

# Documentación del Sistema de Gestión de Inventarios — StockApp

Santiago Correa  
Juan Sebastián Andraus  
Andrés Felipe Giraldo

*Estudiantes de Ingeniería en Sistemas*

**Materia:** Análisis y Diseño de Sistemas II

**Profesor:** Juan Pablo Arango

**Universidad de Antioquia**

**Medellín, Colombia**

**Fecha:** 4 de diciembre de 2025

# Índice general

<b>1. Introducción</b>	<b>2</b>
1.1. Propósito del documento . . . . .	2
1.2. Resumen del proyecto . . . . .	2
1.3. Tecnologías utilizadas . . . . .	2
<b>2. Descripción General del Sistema</b>	<b>3</b>
2.1. Objetivos del sistema . . . . .	3
2.2. Problema que soluciona . . . . .	3
2.3. Usuarios del sistema . . . . .	3
<b>3. Arquitectura y Tecnologías</b>	<b>4</b>
3.1. Arquitectura del sistema . . . . .	4
3.2. Frontend con Next.js . . . . .	4
3.3. Backend con Node.js . . . . .	4
3.4. Diseño con TailwindCSS . . . . .	4
3.5. Gráficas con Recharts . . . . .	4
<b>4. Funcionalidades del Sistema</b>	<b>5</b>
4.1. Gestión de productos . . . . .	5
4.2. Control de stock . . . . .	5
4.3. Dashboard estadístico . . . . .	5
4.4. Notificaciones o alertas . . . . .	5
<b>5. Diseño de Base de Datos</b>	<b>6</b>
5.1. Modelo y gestión . . . . .	6
5.2. Entidades principales . . . . .	6
<b>6. Manual de Usuario</b>	<b>7</b>
6.1. Acceso al sistema . . . . .	7
6.2. Gestión de inventario paso a paso . . . . .	7
6.3. Visualización de reportes . . . . .	7
<b>7. Conclusiones</b>	<b>8</b>
7.1. Conclusiones . . . . .	8
7.2. Mejoras futuras . . . . .	8

# Capítulo 1

## Introducción

### 1.1. Propósito del documento

El presente documento tiene como propósito describir de manera detallada la arquitectura, funcionalidades, diseño técnico, procesos operativos y tecnologías empleadas en el desarrollo de **StockApp**, un sistema web orientado a la gestión de inventarios en tiempo real. Esta documentación está dirigida a desarrolladores, administradores del sistema, personal de soporte técnico y entidades interesadas en comprender el funcionamiento interno de la plataforma, su estructura tecnológica y su alcance como solución para el control eficiente de existencias.

### 1.2. Resumen del proyecto

**StockApp** es una plataforma web diseñada para gestionar inventarios, transacciones y usuarios de manera centralizada, rápida y accesible desde cualquier dispositivo con conexión a internet. Si bien su arquitectura permite utilizarse en distintos sectores, la solución puede implementarse en comercios minoristas, tiendas de ferretería, servicios logísticos o cualquier organización que requiera un control de stock efectivo.

El sistema ofrece funcionalidades como registro y clasificación de productos, control de entradas y salidas, gestión de perfiles de usuario (*Admin* y *User*), visualización de reportes y alertas de inventario bajo.

### 1.3. Tecnologías utilizadas

Para el desarrollo de **StockApp** se implementó una arquitectura basada en tecnologías modernas que permiten rapidez, escalabilidad y actualización en tiempo real:

- **Next.js**: Framework basado en React para la construcción del frontend.
- **Node.js**: Entorno utilizado para el backend mediante API interna.
- **TailwindCSS**: Framework de diseño basado en utilidades.
- **Recharts**: Librería de gráficos para el dashboard estadístico.
- **MySQL**: Base de datos relacional encargada del almacenamiento de información.

# Capítulo 2

## Descripción General del Sistema

### 2.1. Objetivos del sistema

El objetivo principal de **StockApp** es proporcionar una plataforma web integral que permita gestionar inventarios en tiempo real, ofreciendo control, trazabilidad y acceso seguro a la información dentro de una organización.

Los objetivos específicos son:

- Gestionar productos y su disponibilidad.
- Registrar movimientos de entrada y salida.
- Permitir gestión de usuarios.
- Generar estadísticas del inventario.
- Notificar sobre stock bajo.

### 2.2. Problema que soluciona

StockApp soluciona problemas comunes como desorden de registros, pérdida de inventario, ausencia de reportes y falta de control del stock. Centraliza la información y permite trazabilidad en tiempo real.

### 2.3. Usuarios del sistema

- **Administrador:** acceso total.
- **Usuario:** operaciones de inventario.
- **Consulta:** solo lectura.

# Capítulo 3

## Arquitectura y Tecnologías

### 3.1. Arquitectura del sistema

StockApp utiliza una arquitectura cliente-servidor dentro del mismo proyecto Next.js, integrando frontend, backend y acceso a base de datos.

### 3.2. Frontend con Next.js

El frontend se organiza mediante App Router, usando componentes reutilizables dentro de `src/components`.

### 3.3. Backend con Node.js

El backend utiliza las rutas API de Next.js para procesar solicitudes, lógica de negocio e interacción con la base de datos.

### 3.4. Diseño con TailwindCSS

Se aplican clases utilitarias para diseño consistente y rápido sin archivos CSS adicionales.

### 3.5. Gráficas con Recharts

El dashboard muestra datos de inventario, usuarios y movimientos mediante gráficos dinámicos.

# Capítulo 4

## Funcionalidades del Sistema

### 4.1. Gestión de productos

Registrar, editar, eliminar y consultar productos con campos básicos de identificación.

### 4.2. Control de stock

Registra transacciones de entrada y salida, actualiza cantidades y guarda historial.

### 4.3. Dashboard estadístico

Incluye:

- Total de productos
- Total de usuarios
- Movimientos
- Alertas de stock bajo

### 4.4. Notificaciones o alertas

Muestra aviso cuando un producto tiene tres unidades o menos.

# Capítulo 5

## Diseño de Base de Datos

### 5.1. Modelo y gestión

Base de datos MySQL administrada mediante Prisma.

### 5.2. Entidades principales

- Productos
- Movimientos
- Usuarios
- Categorías

# Capítulo 6

## Manual de Usuario

### **6.1. Acceso al sistema**

Ingreso mediante usuario y contraseña con acceso según rol.

### **6.2. Gestión de inventario paso a paso**

Agregar productos, registrar movimientos y consultar historial.

### **6.3. Visualización de reportes**

Interpretación gráfica de disponibilidad y movimientos.

# Capítulo 7

## Conclusiones

### 7.1. Conclusiones

StockApp centraliza el control de inventario, mejora trazabilidad y reduce errores mediante una plataforma moderna y escalable.

### 7.2. Mejoras futuras

- Alertas por correo o mensaje
- Integración con proveedores
- Exportación de reportes
- Auditoría avanzada