

# Tarea 1: Anteproyecto

13 de Octubre del 2025

## Índice

<b>Índice.....</b>	<b>1</b>
<b>Visión general.....</b>	<b>1</b>
<b>1- Introducción.....</b>	<b>2</b>
<b>2- Definición del Problema.....</b>	<b>2</b>
<b>3. Objetivos.....</b>	<b>3</b>
<b>4. Antecedentes.....</b>	<b>4</b>
<b>5. Restricciones.....</b>	<b>4</b>
<b>6. Recursos.....</b>	<b>5</b>

## Visión general

Esta tarea tiene como objetivo analizar una idea de Proyecto Intermodular para final de Ciclo de Desarrollo de Aplicaciones Multiplataforma. En concreto, en este documento vamos a abarcar los siguientes 6 puntos:

1. Introducción.
2. Definición del Problema.
3. Objetivos.
4. Antecedentes.
5. Restricciones.
6. Recursos

# 1- Introducción

Mi Proyecto Final Intermodular del **Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma** consiste en el **desarrollo de una Aplicación Web de Gestión de Citas Online** para un Salón de Belleza y Peluquería unipersonal (autónoma). **El tiempo es Oro**, por ello la aplicación busca digitalizar y automatizar el proceso de reserva, permitiendo a los clientes agendar citas 24/7 y **liberando al profesional de la necesidad de atender llamadas o mensajes** constantes, optimizando su tiempo y flujo de trabajo diario (dentro o fuera de su jornada laboral).

## 2- Definición del Problema

El problema central a resolver es la ineficiencia en la gestión del tiempo y la comunicación del Profesional autónomo:

1. **Interrupciones constantes:** Las llamadas, los WhatsApps o mensajes directos vía redes sociales para solicitar o modificar citas así como la petición de información sobre algún tratamiento específico interrumpen continuamente el servicio al cliente actual, disminuyendo la calidad de trabajo y la concentración en el mismo, perdiendo así eficiencia en el servicio.
2. **Gestión manual de la agenda:** La gestión manual, ya sea en papel, agendas o calendarios, puede llegar a ser propensa a errores (dobles reservas, huecos perdidos, pérdida de citas por olvido de clientes, etc...) y consume tiempo valioso que podría dedicarse al servicio o al descanso fuera de la jornada laboral.

3. **Restricción horaria para el Cliente:** Los clientes sólo podrán reservar durante el horario laboral del Profesional, limitando la oportunidad de reserva fuera de ese rango.
4. **Ineficiencia en la planificación:** La duración variable de los servicios y tratamientos (por ejemplo un corte versus unas mechas) hace que la planificación manual sea más compleja, dificultando la optimización de los huecos disponibles entre un cliente y otro.

## 3. Objetivos

### 3.1 Objetivo Principal

Desarrollar una Aplicación Web funcional y robusta que automatice la reserva de citas, ajustando de manera dinámica la disponibilidad del calendario en función del tratamiento seleccionado , para una Profesional Autónoma.

### 3.2 Objetivos Específicos

1. **Automatización de las reservas:** el sistema permitirá a los clientes la reserva, consulta, cancelación o modificación de citas a través de la interfaz web las 24 horas del día, 7 días a la semana, durante todo el año.
2. **Lógica de agenda dinámica:** Implementar en el backend la capacidad de calcular y bloquear automáticamente los tramos horarios correctos en el calendario basándose en la duración específica del tratamiento y el tiempo de preparación antes y después del Servicio asociado.
3. **Reducción de ausencias:** Implementar un sistema de recordatorios de cita (por ejemplo a través de e-mail o WhatsApp) para disminuir la tasa de ausencias de clientes.
4. **Panel de Administración:** proporcionar a la Autónoma un panel de control intuitivo para visualizar, gestionar y bloquear (en días libres) y configurar sus servicios y lista de precios.

5. **Tecnología:** Utilizar **Java** o Kotlin con Spring Boot para el backend y HTML, CSS y JavaScript para el frontend (Vanilla JS), asegurando un desarrollo profesional y escalable del negocio.

## 4. Antecedentes

El proyecto se basa en la necesidad de replicar la funcionalidad central de software comercial de gestión de salones (como Booksy, Fresha o Koibox), pero adaptado a la escala y presupuesto de un autónomo o microempresa.

- **Soluciones Existentes:** El mercado ofrece soluciones completas de pago. Sin embargo, este proyecto busca ofrecer una solución a medida y gratuita para el profesional, con la ventaja de poder ser totalmente personalizada y controlada por mí.
- **Tecnológicos:** La arquitectura elegida (Backend API REST con Java/Kotlin y Frontend con Vanilla JS) es el estándar de la industria para aplicaciones web modernas que requieren alta interactividad y manejo de lógica de negocio compleja. Se estudiará la posibilidad también de usar Frontend SPA con React.

## 5. Restricciones

Las siguientes restricciones y requisitos deberán considerarse durante la fase de análisis así como en el desarrollo:

1. **Alcance Unipersonal:** La aplicación debe estar diseñada solo para un profesional (una única agenda), simplificando la gestión de recursos humanos.
2. **Base de Datos Relacional:** Se utilizará una base de datos relacional (ej. MySQL o PostgreSQL) para la persistencia de datos (clientes, servicios, citas) con un diseño optimizado para evitar conflictos horarios.
3. **Seguridad (SSL/CORS/Autenticación):** El proyecto debe implementar seguridad básica, incluyendo autenticación (JWT o similar) para el panel de administración y asegurar la comunicación entre frontend y backend (CORS y HTTPS).

4. **Disponibilidad:** La aplicación estará disponible en un entorno de hosting web accesible desde cualquier navegador (con diseño responsivo).
5. **Integración de Pagos (Fase 1 - Exclusión):** Por simplificación y restricción de tiempo, la funcionalidad de pagos o adelantos en línea no se incluirá en esta primera versión de Producto Mínimo Viable (MVP).

## 6. Recursos

### 6.1 Recursos Humanos y Conocimientos

El **coordinador** del Proyecto es **Pedro Rafael García Velasco**. Tutor del segundo curso y encargado de impartir las asignaturas de Proyecto Intermodular, Diseño de Interfaces y Diseño Web.

- **Desarrollador Principal:** Estudiante de Ciclo Superior José Manuel Sánchez Rosal
- **Conocimientos Requeridos:**
  - Backend: Java o Kotlin con framework Spring Boot (Spring Data JPA, Spring Security, REST).
  - Bases de Datos: SQL (manejo de tablas de usuarios, servicios y citas).
  - Frontend: HTML, CSS y JavaScript (o preprocesador como SASS), o uso de Framework React o similar (se decidirá más adelante).
  - Control de versiones: Manejo de Git y GitHub,
  - Despliegue de servidor: Heroku, AWS, VPS (se decidirá más adelante).

### 6.2 Recursos Hardware

- Portátil Medion Akoya Intel CORE i5 de 8 GEN con 16 gb de RAM.

### 6,3 Recursos Software y Herramientas

Los Recursos Software y herramientas que se detallan a continuación pueden ser objeto de modificación a lo largo del estudio, análisis y desarrollo del Anteproyecto:

Componente	Tecnología Seleccionada	Finalidad
<i>Backend</i>	<b>Java</b> o Kotlin con Spring Boot	Lógica de negocio, gestión de la API REST de citas.
<i>Frontend</i>	HTML, CSS y JS. Posibilidad de uso de React.	Interfaz de usuario interactiva y dinámica (calendario).
<i>Base de Datos</i>	PostgreSQL o <b>MySQL</b>	Almacenamiento persistente de citas, clientes y servicios.
<i>Control de Versiones</i>	Git / GitHub	Gestión del código fuente
<i>Entorno de Desarrollo</i>	IntelliJ IDEA y VS Code	IDE's para el desarrollo de Backend y Frontend.
<i>Despliegue</i>	Servidor Web (ej. Digital Ocean o hosting de prueba o similares...)	Acceso público a la aplicación.