

## **Informe Desarrollo Web II**

### **Nombre de la aplicación**

**Gestión virtual de citas legales**

**Versiones del documento**

Fecha	Versión	Descripción	Autor
18/06/2025	1.	<p>Gestión Virtual de Citas Legales</p> <p>Es una aplicación web que conecta abogados y clientes para agendar, organizar y gestionar citas legales de forma sencilla. Elimina trámites manuales, reduce errores y ahorra tiempo con recordatorios automáticos y una interfaz fácil de usar. Desarrollada con tecnologías modernas para garantizar seguridad y eficiencia.</p>	<p>Lauren Ayleen Flores Mejía</p> <p>Jenny Pamela Cárdenas Ferrufino</p> <p>Janeri Daney Juanez Garcia</p> <p>Kevin Geovanny Mendez Herrera</p>

Contenido	
Descripción de la solución .....	4
Resumen gerencial .....	4
Glosario .....	4
Restricciones y metas de arquitectura .....	4
Diagrama de Proceso .....	5
Requerimientos del sistema .....	6
Vista de Casos de uso de la aplicación.....	6
Tecnología.....	7
Desarrollo.....	7
Modelo Entidad Relación .....	9

## Descripción de la solución

### Resumen gerencial

El proyecto "Gestión Virtual de Citas Legales" es una aplicación web diseñada para agilizar y modernizar el proceso de programación y administración de citas en el ámbito legal. La solución permite a usuarios (clientes y profesionales del derecho) programar, gestionar y realizar seguimiento a citas legales de manera eficiente, reduciendo tiempos de espera y mejorando la organización de agendas. La plataforma incluye funcionalidades como autenticación de usuarios, calendario interactivo, notificaciones y gestión de historiales de citas.

### Glosario

Término	Definición
<b>Frontend</b>	Parte visible de la aplicación con la que interactúa el usuario
<b>Backend</b>	Lógica y procesamiento detrás de escenas que soporta la aplicación
<b>React</b>	Biblioteca JavaScript para construir interfaces de usuario
<b>TypeScript</b>	Lenguaje que añade tipos estáticos a JavaScript
<b>Next.js</b>	Framework de React para aplicaciones web renderizadas en servidor
<b>Cita Legal</b>	Reunión programada entre un profesional del derecho y su cliente
<b>API REST</b>	Conjunto de reglas para comunicación entre frontend y backend

### Restricciones y metas de arquitectura

La aplicación debe ser compatible con navegadores modernos (Chrome)

Debe soportar usuarios concurrentes

Debe cumplir con regulaciones de protección de datos personales

#### Metas de arquitectura:

Arquitectura modular y escalable

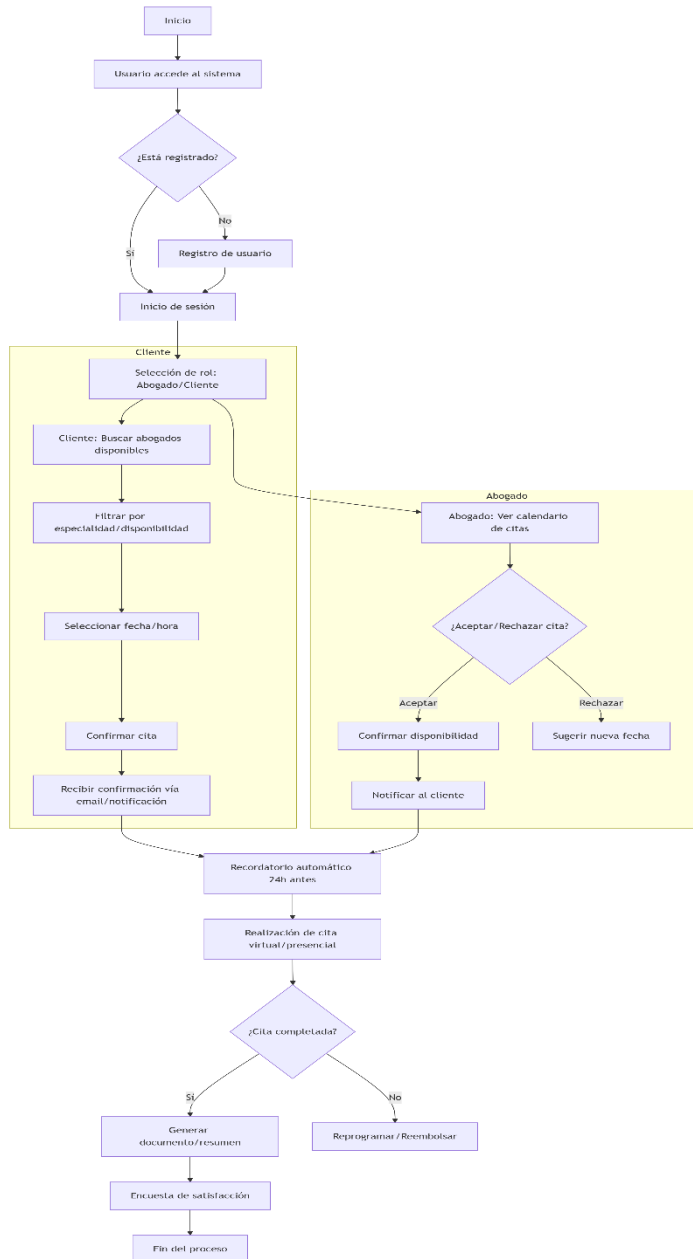
Interfaz intuitiva y accesible

Alto rendimiento en dispositivos móviles

Seguridad robusta para protección de datos sensibles

Fácil mantenimiento y actualización

## Diagrama de Proceso



## Requerimientos del sistema

### Vista de Casos de uso de la aplicación

Nombre del caso de uso	Descripción de la funcionalidad	Ruta del Caso de Uso
<b>Registro de usuario</b>	Permite a nuevos usuarios crear una cuenta en el sistema	<ul style="list-style-type: none"><li>• Validar datos personales</li><li>• Enviar correo de confirmación</li><li>• Almacenar información en base de datos</li></ul>
<b>Inicio de sesión</b>	Autentica a usuarios registrados	<ul style="list-style-type: none"><li>• Validar credenciales</li><li>• Asignar token de sesión</li><li>• Redirigir según rol (abogado/cliente)</li></ul>
<b>Programación de cita</b>	Permite agendar nuevas citas legales	<ul style="list-style-type: none"><li>• Mostrar disponibilidad de profesionales</li><li>• Validar horarios</li><li>• Enviar confirmación a ambas partes</li></ul>
<b>Gestión de citas</b>	Permite ver, editar o cancelar citas existentes	<ul style="list-style-type: none"><li>• Mostrar listado de citas</li><li>• Permitir modificaciones con restricciones</li><li>• Actualizar estado de la cita</li></ul>
<b>Notificaciones</b>	Envía recordatorios y actualizaciones	<ul style="list-style-type: none"><li>• Programar recordatorios automáticos</li><li>• Notificar cambios de estado</li><li>• Soporte para email y push notificaciones</li></ul>

## **Tecnología**

Frontend:

React.js con TypeScript para construcción de interfaces

Next.js para renderizado del lado del servidor

Tailwind CSS o Material-UI para estilos y componentes como una caja de herramientas

## **Desarrollo**

Backend:

Node.js con Express

Base de datos PostgreSQL

APIs Implementadas:

✓ Registro de Usuarios

POST /api/auth/register

Campos requeridos: nombre, email, contraseña, rol

Respuesta: (usuario creado)

✓ Inicio de Sesión

POST /api/auth/login

Campos requeridos: email, contraseña

Respuesta: 200 OK + JWT token

✓ Ruta de Prueba

GET /api/

Verifica que el servidor esté activo.

APIs Futuras (Roadmap):

➡ SOON Perfil de Usuario

GET /api/users/me (Obtener perfil)

PUT /api/users/me (Actualizar datos)

➡ SOON Autenticación Avanzada

POST /api/auth/logout (Cerrar sesión)

PUT /api/auth/password (Cambiar contraseña)

➡ SOON Gestión de Datos

GET /api/citas (Listar citas por usuario/abogado)

POST /api/documentos (Subir archivos legales)

**Herramientas adicionales:**

Git para control de versiones

Postman para pruebas de API



## Modelo Entidad Relación

