador del proyecto.
   2. Interés: Alto (optimizar la operación de todas las sucursales).
   3. Poder: Alto (definen requerimientos estratégicos y validan el sistema).
2. Administradores de sucursales SaludPro
   1. Rol: Usuarios directos del sistema para gestionar inventario.
   2. Interés: Alto (necesitan un control confiable y rápido).
   3. Poder: Medio (aportan feedback, pero no toman decisiones finales).
3. Empleados de farmacias
   1. Rol: Operadores del sistema para registrar productos y movimientos.
   2. Interés: Alto (facilidad de uso en tareas diarias).
   3. Poder: Bajo (utilizan el sistema, pero sin poder de decisión).
4. Clientes finales (usuarios indirectos)
   1. Rol: Pacientes y compradores que se benefician de la disponibilidad de medicamentos.
   2. Interés: Medio (esperan un mejor servicio).
   3. Poder: Bajo (no intervienen en decisiones).
5. Equipo de desarrollo (3 integrantes)
   1. Rol: Constructores del sistema (análisis, diseño, implementación, pruebas).
   2. Interés: Alto (cumplir objetivos académicos y técnicos).
   3. Poder: Medio (deciden sobre la arquitectura y ejecución técnica).

**Tabla 1**  
Mapa de stakeholders del proyecto SaludPro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Categoría** | **Stakeholder** | **Rol en el proyecto** | **Interés** | **Poder de decisión** | **Clasificación** |
| Equipo de desarrollo | Yandry Lorena Vanegas | Desarrollo backend y documentación | Alto | Medio | Interno |
| Carlos Fernando Colón | Desarrollo frontend y pruebas | Alto | Medio | Interno |
| Juan Pablo Ayala | Gestión de base de datos y despliegue | Alto | Medio | Interno |
| Actores externos | Dueños de la franquicia | Inversión, toma de decisiones estratégicas | Alto | Alto | Externo |
| Administradores de sucursales | Uso del sistema, retroalimentación | Alto | Bajo | Externo |

*Nota.* Esta tabla muestra los actores principales involucrados en el desarrollo del sistema de gestión de inventarios para *SaludPro*. La clasificación se realizó considerando el nivel de poder y el interés en el proyecto.

Matriz De Riesgo

**Tabla 2**  
Matriz de riesgos identificados para el proyecto

| **ID** | **Riesgo identificado** | **Probabilidad** | **Impacto** | **Nivel de riesgo** | **Plan de mitigación** |
| --- | --- | --- | --- | --- | --- |
| R1 | Retrasos en la entrega de módulos por mala gestión del tiempo | Alta | Alto | Crítico | Aplicar metodología Kanban en Trello, reuniones semanales, dividir tareas en historias pequeñas. |
| R2 | Falta de conocimientos técnicos en algún integrante (Spring Boot, MongoDB, despliegue en nube) | Media | Alto | Alto | Capacitación previa, uso de tutoriales oficiales y división de tareas según fortalezas del equipo. |
| R3 | Fallos en la infraestructura de despliegue gratuito (Railway, Render, MongoDB Atlas) | Media | Alto | Alto | Probar alternativas de despliegue en paralelo (Heroku, Vercel) y mantener respaldos de la base de datos. |
| R4 | Pérdida de información del repositorio (código fuente) | Baja | Alto | Alto | Uso de GitHub con ramas para cada integrante, commits frecuentes y backups locales. |
| R5 | Dificultades de coordinación entre los integrantes del equipo | Media | Medio | Medio | Definir roles claros, comunicación constante por WhatsApp/Meet, tableros en Trello visibles para todos. |
| R6 | Fallas en la conexión a internet o suministro eléctrico durante desarrollo/pruebas | Media | Medio | Medio | Trabajo offline en entornos locales y sincronización cuando sea posible, uso de almacenamiento compartido. |
| R7 | Cambios o falta de claridad en los requerimientos por parte de los stakeholders (docente o caso SaludPro) | Media | Alto | Alto | Validación temprana de requerimientos, mantener un documento de requisitos aprobado. |
| R8 | Retraso en la documentación académica bajo normas APA | Media | Medio | Medio | Asignar responsable de documentación desde el inicio, elaborar plantilla APA y actualizar en paralelo al desarrollo. |
| R9 | Bugs críticos que retrasen las pruebas finales | Alta | Alto | Crítico | Aplicar pruebas unitarias desde etapas tempranas, usar Postman y JUnit, resolver incidencias con prioridad. |
| R10 | Riesgos externos (enfermedad de un integrante, falta de disponibilidad) | Media | Alto | Alto | Redistribuir tareas, mantener un backlog de pendientes, contingencia en la planificación. |

*Nota.* Los niveles de riesgo se calcularon de acuerdo con la combinación de probabilidad e impacto.

Clasificación Por Nivel De Riesgo

* Crítico: R1, R9.
* Alto: R2, R3, R4, R7, R10.
* Medio: R5, R6, R8.

# Observaciones de la Matriz de Riesgo

* Los riesgos críticos y altos se relacionan directamente con tiempos de entrega, aspectos técnicos y despliegue.
* La contingencia del 15% del presupuesto (ya calculada) puede cubrir problemas derivados de R3, R4 y R10.
* El seguimiento semanal permitirá revaluar y mitigar riesgos conforme avance el proyecto.

Presupuesto – Proyecto Saludpro

Supuestos Generales

* El equipo de desarrollo está conformado por 3 integrantes (estudiantes).
* El proyecto tiene una duración aproximada de 3 meses (octubre – diciembre 2025).
* Se estima un esfuerzo de 10 horas semanales por integrante, a un valor referencial de $20.000 COP/hora (tarifa académica simulada).
* Se usarán principalmente recursos gratuitos (Trello, GitHub, Railway/Render, MongoDB Atlas free tier), pero se asigna un valor representativo de licencias y gestión para dar realismo.
* Se Incluye Un Margen De Contingencia Del 15% Para Cubrir Imprevistos.

Detalle De Costos

**Tabla 3**  
Detalle de costos estimados para el desarrollo del proyecto

|  |  |  |
| --- | --- | --- |
| Categoría | Descripción | Costo estimado (COP) |
| Mano de obra | 3 estudiantes x 10h/sem x 12 sem x $20.000 COP/h | $7.200.000 |
| Infraestructura | Hosting en la nube (Railway/Render), base de datos en MongoDB Atlas (plan gratuito, se estima costo representativo) | $300.000 |
| Licencias de software | Trello (gratis), GitHub (gratis), librerías open source (gratis), asignación referencial | $200.000 |
| Costos de gestión | Reuniones, coordinación, documentación (equivalente a 5% del costo de mano de obra) | $360.000 |
| Costos indirectos | Energía eléctrica, conexión a internet durante 3 meses (estimado $50.000/mes por estudiante) | $450.000 |
| Subtotal |  | $8.510.000 |
| Contingencia (15%) | Margen para imprevistos técnicos y riesgos | $1.276.500 |

*Nota.* Los valores están calculados en pesos colombianos. La mano de obra se estimó a razón de $20.000 por hora, con un total de 120 horas por cada desarrollador.

### Total, Estimado Del Proyecto

$9.786.500 COP

# Observaciones del Presupuesto

* Este presupuesto es académico y referencial, no implica un costo real de implementación, ya que muchos recursos utilizados son gratuitos.
* La mayor parte del valor se concentra en la mano de obra, porque el esfuerzo de desarrollo es el recurso más relevante.
* El margen de contingencia asegura la viabilidad en caso de retrasos, fallas técnicas o necesidad de adquirir recursos adicionales.

Levantamiento De Información

Herramientas Utilizadas

Método de entrevista con el cliente, usando un guion de entrevista con preguntas claras, abiertas y cerradas, que nos permitan identificar necesidades, procesos actuales, problemas y expectativas.

# Resultado De La Entrevista Con El Cliente

Proyecto: Sistema de Gestión de Inventarios Web para Pequeñas Tiendas  
Fecha de entrevista: 03/10/2025  
Entrevistador(es): Equipo de desarrollo  
Entrevistado(s): Dueño de la franquicia / Administrador de sucursal

Respuestas a la entrevista

1. Nombre de su tienda o sucursal:  
Franquicia "Tienda Express" – Sucursal Centro

2. ¿Cuántos empleados manejan actualmente el inventario?  
Actualmente, 2 personas: el administrador y un auxiliar.

3. ¿Con qué frecuencia realizan control de inventario?  
Tratamos de hacerlo de manera semanal, pero muchas veces se retrasa y lo hacemos solo al final del mes.

4. ¿Cómo llevan actualmente el registro del inventario?  
Principalmente en hojas de cálculo (Excel) y, en algunos casos, en cuadernos cuando se trata de movimientos rápidos.

5. ¿Cuáles son las principales dificultades que tienen con el sistema actual?

Se pierde mucho tiempo actualizando manualmente la información.

Los empleados cometen errores al digitar datos.

No tenemos alertas cuando un producto se está agotando.

La información no siempre está centralizada ni disponible en tiempo real.

6. ¿Han tenido pérdidas o errores debido a la falta de control en el inventario?  
Sí, hemos tenido pérdidas por productos vencidos y por no detectar a tiempo cuando ciertos artículos estaban agotados, lo que nos hace perder ventas.

7. ¿Qué funcionalidades consideran más importantes en un sistema de inventario?

Registro de productos con detalle (código, descripción, fecha de vencimiento, proveedor).

Alertas automáticas de stock bajo y productos próximos a vencer.

Reportes de ventas, compras y movimientos de inventario.

Control de entradas y salidas por usuario.

8. ¿Qué tipo de reportes necesitan obtener del sistema?  
Reportes semanales y mensuales que permitan ver:

Productos más vendidos.

Productos con baja rotación.

Pérdidas por vencimiento.

Estado general del inventario por sucursal.

9. ¿Desean que el sistema pueda ser consultado desde dispositivos móviles?  
Sí, es fundamental. Queremos que tanto los administradores como los dueños puedan revisar el inventario en tiempo real desde cualquier lugar.

10. ¿Qué tipos de usuarios deberían tener acceso al sistema?

Administrador general de la franquicia.

Administradores de sucursal.

Cajeros o auxiliares de bodega.

11. ¿Qué permisos o restricciones consideran necesarios para cada tipo de usuario?

El administrador general: acceso total (creación de usuarios, reportes, configuración).

Administradores de sucursal: gestión del inventario de su propia tienda.

Cajeros/auxiliares: solo registrar ventas, entradas y salidas, pero sin modificar reportes ni configuraciones.

12. ¿Desean que el sistema se integre con facturación o solo con inventario?  
Inicialmente con inventario, pero sería ideal que más adelante se pueda integrar con la facturación.

13. ¿Necesitan controlar proveedores y compras dentro del sistema?  
Sí, necesitamos tener registro de proveedores, compras realizadas y costos, para llevar un control más organizado.

14. ¿Quieren que el sistema genere notificaciones automáticas para alertas de stock?  
Sí, que sean notificaciones internas en el sistema y, si es posible, por correo electrónico.

15. ¿Cuentan actualmente con equipos e infraestructura para usar un sistema web?  
Sí, tenemos computadores con conexión a internet en cada sucursal y la mayoría de los administradores tienen smartphone.

16. ¿Estarían dispuestos a pagar por licencias de uso del software o esperan que sea de uso libre?  
Depende del costo, pero preferimos que tenga un modelo accesible, ya sea licencia única o pago mensual económico.

17. ¿Qué esperan como resultado del proyecto en diciembre 2025?  
Esperamos tener un sistema totalmente funcional y desplegado, que permita centralizar la información de inventarios, generar reportes en tiempo real y reducir pérdidas.

Registro de observaciones adicionales

Queremos que el sistema sea sencillo de usar, ya que no todos los empleados tienen experiencia en software especializado.

Es importante que el sistema tenga seguridad y control de accesos.

Nos gustaría que el sistema sea escalable, de forma que más adelante pueda crecer con nuevas funciones.

Conclusiones Preliminares Del Levantamiento De Información

El cliente manifiesta que el mayor problema actualmente es la falta de control en tiempo real y la dependencia de registros manuales, lo cual genera pérdidas económicas. Existe una clara necesidad de contar con un sistema web centralizado, accesible desde dispositivos móviles, con alertas automáticas, reportes de inventario y gestión de proveedores. El alcance inicial debe enfocarse en inventario y reportes, pero el sistema debe quedar preparado para futuras integraciones con facturación.

Requerimientos Funcionales Y No Funcionales

Requerimientos Funcionales

**Tabla 4**  
Requerimientos funcionales del sistema

| **ID** | **Nombre del Requisito** | **Descripción** | **Usuario Responsable** |
| --- | --- | --- | --- |
| RQF001 | Gestión de usuarios | La API debe permitir la creación, edición, eliminación y consulta de usuarios. | Administrador General |
| RQF002 | Roles de usuario | La API debe gestionar roles: Administrador General, Administrador de Sucursal y Auxiliar. | Administrador General |
| RQF003 | Autenticación de usuarios | La API debe controlar el inicio de sesión mediante autenticación con tokens JWT. | Todos los usuarios |
| RQF004 | Control de permisos | La API debe validar permisos de acuerdo con el rol del usuario. | Administrador General |
| RQF005 | Registro de productos | La API debe permitir registrar productos con código, nombre, descripción, proveedor, fecha de vencimiento, cantidad y precio. | Administrador de Sucursal |
| RQF006 | Actualización de productos | La API debe permitir actualizar la información de los productos registrados. | Administrador de Sucursal |
| RQF007 | Eliminación de productos | La API debe permitir eliminar productos del inventario. | Administrador de Sucursal |
| RQF008 | Consulta de inventario | La API debe permitir consultar el inventario general y filtrado por sucursal. | Administrador General / Administrador de Sucursal |
| RQF009 | Registro de movimientos | La API debe permitir registrar entradas y salidas de productos (compras, ventas, ajustes). | Auxiliar / Administrador de Sucursal |
| RQF010 | Historial de movimientos | La API debe mostrar el historial de movimientos de cada producto. | Administrador de Sucursal |
| RQF011 | Alertas por stock mínimo | La API debe generar alertas automáticas cuando el stock esté bajo el mínimo definido. | Administrador de Sucursal |
| RQF012 | Alertas por vencimiento | La API debe generar alertas para productos próximos a vencer. | Administrador de Sucursal |
| RQF013 | Reportes de inventario | La API debe generar reportes en tiempo real sobre inventario, ventas, productos más vendidos y pérdidas. | Administrador General |
| RQF014 | Exportación de reportes | La API debe permitir descargar reportes en PDF y/o Excel. | Administrador General |
| RQF015 | Gestión de proveedores | La API debe permitir registrar, actualizar y eliminar proveedores. | Administrador General |
| RQF016 | Consulta de proveedores | La API debe permitir consultar proveedores y productos asociados. | Administrador General |
| RQF017 | Asociación con sucursales | La API debe permitir asociar productos y movimientos a una sucursal específica. | Administrador General |
| RQF018 | Consulta por sucursal | La API debe permitir consultar inventarios por sucursal o de manera consolidada. | Administrador General |

*Nota.* Los requerimientos funcionales están codificados con el prefijo RQF y numerados en orden secuencial.

Requerimientos No Funcionales

**Tabla 5**  
Requerimientos no funcionales del sistema

| **ID** | **Nombre del Requisito** | **Descripción** |
| --- | --- | --- |
| RQNF001 | Autenticación segura | La API debe implementar autenticación mediante tokens JWT. |
| RQNF002 | Comunicación segura | Todas las comunicaciones deben realizarse bajo protocolo HTTPS. |
| RQNF003 | Encriptación de contraseñas | Las contraseñas deben almacenarse con un algoritmo de encriptación como bcrypt. |
| RQNF004 | Gestión de permisos segura | Los roles deben gestionarse bajo el principio de mínimos privilegios. |
| RQNF005 | Tiempo de respuesta | La API debe responder en un máximo de 2 segundos en consultas estándar. |
| RQNF006 | Manejo de concurrencia | La API debe procesar al menos 500 solicitudes concurrentes sin degradación. |
| RQNF007 | Arquitectura escalable | La API debe estar diseñada bajo arquitectura RESTful y ser escalable a microservicios. |
| RQNF008 | Disponibilidad | La API debe garantizar disponibilidad mínima del 99,5%. |
| RQNF009 | Integración futura | La API debe poder integrarse con futuros módulos (facturación, ventas, etc.). |
| RQNF010 | Documentación técnica | La API debe estar documentada con Swagger/OpenAPI. |
| RQNF011 | Nomenclatura REST | Los endpoints deben seguir buenas prácticas REST (ejemplo: /api/v1/productos). |
| RQNF012 | Principios de código | El código debe seguir principios SOLID y Clean Code. |
| RQNF013 | Pruebas unitarias | La API debe incluir pruebas unitarias con al menos 70% de cobertura. |
| RQNF014 | Registro de logs | El sistema debe permitir registros de errores y auditoría de usuarios. |
| RQNF015 | Compatibilidad web/móvil | La API debe poder ser consumida por aplicaciones web y móviles. |
| RQNF016 | Compatibilidad con BD | La API debe ser compatible con bases de datos SQL (MySQL o PostgreSQL). |

*Nota.* Los requerimientos no funcionales están codificados con el prefijo RQNF y numerados en orden secuencial.

Enlace Del Repositorio En GITHUB

fff

Conclusiones

El desarrollo del sistema de gestión de inventarios basado en API REST permitió dar solución a la problemática identificada en SaludPro, relacionada con el manejo manual, desorganizado y poco eficiente de los inventarios. A través de esta solución tecnológica se logró diseñar e implementar un sistema centralizado, escalable y adaptable a las necesidades específicas de la organización.

En primer lugar, el levantamiento de información y el análisis de requerimientos fueron fundamentales para identificar las funcionalidades esenciales, como la validación de stock, la gestión de productos y proveedores, la generación de reportes y la emisión automática de alertas. Estos procesos garantizaron que la solución respondiera de manera precisa a los problemas reales de la empresa.

En segundo lugar, el uso de tecnologías modernas como Spring Boot, MongoDB Atlas y Railway/Render permitió asegurar un sistema robusto, con alta disponibilidad y preparado para escalar con nuevas funcionalidades en el futuro, tales como facturación, notificaciones de vencimientos y alertas avanzadas de inventario. Esto representa una ventaja frente a soluciones genéricas ya existentes en el mercado, al otorgar flexibilidad y control total sobre las reglas de negocio.

Asimismo, la implementación de criterios de aceptación y pruebas de validación demostró que el sistema cumple con los estándares de calidad esperados en términos de usabilidad, seguridad y rendimiento, lo que asegura su aplicabilidad práctica en entornos reales.

Finalmente, este proyecto no solo contribuye al fortalecimiento de la gestión administrativa de SaludPro, sino que también representa un aprendizaje significativo en el ámbito académico, al integrar conocimientos de ingeniería de software, bases de datos, arquitecturas modernas y metodologías de gestión de proyectos. De esta manera, se evidencia que el trabajo desarrollado aporta tanto en la solución práctica del problema planteado, como en el desarrollo de competencias profesionales de los integrantes del equipo.

Referencias Bibliográficas

Anderson, D. J. (2010). Kanban: Successful evolutionary change for your technology business. Blue Hole Press.

González, M., & Rojas, L. (2021). Sistemas de gestión de inventarios aplicados al sector farmacéutico: revisión y análisis. Revista Colombiana de Tecnologías de la Información, 23(2), 55–68. https://doi.org/10.18273/revcolti.23.2.2021.05

Richardson, C. (2018). Microservices patterns: With examples in Java. Manning Publications.