

Proyecto Integrador: Kingdom Barber

Integrantes:

Alejandro Urrego

Andrés Vallejo

Juan Manuel Rivera

Institución:

Institución Técnica CESDE

Fecha:

2025

1. Introducción y Visión General

1.1. Resumen del Proyecto

El proyecto **Kingdom Barber** nace como una plataforma digital integral, diseñada para transformar y modernizar la gestión de barberías. Su objetivo es ofrecer una solución robusta que simplifique la administración de citas, enriquezca la interacción con el cliente y proporcione herramientas avanzadas para la toma de decisiones.

La solución se compone de un ecosistema de aplicaciones que funcionan de manera sinérgica:

- **Una API Central (Node.js)** que actúa como el cerebro de la operación, gestionando la lógica de negocio y la persistencia de los datos.
- **Una Plataforma Web (Next.js)** orientada al cliente, que garantiza una experiencia de usuario fluida para la reserva de servicios.
- **Un Panel de Gestión y Análisis (Python + Streamlit)**, destinado a los administradores para ofrecer una visión profunda del rendimiento del negocio e integrar un Asistente de Inteligencia Artificial.

1.2. Planteamiento del Problema

La gestión tradicional de barberías enfrenta desafíos en la era digital, como la dependencia de procesos manuales, una comunicación ineficiente y la falta de análisis de datos. Esto resulta en una menor eficiencia operativa y una experiencia que no cumple con las expectativas actuales de los clientes, quienes buscan comodidad, seguridad y personalización.

1.3. Solución Propuesta

Kingdom Barber aborda estos desafíos mediante una plataforma unificada que digitaliza y optimiza las operaciones. La solución incluye funcionalidades clave como la gestión de reservas, la personalización de servicios y la comunicación directa con los clientes. Adicionalmente, el panel administrativo empodera a los dueños del negocio con herramientas de análisis de datos e IA para identificar patrones, detectar oportunidades y automatizar tareas de marketing.

2. Objetivos del Proyecto

2.1. Objetivo General

Mejorar la eficiencia operativa de las barberías mediante la digitalización de la gestión de reservas y la comunicación con los clientes, ofreciendo una experiencia personalizada y aumentando la satisfacción del cliente.

2.2. Objetivos Específicos

- **Seguridad y Acceso:** Desarrollar un sistema de autenticación robusto.
- **Usabilidad y Navegación:** Crear interfaces de usuario intuitivas para una experiencia fluida.
- **Gestión de Reservas:** Implementar un sistema de citas completo con calendarios interactivos.
- **Experiencia Personalizada:** Ofrecer opciones de personalización y un catálogo detallado de servicios.

3. Arquitectura General del Sistema

3.1. Diseño de la Arquitectura

El proyecto está diseñado bajo una **arquitectura de servicios desacoplados**, donde cada componente tiene una responsabilidad única. El núcleo del sistema es la **API Central**, que centraliza toda la lógica de negocio y los datos.

- **API Central (Núcleo del Sistema):** Es el componente central desarrollado en **Node.js y Express**. Sirve como única fuente de verdad, gestionando los datos de usuarios, citas y servicios a través de endpoints RESTful.
- **Frontend (Plataforma del Cliente - PI_WEB2):** Es la capa de presentación desarrollada con **Next.js (React)**. Es un cliente puro que consume los datos de la API Central.
- **Panel de Administración (Inteligencia de Negocio - PI_NTP):** Es una aplicación web independiente en **Python y Streamlit**. Se conecta a la misma API Central para realizar análisis y generar visualizaciones.
- **Aplicación Independiente (PI_MOVIL2):** Es un proyecto autocontenido con un backend en **Java + Spring Boot**. **No se conecta a la API Central** y funciona de manera autónoma.

3.2. Flujo de Datos

El flujo de datos sigue un patrón cliente-servidor a través de la API. Por ejemplo, al crear una cita desde la web:

1. **Cliente (Frontend - PI_WEB2):** El usuario rellena el formulario de reserva en la interfaz de Next.js.
2. **Petición:** El frontend empaqueta los datos en un objeto JSON y envía una petición POST al endpoint `/nuevas_citas` de la API Central.
3. **Lógica (API Central):** El servidor Express en Node.js recibe la petición, valida los datos y los guarda en el archivo `nuevas_citas.json`.
4. **Respuesta:** La API devuelve una respuesta JSON al frontend con un código de estado 201 Created, confirmando la creación.
5. **Actualización de UI:** El frontend de Next.js recibe la confirmación y actualiza el estado de la aplicación para mostrar la cita agendada.

4. Módulos del Proyecto

4.1. Módulo 1: API Central (Node.js + Express)

- **Descripción:** Es el corazón del sistema. Gestiona toda la lógica, los datos (en archivos JSON) y la seguridad a través de una API REST.
- **Funcionalidades Principales:**
 - Endpoints RESTful para CRUD de clientes, barberos, sedes, servicios, citas y galería.
 - Manejo de subida de archivos con Multer.

4.2. Módulo 2: Plataforma Web para Clientes (PI_WEB2)

- **Descripción:** La cara visible del proyecto, construida con Next.js, React y TypeScript para una experiencia de usuario excepcional.
- **Funcionalidades Principales:**
 - Sistema de Reservas Avanzado que consume la API.
 - Exploración de Servicios y Sedes.
 - Panel para Barberos para gestionar su agenda y galería.

4.3. Módulo 3: Panel de Gestión y Análisis (PI_NTP)

- **Descripción:** Herramienta estratégica en Python con Streamlit, diseñada para el análisis de datos y la administración, consumiendo la API Central.
- **Funcionalidades Principales:**
 - Dashboard Interactivo con KPIs y filtros en tiempo real.
 - Gestión y consulta del historial de citas.
 - Asistente de Inteligencia Artificial (Gemini) para análisis y reportes.

4.4. Módulo Adicional: Aplicación Web Independiente (PI_MOVIL2)

- **Descripción:** Un proyecto autocontenido desarrollado con **Spring Boot (Java)** para el backend y **VanillaJS** para el frontend.
- **Nota:** Este módulo **no se integra con la API Central** de Node.js y funciona como un sistema independiente.

5. Stack Tecnológico Consolidado

Capa	Tecnología	Propósito
Backend (API Central)	Node.js, Express.js	Lógica de negocio, API RESTful
Frontend (Cliente)	Next.js, React, TypeScript	Interfaz de usuario
Análisis de Datos	Python, Streamlit, Pandas	Panel de administración, IA

Base de Datos (API)	Archivos JSON	Almacenamiento de datos
Backend (Independiente)	Java, Spring Boot	Lógica del proyecto PI_MOVIL2

6. Conclusiones del Proyecto

La decisión de migrar a una arquitectura desacoplada con una API Central en Node.js ha sido clave para la flexibilidad y escalabilidad del ecosistema.

- **Evolución del diseño:** Se transitó de un modelo monolítico a uno de servicios, permitiendo que cada frontend (Next.js y Streamlit) evolucione de forma independiente.
- **Importancia de la comunicación:** La definición clara de los contratos de la API fue fundamental para que los equipos de frontend pudieran trabajar en paralelo.
- **Lecciones aprendidas:** Aprendimos a no subestimar el tiempo de integración entre servicios y la importancia de una documentación de API clara y actualizada.

El proyecto ha entregado una solución tecnológica funcional y moderna, sentando las bases para futuras expansiones.